Hosted Sites

Database Schema

Introduction

The following outlines the details of each database schema (System, Vertical Market, Client). Each schema is broken into core and additional sections.

Each section has a header with all defined tables hyperlinked by name, their Create/Update/Delete (CUD) logging status, and a short description.

Each table shows the column definitions with a description of the intent of each column. Primary Key columns are underlined. Required columns are *italicized*.

The schema defined here is the same schema defined in the Visio diagram, and should be referenced for a clear understanding of table relationships.

System Schema

They system schema is broken into three parts: Core, Reports, and Template.

Core contains the main tables that define the client entities as well as their default settings and configuration.

Reports contains the tables that define available system reports, when to run them, and who to deliver them to.

Template contains the tables that define the templates, pages and features available for any client to use in their hosted site solution.

Core Schema

The core schema contains a handful of tables that are essential client identification, user profile and roles storage, and basic site functionality.

Table Name	Logged	Description
VerticalMarkets	Yes	Contains the main vertical market details such as id, name, connection string name, database instance.
Clients	Yes	Contains the main client details such as id, name, VerticalMarket affiliation, connection string name, database instance.
ClientDns	Yes	Contains the recognized DNS entries for any particular client. This will be read out of the request headers to determine appropriate site context.
Users	Yes	Contains user account information. ASP.NET Identity 2.0 will utilize this table for user authentication.
ClientUsers	Yes	Contains Client to User associations. A User can only access clients they are associated to.
Roles	Yes	Contains global Roles available for the system, such as *System Administrator* and *Authenticated User*.
UserRoles	Yes	Contains User to Role associations. Each User must be associated to minimally one Role - *Authenticated User*.
CUDHistory	No	Contains an audit trail of all create, update, and delete operations against auditable tables. **NOTE:** Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.
CUDHistoryData	No	Contains the change data pertaining to a CUDHistory entity. **NOTE:** Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.
ErrorLog	No	Allows logging of any error state or exception that terminates processing of a request, thread, or service, such as code generated exceptions or an invalid context (such as invalid path or query string parameter). **NOTE:** Client specific exceptions are logged here only in the

		event that the client database could not be determined or reached.

VerticalMarkets

Contains the main vertical market details such as id, name, connection string name, database instance.

Column Name	**Data Type**	**Description**
VerticalMarketId	int	Primary key identifier of the vertical market.
MarketName	nvarchar(200)	Display name of the vertical market.
ConnectionStringName	varchar(200)	Name of the connection string (defined in application and web config files) to use for the vertical market.
DatabaseName	varchar(200)	Name of the database instance (not server instance) for the vertical market. Will be injected into the connection string associated to the vertical market.
MarketDescription	nvarchar(400)	Description of the vertical market. May be used in a help context such as a tooltip.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

Clients

Contains the main client details such as id, name, VerticalMarket affiliation, connection string name, database instance.

Column Name	**Data Type**	**Description**
ClientId	int identity	Primary key identifier for the client.
ClientName	nvarchar(200)	Display name of the client.

ConnectionStringName	varchar(200)	Name of the connection string (defined in application and web config files) to use for the client.
DatabaseName	varchar(200)	Name of the database instance (not server instance) for the client. Will be injected into the connection string associated to the client.
VerticalMarketId	int	Identity of the VerticalMarket the client is associated with.
ClientDescription	nvarchar(400)	Description of the client. May be used in a help context such as a tooltip.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientDns

Contains the recognized DNS entries for any particular client. This will be read out of the request headers to determine appropriate site context.

Column Name	**Data Type**	**Description**
ClientDnsId	int identity	Primary key identifier for the ClientDns entry.
ClientId	int	Identifier of the client to which this dns entry belongs.
Dns	varchar(255)	Dns entry to recognize as belonging to the client. There must be a unique index on this column, and *ClientId* must be an include column for this index.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

Users

Contains user account information. ASP.NET Identity 2.0 will utilize this table for user authentication.

Column Name	**Data Type**	**Description**
UserId	int identity	Primary key identifier of the user account.
Email	nvarchar(256)	Email address of the user. This should be the same as *UserName*.
EmailConfirmed	bit	Boolean determining if the *Email* ownership has been confirmed. **NOTE:** This column should have a default constraint set to zero (0).
PasswordHash	nvarchar(max)	Hash of the password, used for authentication validation.
SecurityStamp	nvarchar(max)	Security stamp of the account, used for authentication validation.
PhoneNumber	nvarchar(max)	Phone number of the user.
PhoneNumberConfirmed	bit	Boolean determining if *PhoneNumber* ownership has been confirmed via SMS messaging. **NOTE:** This column should have a default constraint set to zero (0).
TwoFactorEnabled	bit	Boolean determining if two factor authentication is enabled. If true (1), either an *Email* or *PhoneNumber* entry must be confirmed for the user to authenticate. **NOTE:** This column should have a default constraint set to zero (0).
LockOutEndDateUtc	datetime	Utc date and time of the last account lock out.
LockoutEnabled	bit	Boolean determining if the user account can be locked out. **NOTE:** This column should have a default constraint set to zero (0).
AccessFailedCount	int	Most recent count of concurrent failed

		authentication attempts. **NOTE:** This column should have a default constraint set to zero (0).
UserName	nvarchar(256)	User identification used for authentication. This should be the same as the *Email* address for now. **NOTE:** There should be a unique index or constraint on this column.
FirstName	nvarchar(100)	Optional first name of the user.
LastName	nvarchar(100)	Optional last name of the user.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientUsers

Contains Client to User associations. A User can only access clients they are associated to.

Column Name	**Data Type**	**Description**
ClientId	int	First part composite primary key identifying the Client portion of the association.
UserId	int	Second part composite primary key identifying the User portion of the association.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this

		entity.

Roles

Contains global Roles available for the system, such as *System Administrator* and *Authenticated User*.

Column Name	Data Type	Description
RoleId	int	Primary key identifier of the Roles entity.
Name	nvarchar(256)	Descriptive name of the role. **NOTE:** There should be a unique index or constraint on this column.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

UserRoles

Contains User to Role associations. Each User must be associated to minimally one Role - Authenticated User.

Column Name	Data Type	Description
UserId	int	First part composite key identifying the Users entity of the association.
RoleId	int	Second part composite key identifying the Roles entity of the association.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.

ModifiedBy	int	Identity of the User who performed the last update of this entity.

CUDHistory

Contains an audit trail of all create, update and delete operations against auditable tables.

NOTE: Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Column Name	**Data Type**	**Description**
CUDHistoryId	int identity	Primary key identifier of the CUDHistory entity.
TableName	nvarchar(128)	Name of the table the CUD operation occurred on.
Key	int	Primary key identifier of the row the CUD operation occurred on.
SecondKey	nvarchar(200)	Secondary key identifier of the row the CUD operation occurred on. *NULL* if there is no secondary key.
ThirdKey	nvarchar(200)	Tertiary key identifier of the row the CUD operation occurred on. *NULL* if there is no tertiary key.
CUDType	char(1)	Character representing the type of update: *C* for create, *U* for update, *D* for delete.
UtcCUDDate	datetime	Utc date and time of the update.
BatchId	uniqueidentifier	Unique identifier for the update. Used to identify which rows were part of the same CUD operation. **NOTE:** A new id should be generated at the beginning of any procedure or trigger performing any audit logging and used for each row inserted during that operation.
UserId	int	Identity of the User who performed the CUD operation.

CUDHistoryData

Contains the change data pertaining to a CUDHistory entity.

NOTE: Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Column Name	**Data Type**	**Description**

CUDHistoryId	int	First part composite key identifying the CUDHistory entity this change data belongs to.
ColumnName	nvarchar(128)	Second part composite key identifying the column this change data is for.
SqlDbType	int	Type of data stored in the column. Should be one of the values available from the *System.Data.SqlDbType* enumeration..
OldValue	nvarchar(max)	Original value before the CUD operation. Always *NULL* for create operations.
NewValue	nvarchar(max)	New value after the CUD operation. Always *NULL* for delete operations.

ErrorLog

Allows logging of any error state or exception that terminates processing of a request, thread, or service, such as code generated exceptions or an invalid context (such as invalid path or query string parameter).

NOTE: Client specific exceptions are logged here only in the event that the client database could not be determined or reached.

Column Name	**Data Type**	**Description**
ErrorLogId	int identity	Primary key identifier of the ErrorLog entity.
ErrorCode	int	Error code identifying the type of error that stopped processing of the request. Examples might be context validation errors (invalid identifier or external identifier), malformed paths, not found errors (context identifiers such as taxonomy association id not found in client db), unhandled exceptions, etc. **NOTE:** The available values should be defined in an enumeration, as well as included in a scalar user defined function used as part of a check constraint for this column to ensure data integrity.
ErrorUtcDate	datetime	Utc date and time of the error occurrance.
Priority	int	Priority level of the exception. Inherited from Enterprise Library.
Severity	nvarchar(32)	Severity of the exception. Inherited from Enterprise

		Library.
Title	nvarchar(256)	Title of the exception handling policy. Inherited from Enterprise Library.
MachineName	nvarchar(32)	Name of the machine reporting the exception. Inherited from Enterprise Library.
AppDomainName	nvarchar(512)	Name of the application domain reporting the exception. Inherited from Enterprise Library.
ProcessID	nvarchar(256)	ID of the process reporting the exception. Inherited from Enterprise Library.
ProcessName	nvarchar(512)	Name of the process (usually executable + path) reporting the exception. Inherited from Enterprise Library.
ThreadName	nvarchar(512)	Name of the thread the exception occurred on. As a thread only has a name if explicitly set, this is usually *NULL*. Inherited from Enterprise Library.
Win32ThreadId	nvarchar(128)	ID of the thread the exception occurred on. Inherited from Enterprise Library.
EventId	int	Event ID of the exception. Equivalent to the event id of an event log entry. Inherited from Enterprise Library.
ClientId	int	Identifies the Client context this exception occurred in. *NULL* if client context could not be determined. **NOTE:** This column does NOT have a foreign key constraint to the Clients table, so client data can be deleted and/or changed without affecting the logging function.
SiteActivityId	int	If set, identifies which SiteActivity entity the error belongs to.
Message	nvarchar(1500)	Short message describing the error, such as context validation error details. Inherited from Enterprise Library.
FormattedMessage	nvarchar(max)	Error details such as exception message and stack trace. Inherited from Enterprise Library.

Reports Schema

The reports schema contains the tables that define the available system reports, scheduled reports, and report recipients.

Table Name	**Logged**	**Description**
Reports	No	Defines the global system reports available for each client.
ReportSchedule	Yes	Defines any scheduled reports, their reporting interval, last and next run dates.
ReportScheduleRecipients	Yes	Defines the recipients of a specific scheduled report.

Reports

Defines the global system reports available for each client.

Column Name	**Data Type**	**Description**
ReportId	int identity	Primary key identifier of the report entity.
ReportName	nvarchar(200)	Name of the report.
ReportDescription	nvarchar(400)	Description of the report.

ReportSchedule

Defines any scheduled reports, their reporting interval, last and next run dates.

Column Name	**Data Type**	**Description**
ReportScheduleId	int	Primary key identifier of the scheduled report.
ReportId	int	Identifies the Report this schedule is for.
ClientId	int	Identifies the Client this schedule is for.
IsEnabled	bit	Boolean flag that determines if the report is enabled.
FrequencyType	int	Defines the type of frequency for this report. Available types should be defined in an

		enumeration as follows: 1. Run Once 2. Every X Days 3. Weekly 4. Monthly 5. Quarterly 6. Bi-Annually 7. Annually
FrequencyInterval	int	Defines the interval to be used for determining next run date. Interval value is dependent on the FrequencyType setting with the allowed values dependent on the type: ● Run Once ○ Must be 0. ● Every X Days ○ Must be greater than 0. ● Weekly ○ Must be between 1 and 7. ● Monthly ○ Must be between 1 and 31. ● Quarterly ○ Must be 0. ● Bi-Annually ○ Must be 0. ● Annually ○ Must be 0. **NOTE:** There should be a check constraint on the table that validates the type and interval based on the allowed values defined here. Please see the POC for details.
UtcFirstScheduledRunDate	smalldatetime	First scheduled date for the report to run.
UtcLastScheduledRunDate	smalldatetime	The date the report was scheduled to run when it was run last. Used to determine next date.
UtcLastActualRunDate	datetime	The actual date and time the report ran last. May not be the same as the last scheduled report, in the case of a report being missed or skipped due to an error.
UtcModifiedDate	datetime	Utc date and time of last update.

		NOTE: This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.
UtcNextScheduledRunDate	smalldatetime computed	Computed persisted column that calculates the next run date based off of the *FirstScheduledRunDate*, *LastScheduledRunDate*, *FrequencyeType*, and *FrequencyInterval*. **NOTE:** There should be a deterministic udf scalar function that calculates this value. Please see the POC for details.
FrequencyDescription	varchar(200) computed	Computed persisted column that describes the schedule pattern in easy to understand terms. **NOTE:** There should be a deterministic udf scalar function that calculates this value. Please see the POC for details.

ReportScheduleRecipients

Defines the recipients of a specific scheduled report.

Column Name	**Data Type**	**Description**
ReportScheduleId	int	First part composite key identifying the report schedule this recipient is for.
Email	nvarchar(256)	Second part composite key identifying the recipient for the report schedule.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

Template Schema

The template schema contains four tables that define the templates, pages, features and associations.

NOTE: Because all updates to these tables are done directly with sql script during releases, none of the CUD operations on these tables are logged for auditing purposes.

Table Name	Logged	Description
Template	No	Defines available templates for hosted sites. Each entry must correspond with a developed solution available as part of the application.
Page	No	Defines available pages for hosted sites. Each Page corresponds to a particular function or display. The actual layout will be defined by the Template associations. In the MVC world, the Page comprises of the Mode while the Template associated defines the View.
TemplatePage	No	Defines associations between pages and templates.
TemplateFeature	No	Defines Template specific features that can be configured by the client. This provides a level of functional customization.
TemplatePageFeature	No	Defines Template / Page combination specific features that can be configured by the client. This provides a level of functional customization.
TemplateText	No	Defines all available resource keys that are global to a specific Template such as CSS text, copyright notice, button labels and tooltips, etc.
TemplatePageText	No	Defines all available resource keys that are local to each specific Template / Page combination such as header, footer, glossary, etc.
TemplateNavigation	No	Defines available navigation areas for the entire Template. Includes the default xml for this area, as well as the xsl transformation to convert the xml to appropriate html for the Template.
TemplatePageNavigation	No	Defines available navigation areas for a specific Template / Page combination. Includes the default xml for this area, as well as the xsl transformation to convert the xml to appropriate html for the Template / Page combination.

Template

Defines available templates for hosted sites. Each entry must correspond with a developed solution available as part of the application.

Column Name	Data Type	Description
TemplateId	int	Primary key identifier for the Template entity.
Name	nvarchar(200)	Name of the Template.
Description	nvarchar(400)	Description of the Template.

Page

Defines available pages for hosted sites. Each Page corresponds to a particular function or display. The actual layout will be defined by the Template associations. In the MVC world, the Page comprises of the Mode while the Template associated defines the View.

Column Name	Data Type	Description
PageId	int	Primary key identifier for the Page entity.
Name	nvarchar(200)	Name of the Page.
Description	nvarchar(400)	Description of the Page.

TemplatePage

Defines associations between pages and templates.

Column Name	Data Type	Description
TemplateId	int	First part composite key defining the Template association.
PageId	int	Second part composite key defining the Page association.

TemplateFeature

Defines Template specific features that can be configured by the client. This provides a level of functional customization.

Column Name	Data Type	Description
TemplateId	int	First part composite key identifying the Template the feature belongs to.
Key	varchar(200)	Second part composite key defining a meaningful

		identifier for the feature.
Description	nvarchar(400)	Description of the feature.

TemplatePageFeature

Defines Template / Page combination specific features that can be configured by the client. This provides a level of functional customization.

Column Name	**Data Type**	**Description**
TemplateId	int	First part composite key identifying the Template the feature belongs to.
PageId	int	Second part composite key identifying the Page the feature belongs to.
Key	varchar(200)	Third part composite key defining a meaningful identifier for the feature.
Description	varchar(400)	Description of the feature.

TemplateText

Defines all available resource keys that are global to a specific Template such as CSS text, copyright notice, button labels and tooltips, etc.

Column Name	**Data Type**	**Description**
TemplateId	int	First part composite key identifying the Template the text belongs to.
ResourceKey	varchar(200)	Second part composite key defining the resource key for the feature.
Name	nvarchar(200)	Descriptive name of the text.
IsHtml	bit	Boolean flag that determines if the text is html (such as a copyright or disclaimer) or plain text (such as a label).
DefaultText	nvarchar(max)	Defines the default text to use for this resource if none is set in the client database.
Description	nvarchar(400)	Longer description of the text.

TemplatePageText

Defines all available resource keys that are local to each specific Template / Page combination such as header, footer, glossary, etc.

Column Name	Data Type	Description
TemplateId	int	First part composite key identifying the Template the text belongs to.
PageId	int	Second part composite key identifying the Page the text belongs to.
ResourceKey	varchar(200)	Third part composite key defining the resource key for the feature.
Name	nvarchar(200)	Descriptive name of the text.
IsHtml	bit	Boolean flag that determines if the text is html (such as header or glossary) or plain text (such as a tooltip).
DefaultText	nvarchar(max)	Defines the default text to use for this resource if none is set in the client database.
Description	nvarchar(400)	Longer description of the text.

TemplateNavigation

Defines available navigation areas for the entire Template. Includes the default xml for this area, as well as the xsl transformation to convert the xml to appropriate html for the Template.

Column Name	Data Type	Description
TemplateId	int	First part composite key identifying the Template the navigation entity belongs to.
NavigationKey	varchar(200)	Second part composite key defining the navigation key for the entity.
Name	nvarchar(200)	Descriptive name of the navigation area.
XslTransform	xml (schemabound)	XSLT XML document that defines how to transform the navigation xml to html for the Site. **NOTE:** This xml should be typed (schemabound) using the following schema: http://www.w3.org/1999/XSL/Transform
DefaultNavigationXml	xml (schemabound)	Default xml to use in case the client has not

		customized the menu.
Description	nvarchar(400)	Detailed description of the navigation area.

TemplatePageNavigation

Defines available navigation areas for a specific Template / Page combination. Includes the default xml for this area, as well as the xsl transformation to convert the xml to appropriate html for the Template / Page combination.

Column Name	**Data Type**	**Description**
TemplateId	int	First part composite key identifying the Template the navigation entity belongs to.
PageId	int	Second part composite key identifying the Page the navigation entity belongs to.
NavigationKey	varchar(200)	Third part composite key defining the navigation key for the entity.
Name	nvarchar(200)	Descriptive name of the navigation area.
XslTransform	xml (schemabound)	XSLT XML document that defines how to transform the navigation xml to html for the Site. **NOTE:** This xml should be typed (schemabound) using the following schema: http://www.w3.org/1999/XSL/Transform
DefaultNavigationXml	xml (schemabound)	Default xml to use in case the client has not customized the menu.
Description	nvarchar(400)	Detailed description of the navigation area.

Vertical Market Schema

The Vertical Market Schema consists of a Core schema that defines shared table structure across all vertical markets, as well as a market specific schema that defines the market’s taxonomy.

Core schema

The Core schema shared by all vertical markets mainly consists of document tables and associated metadata. Whereas taxonomy architecture of each market can vary greatly, metadata pertaining to specific documents is more universal which is what allows us to create a common description for these tables.

Table Name	Logged	Description
DocumentType	Yes	Defines the available document types specific to the vertical market.
Document	Yes	Defines the vertical market documents and basic metadata such as file name and size.
DocumentMetaData	Yes	Defines additional metadata for a Document entity in a dictionary format.
TaxonomyLevel	Yes	Defines the taxonomy hierarchy for the vertical market and associates them to market specific schema.
CUDHistory	No	Contains an audit trail of all create, update, and delete operations against auditable tables. **NOTE:** Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.
CUDHistoryData	No	Contains the change data pertaining to a CUDHistory entity. **NOTE:** Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

DocumentType

Defines the available document types specific to the vertical market.

Column Name	Data Type	Description
DocumentTypeId	int	Primary key identifier for the DocumentType entity.
Name	nvarchar(100)	Unique meaningful name for the DocumentType entity. **NOTE:** There must be a unique index or constraint on this field.

MarketId	nvarchar(100)	Unique market identifier for the DocumentType entity. **NOTE:** There must be a unique index of constraint on this field.
Description	nvarchar(400)	Optional additional description of the DocumentType entity.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

Document

Defines the vertical market documents and basic metadata such as file name and size.

Column Name	**Data Type**	**Description**
DocumentId	int	Primary key identifier for the Document entity.
DocumentTypeId	int	Associates the Document entity to a specific DocumentType entity.
TaxonomyLevel	int	Hierarchical level of the taxonomy entity that owns this Document..
TaxonomyId	int	Identifier of the taxonomy entity that owns this Document.
FileName	nvarchar(260)	Original name of the uploaded file for this Document. 260 characters defines maximum file name length per Microsoft.
ContentUri	nvarchar(2083)	Universal Resource Identifier for the content (file) of the Document. points to the location accessible by the service or application for content retrieval.
MimeType	varchar(127)	Mimetype of the Document, inferred from the file name extension during upload.
Size	bigint	Number of bytes in the file.
IsPrivate	bit	Boolean determining whether this Document is

		browsable without a specific association
Name	nvarchar(100)	Name of Document. This is originally set to *FileName* during creation, but can be changed when editing the metadata.
Description	nvarchar(400)	Optional detailed description of the Document.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

DocumentMetaData

Defines additional metadata for a Document entity in a dictionary format.

Column Name	Data Type	Description
DocumentId	int	First part composite key identifying the Document this key/value pair belongs to.
Key	varchar(20)	Unique identifier DocumentMetaData entity.
DataType	int	Identifies the supported data type of this entity. **NOTE:** Valid values should be defined in an enumeration for code validation, and a check constraint with a udf function should be used for data validation of this value.
Order	int	Optional order of entries for the parent Document entity.
IntegerValue	int	Holds the value for integer metadata.
BooleanValue	bit	Holds the value for boolean metadata.
DateTimeValue	datetime	Holds the value for datetime metadata.
StringValue	nvarchar(max)	Holds the value for string metadata.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint

		setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyLevel

Defines the taxonomy hierarchy for the vertical market and associates them to market specific schema.

Column Name	**Data Type**	**Description**
Level	int	Primary Key identifier of the TaxonomyLevel entity which corresponds to the hierarchy level of the entity.
Name	nvarchar(200)	Unique name of the TaxonomyLevel. **NOTE:** There must be a unique constraint or index on this column.
SourceTable	varchar(128)	Name of the table in the schema that holds the data for this TaxonomyLevel. **NOTE:** There must be a unique constraint or index on this column.
MarketIdColumn	varchar(128)	Name of the columns from the *SourceTable* that contains the unique market identifier for the taxonomy level. **NOTE:** The column can be of any type, but must not be longer than 100 characters in string representation.
Description	nvarchar(400)	Optional detailed description of the TaxonomyLevel.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

CUDHistory

Contains an audit trail of all create, update and delete operations against auditable tables.

NOTE: Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Column Name	Data Type	Description
CUDHistoryId	int identity	Primary key identifier of the CUDHistory entity.
TableName	nvarchar(128)	Name of the table the CUD operation occurred on.
Key	int	Primary key identifier of the row the CUD operation occurred on.
SecondKey	nvarchar(200)	Secondary key identifier of the row the CUD operation occurred on. *NULL* if there is no secondary key.
ThirdKey	nvarchar(200)	Tertiary key identifier of the row the CUD operation occurred on. *NULL* if there is no tertiary key.
CUDType	char(1)	Character representing the type of update: *C* for create, *U* for update, *D* for delete.
UtcCUDDate	datetime	Utc date and time of the update.
BatchId	uniqueidentifier	Unique identifier for the update. Used to identify which rows were part of the same CUD operation. **NOTE:** A new id should be generated at the beginning of any procedure or trigger performing any audit logging and used for each row inserted during that operation.
UserId	int	Identity of the User who performed the CUD operation.

CUDHistoryData

Contains the change data pertaining to a CUDHistory entity.

NOTE: Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Column Name	Data Type	Description
CUDHistoryId	int	First part composite key identifying the CUDHistory entity this change data belongs to.
ColumnName	nvarchar(128)	Second part composite key identifying the column this change data is for.
SqlDbType	int	Type of data stored in the column. Should be one of the

		values available from the *System.Data.SqlDbType* enumeration..
OldValue	nvarchar(max)	Original value before the CUD operation. Always *NULL* for create operations.
NewValue	nvarchar(max)	New value after the CUD operation. Always *NULL* for delete operations.

U.S.A Market Schema

The USA Vertical Market has an already existent database schema. The pre existing Taxonomy schema works as is however there must be a translation between existing document tables and the schema defined by the Vertical Market Core Schema above.

This will be done by translating the document specific tables (ProsDocs and DocType) into indexed views defining the schema from the Core (Document and DocumentType respectively).

NOTE: Because these tables are only updated directly (via script) or through internal processes (such as the *Ingestor*), there is no need to log CUD operations of these tables for auditing purposes.

If it is determined there is a need in a future release, all the legacy processes that could update these tables would need to be updated to perform such audit logging in the CUDHistory table.

Table Name	Logged	Description
Company	No	This table defines Fund Families as Company entities. Fund Families are the highest level in the USA taxonomy hierarchy, and have a taxonomy level of 0.
Prospectus	No	This table defines Funds as Prospectus entities. Funds are directly under a Company, and have a taxonomy level of 1.
ProsTicker	No	This table defines Shares as ProsTicker entities. Shares are directly under Prospectus, and have a taxonomy level of 2.
ProsDocs	No	This table defines all the USA market specific documents ingested. Documents are associated with the Fund, or Prospectus entity. This table is the source for the Document view that is part of the core schema.
DocType	No	This table defines the available document types for the USA market. This table is the source for the DocumentType view that is part of the core schema.

Company

This table defines Fund Families as Company entities. Fund Families are the highest level in the USA taxonomy hierarchy, and have a taxonomy level of 0.

Column Name	Data Type	Description
CompanyID	int	Primary key identifier of the Company entity.

CompanyName	nvarchar(255)	This holds the actual Family Name/Company Name used in U.S. vertical market Ingestion.
CompanyURL	nvarchar(255)	The actual URL of the Company added to this table.
Image	nvarchar(50)	Image field holds the Image of the company.only used on RP Admin site.
AlertMessage	text	any alertmessage that needs to be defined. currently this field is null for all companies
Online	char(1)	defines if the Company is online or offline Y being online and N being offline
CompLevel	int	1 or 2 .1 means this company is visible in Hosted Sites,FundCompli and Rightprospectus.com . 2 means this company can only be viewed by RRD folks. Mostly this flag is set if we add a new Fund Family /Company and it needs to be visible only to RRD folks until it is supposed to be viewed by the public.

Prospectus

This table defines Funds as Prospectus entities. Funds are directly under a Company, and have a taxonomy level of 1.

Column Name	**Data Type**	**Description**
ProsId	int	Primary key identifier of the Prospectus entity.
ProsName	nvarchar(255)	
Company	nvarchar(255)	
URL	nvarchar(255)	
ProsDate	datetime	
RevisedProsDate	datetime	
SAISURL	nvarchar(255)	
SuppURL	nvarchar(255)	
SAISURL2	nvarchar(255)	
SuppURL2	nvarchar(255)	
SAISURL3	nvarchar(255)	

SuppURL3	nvarchar(255)	
SAISURL4	nvarchar(255)	
SuppURL4	nvarchar(255)	
SAISURL5	nvarchar(255)	
SuppURL5	nvarchar(255)	
AltURL	nvarchar(255)	
CompanyId	int	Identifies the Company this entity belongs to.
Online	char(1)	
ARDate	datetime	
RevisedARDate	datetime	
FEY	varchar(5)	
PHDate	datetime	
RevisedPHDate	datetime	
PVRDate	datetime	
RevisedPVRDate	datetime	
SDate	datetime	
RevisedSDate	datetime	
SSDate	datetime	
SPDate	datetime	
RevisedSPDate	datetime	
FSDate	datetime	
RevisedFSDate	dateitme	
COMDate	datetime	
RevisedCOMDate	datetime	
GWFDate	datetime	
RevisedGWFDate	datetime	

ProsTicker

This table defines Shares as ProsTicker entities. Shares are directly under Prospectus, and have a taxonomy level of 2.

Column Name	Data Type	Description
TickerID	int	Primary key identifier for the ProsTicker entity.
ProspectusID	int	Identifies the Prospectus this entity belongs to.
TickerSymbol	nvarchar(10)	
Class	varchar(100)	
FileNumber	varchar(50)	
CIK	varchar(50)	
SeriesId	varchar(50)	
ClassContractID	varchar(50)	
CUSIP	varchar(10)	
LIPPER	varchar(50)	

ProsDocs

This table defines all the USA market specific documents ingested. Documents are associated with the Fund, or Prospectus entity. This table is the source for the Document view that is part of the core schema

Column Name	Data Type	Description
ProsDocId	int	Primary key identifier of the ProsDocs entity.
ProsId	int	Identifies the Prospectus the document belongs to.
ProsDocTypeId	varchar(3)	Identifies the DocType the document belongs to.
ProsDocOrder	int	
ProsDocURL	varchar(500)	
ProsDocAltURL	varchar(500)	
ProsDocUseAltURL	bit	

ProsDocLevel	int	
Removed	bit	
ProsDocPDFInitial	int	
ProsDocUsePDF	bit	
ProsDocBackUpURL	varchar(500)	
ProsDocBackUpURLArchive	int	
isBackUpURLSynchronized	tinyint	
ClientID	int	
CustomizedTypeID	int	
ProsDocFundSiteURL	varchar(500)	
PageCount	int	
PageSizeHeight	decimal(5, 2)	

DocType

This table defines the available document types for the USA market. This table is the source for the DocumentType view that is part of the core schema.

NOTE: the columns in orange are new columns necessary to match the new core schema.

Column Name	**Data Type**	**Description**
DocTypeId	varchar(4)	Primary key identifier for the DocType entity.
DocTypeDesc	varchar(255)	
DocPriority	int	
DocumentTypeId	int identity	Identity integer for the DocType entity. To be used as the primary key identifier in the indexed view.
Name	nvarchar(100)	Name of the DocType entity. To be used as the *Name* in the indexed view.

Client Schema

The Client Schema is broken into three parts: Core, Vertical Market Integration, and Template & Design. Core contains the main tables that either contain data specific to the client, and/or are vital to both the Vertical Market Integration and Template & Design schemas.

Core Schema

Core Schema contains the following six tables. Site specifically defines the entity that connects the taxonomy associations from the vertical market to the Template and Page customizations. ClientSettings simply defines the default configurations such as the default Site for the client.

The rest of the tables define client specific documents and their metadata and groupings.

Table Name	Logged	Description
Site	Yes	Defines a Site entity, which is the root entity for vertical market specific associations and customizations, as well as Template specific associations and customizations. Every client must have at least one Site.
ClientSettings	Yes	Defines the default values for the client, such as the default Site. **NOTE:** This table is designed to have a single row. If more than one row is inserted, only the top level row will ever be read.
ClientDocumentType	Yes	Defines the document types for the client specific documents. An example is Money Market Document.
ClientDocument	Yes	Houses the metadata for a client specific document.
ClientDocumentData	Yes	Contains the actual content for the ClientDocument entity.
ClientDocumentGroup	Yes	Defines a grouping of ClientDocument entities.
ClientDocumentGroupClientDocument	Yes	Associates a ClientDocument to a ClientDocumentGroup.
UrlRewrite	Yes	Contains regular expression matches to match against request url paths, as well as rewrite format strings. The regular expression match pattern must contain named grouped submatches. These submatches will be used in the rewrite format string as token replacements. Each

		named group will use the format of {namedgroup} in the format string As an example: a submatch group named group1 would be replaced in a format string that contained {group1}. There is an assumption that each grouped match will only have one occurrence. **NOTE:** This table should be cached on the web server as part of the solution for validating requests against regular expressions. Each entry should be cached as a struct containing the regular expression pattern as a compiled *RegEx* instance as a member, as well as the format string for the rewrite as a member of the struct. This will help on any performance concerns relates to this feature. **NOTE 2:** Ideally this would be handled in an http module before any other processing occurs on the pipeline. There is a chance that this might interfere with MVC processing, if so we will need to follow a method similar to the one outlined here: http://stackoverflow.com/questions/799511/how-to-simulate-server-transfer-in-asp-net-mvc.
[SiteActivity]()	No	Contains a log of all site requests for the client. When a request is processed a stored procedure should be called to create the activity entry and an id should be returned. **NOTE:** This id should be stored for the request lifecycle so exceptions and error states can be associated with it.
[UserAgent]()	No	Normalized format for storing user agent strings which are not unique to a request. **NOTE:** There must be a unique index on the *UserAgentHash* and *UserAgentLength* persisted computed columns. This ensures integrity and provides an indexed solution for looking up matching user agents during

		activity logging events.
Uri	No	Normalized storage of Uri’s (Request, Parsed Request, Referrer) that are associated to a SiteActivity entity. Many of the entities will have the same value for these Uri’s - normalizing them to a separate table reduces burden on storage capacity with request activity logging. **NOTE:** There must be a unique index on the *UriHash* and *UriLength* persisted computed columns. This provides an indexed solution for looking up matching uri’s during activity logging events.
ErrorLog	No	Allows logging of any error state or exception that terminates processing of a request, thread, or service, such as code generated exceptions or an invalid context (such as invalid path or query string parameter).
ClientRole	Yes	Defines all the client specific roles a User can belong to.
UserClientRole	Yes	Associates a ClientRole to a User.
CUDHistory	No	Contains an audit trail of all create, update, and delete operations against auditable tables. **NOTE:** Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.
CUDHistoryData	No	Contains the change data pertaining to a CUDHistory entity. **NOTE:** Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Site

Defines a Site entity, which is the root entity for vertical market specific associations and customizations, as well as Template specific associations and customizations. Every client must have at least one Site.

Column Name	**Data Type**	**Description**
SiteId	int identity	Primary key for the row.
Name	nvarchar(100)	Name of the Site. **NOTE:** There must be a unique index or constraint on this column.
TemplateId	int	Identity of the Template the Site uses.
DefaultPageId	int	Id of the Page to render as the landing page for the root of the Site.
ParentSiteId	int	Identifies the parent Site, if it exists. This allows a secondary site to share the data association of a primary site, but with the ability to override Site specific customizations. This feature will be useful for clients who want to slightly customize the look and feel of the same data based on origin of the use reflected in the path and query string of the request. This also allows tracking of these requests by origin (reflected in path and query string).
Description	nvarchar(400)	Description of the Site.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientSettings

Defines the default values for the client, such as the default Site.

NOTE: This table is designed to have a single row. If more than one row is inserted, only the top level row will ever be read.

Column Name	Data Type	Description
ClientId	int	Contains the client specific identity found in the client table in the system schema.
DefaultSiteId	int	Contains the identifier of the default Site for the client. This is the Site used when requesting a resource without specifying a Site in the path or query string of the resource Uri.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting theActivityError initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientDocumentType

Defines the document types for the client specific documents. An example is Money Market Document.

Column Name	Data Type	Description
ClientDocumentTypeId	int identity	Primary key of the client document type.
Name	nvarchar(100)	Name of the client document type.
Description	nvarchar(400)	Description of the client document type.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientDocument

Houses the metadata for a client specific document.

Column Name	**Data Type**	**Description**
ClientDocumentId	int identity	Primary key of the client document.
ClientDocumentTypeId	int	Type of the client document.
FileName	nvarchar(260)	Original file name of the client document. The 260 character limit is defined by the NTFS and windows file system as max file name length.
MimeType	nvarchar(127)	Mime type of the document. This is required in the event of serving the file as an attachment to an end user. The 127 character limit is defined by various internet standards.
IsPrivate	bit	Bit determining whether the document is browsable by its associations
ContentUri	nvarchar(2083)	Uri mapped to the resource, allowing flexibility for network storage of the document. When file is stored in ClientDocumentData, this column should be null.
Name	nvarchar(100)	Display name of the document.
Description	nvarchar(400)	Description of the document.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientDocumentData

Contains the actual content for the ClientDocument entity.

Column Name	**Data Type**	**Description**
ClientDocumentId	int	Primary key, referencing ClientDocument table.

Data	varbinary(max)	Actual data of the document.
HasData	bit (computed)	Persisted computed column set to 1 if *Data* is not null, 0 otherwise.
DataLength	int (computed)	Persisted computed column calculating the length of the data. Can be Null if *Data* is Null.
DataHash	varbinary(20) (computed)	Persisted computed column calculating an indexible hash for the data which can be used for data lookups. Can be Null if *Data* is null.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientDocumentGroup

Defines a grouping of ClientDocument entities.

Column Name	Data Type	Description
ClientDocumentGroupId	int identity	Primary key of the client document group entity.
Name	nvarchar(100)	Display name of the client document group.
Description	nvarchar(400)	Description of the client document group.
ParentClientDocumentGroupId	int	Parent group of the client document group. Null if there is no parent.
CssClass	varchar(50)	If set, defines a css class name to inject into any element displaying this ClientDocumentGroup entity. Can be used for customization of look and feel.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default

		constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

ClientDocumentGroupClientDocument

Associates a ClientDocument to a ClientDocumentGroup.

Column Name	**Data Type**	**Description**
ClientDocumentGroupId	int	First part of composite primary key defining association to the client document group.
ClientDocumentId	int	Second part of the composite primary key defining association to the client document.
Order	int	Defines the order of associated client documents to a client document group.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

UrlRewrite

Contains regular expression matches to match against request url paths, as well as rewrite format strings.

The regular expression match pattern must contain named grouped submatches. These submatches will be used in the rewrite format string as token replacements. Each named group will use the format of {namedgroup} in the format string

As an example: a submatch group named group1 would be replaced in a format string that contained {group1}.

There is an assumption that each grouped match will only have one occurrence.

NOTE: This table should be cached on the web server as part of the solution for validating requests against regular expressions. Each entry should be cached as a struct containing the regular expression pattern as a compiled *RegEx* instance as a member, as well as the format string for the rewrite as a member of the struct. This will help on any performance concerns relates to this feature.

NOTE 2: Ideally this would be handled in an http module before any other processing occurs on the pipeline. There is a chance that this might interfere with MVC processing, if so we will need to follow a method similar to the one outlined here: http://stackoverflow.com/questions/799511/how-to-simulate-server-transfer-in-asp-net-mvc.

Column Name	**Data Type**	**Description**
UrlRewriteId	int identity	Primary key identity of the UrlRewrite entity.
MatchPattern	nvarchar(2083)	Contains the regular expression pattern to match against the path of the incoming request. Must contained grouped submatches to be used in the replacement rewrite format. This pattern will be cached by the web application as a compiled Regex instance.
RewriteFormat	nvarchar(2083)	Contains the format string to be used for the UrlRewrite entity. Each named group match from the regular expression should be included in the format string in the format of *{namedgroup}* where *namedgroup* is the name of the matched group.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

SiteActivity

Contains a log of all site requests for the client. When a request is processed a stored procedure should be called to create the activity entry and an id should be returned.

NOTE: This id should be stored for the request lifecycle so exceptions and error states can be associated with it.

Column Name	**Data Type**	**Description**

SiteActivityId	int identity	Primary key identity of the SiteActivity entity.
SiteId	int	Identifies the Site this request was for (determined by context or by the default set in ClientSettings).
ClientIPAddress	varchar(15)	IP address of the client sending the request.
UserAgentId	int	ID of the matching UserAgent entity. If no match exists at time of entity creation, a new entity will be created in the UserAgent table and then associated here.
RequestUtcDate	datetime	Utc date and time of the request.
HttpMethod	varchar(20)	Method of the request (usually GET or POST).
RequestUri	int	Identifies the Uri entity containing the original request uri string.
ParsedRequestUri	int	Identifies the Uri entity containing the parsed (for url rewrite matches and external identifiers) request uri string.
ServerName	varchar(15)	Name of the server responding to the request.
ReferrerUri	int	Identifies the Uri entity containing the referrer uri. Null if none.
UserId	int	Associates the entity to an authenticated User. Null if the request is anonymous.
PageId	int	Associates the entity to a specific Page. Null if Page cannot be determined from request context.
TaxonomyAssociationGroupId	int	Associates the entity to a specific TaxonomyAssociationGroup. Null if not included in the request context.
TaxonomyAssociationId	int	Associates the entity to a specific TaxonomyAssociation. Null if not included in the request context.
DocumentTypeId	int	Associates the entity to a specific DocumentType. Null if not included in the request context.
ClientDocumentGroupId	int	Associates the entity to a specific ClientDocumentGroup. Null if not included in

		the request context.
ClientDocumentId	int	Associates the entity to a specific ClientDocument. Null if not included in the request context.

UserAgent

Normalized format for storing user agent strings which are not unique to a request.

NOTE: There must be a unique index on the *UserAgentHash* and *UserAgentLength* persisted computed columns. This ensures integrity and provides an indexed solution for looking up matching user agents during activity logging events.

Column Name	**Data Type**	**Description**
UserAgentId	int identity	Primary key identity of the UserAgent entity.
UserAgentString	nvarchar(max)	Unique user agent string value associated to the entity.
UserAgentHash	varbinary(20) computed	Computed persisted hash value of the *UserAgentString*.
UserAgentLength	int computed	Computed persisted length of the *UserAgentString*.

Uri

Normalized storage of Uris (Request, Parsed Request, Referrer) that are associated to a SiteActivity entity.

Many of the entities will have the same value for these Uri's - normalizing them to a separate table reduces burden on storage capacity with request activity logging.

NOTE: There must be a unique index on the *UriHash* and *UriLength* persisted computed columns. This provides an indexed solution for looking up matching uri's during activity logging events.

Column Name	**Data Type**	**Description**
UriId	int identity	Primary key identifier of the Uri entity.
UriString	nvarchar(2083)	The Uri associated with the SiteActivity entity.
UriHash	varbinary(20) computed	Persisted computed hash for the *UriString* column, allowing for indexed lookup (in combination of

		UriLength) and comparison during activity logging events.
UriLength	int computed	Persisted computed length of the *UriString* column, allowing for indexed lookup (in combination with *UriHash*) and comparison during activity logging events.

ErrorLog

Allows logging of any error state or exception that terminates processing of a request, thread, or service, such as code generated exceptions or an invalid context (such as invalid path or query string parameter).

Column Name	**Data Type**	**Description**
ErrorLogId	int identity	Primary key identifier of the ErrorLog entity.
ErrorCode	int	Error code identifying the type of error that stopped processing of the request. Examples might be context validation errors (invalid identifier or external identifier), malformed paths, not found errors (context identifiers such as taxonomy association id not found in client db), unhandled exceptions, etc. **NOTE:** The available values should be defined in an enumeration, as well as included in a scalar user defined function used as part of a check constraint for this column to ensure data integrity.
ErrorUtcDate	datetime	Utc date and time of the error occurrance.
Priority	int	Priority level of the exception. Inherited from Enterprise Library.
Severity	nvarchar(32)	Severity of the exception. Inherited from Enterprise Library.
Title	nvarchar(256)	Title of the exception handling policy. Inherited from Enterprise Library.
MachineName	nvarchar(32)	Name of the machine reporting the exception. Inherited from Enterprise Library.
AppDomainName	nvarchar(512)	Name of the application domain reporting the exception. Inherited from Enterprise Library.

ProcessID	nvarchar(256)	ID of the process reporting the exception. Inherited from Enterprise Library.
ProcessName	nvarchar(512)	Name of the process (usually executable + path) reporting the exception. Inherited from Enterprise Library.
ThreadName	nvarchar(512)	Name of the thread the exception occurred on. As a thread only has a name if explicitly set, this is usually *NULL*. Inherited from Enterprise Library.
Win32ThreadId	nvarchar(128)	ID of the thread the exception occurred on. Inherited from Enterprise Library.
EventId	int	Event ID of the exception. Equivalent to the event id of an event log entry. Inherited from Enterprise Library.
SiteActivityId	int	If set, identifies which SiteActivity entity the error belongs to.
Message	nvarchar(1500)	Short message describing the error, such as context validation error details. Inherited from Enterprise Library.
FormattedMessage	nvarchar(max)	Error details such as exception message and stack trace. Inherited from Enterprise Library.

ClientRole

Defines all the client specific roles a User can belong to.

Column Name	**Data Type**	**Description**
ClientRoleId	int	Primary key identifier of the ClientRole entity.
Name	nvarchar(100)	Descriptive name of the role. **NOTE:** There must be a unique index or constraint on this column.
Description	nvarchar(400)	Detailed description of the role.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.

ModifiedBy	int	Identity of the User who performed the last update of this entity.

UserClientRole

Associates a ClientRole to a User.

Column Name	Data Type	Description
UserId	int	First part composite key identifying the User the role is associated with.
ClientRoleId	int	Second part composite key identifying the ClientRole the user belongs to.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

CUDHistory

Contains an audit trail of all create, update and delete operations against auditable tables.

NOTE: Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Column Name	Data Type	Description
CUDHistoryId	int identity	Primary key identifier of the CUDHistory entity.
TableName	nvarchar(128)	Name of the table the CUD operation occurred on.
Key	int	Primary key identifier of the row the CUD operation occurred on.
SecondKey	nvarchar(200)	Secondary key identifier of the row the CUD operation occurred on. *NULL* if there is no secondary key.
ThirdKey	nvarchar(200)	Tertiary key identifier of the row the CUD operation occurred on. *NULL* if there is no tertiary key.

CUDType	char(1)	Character representing the type of update: *C* for create, *U* for update, *D* for delete.
UtcCUDDate	datetime	Utc date and time of the update.
BatchId	uniqueidentifier	Unique identifier for the update. Used to identify which rows were part of the same CUD operation. **NOTE:** A new id should be generated at the beginning of any procedure or trigger performing any audit logging and used for each row inserted during that operation.
UserId	int	Identity of the User who performed the CUD operation.

CUDHistoryData

Contains the change data pertaining to a CUDHistory entity.

NOTE: Each auditable table must contain either triggers or gateway CUD procedures that handle inserting appropriate data into the CUDHistory and CUDHistoryData tables.

Column Name	**Data Type**	**Description**
CUDHistoryId	int	First part composite key identifying the CUDHistory entity this change data belongs to.
ColumnName	nvarchar(128)	Second part composite key identifying the column this change data is for.
SqlDbType	int	Type of data stored in the column. Should be one of the values available from the *System.Data.SqlDbType* enumeration..
OldValue	nvarchar(max)	Original value before the CUD operation. Always *NULL* for create operations.
NewValue	nvarchar(max)	New value after the CUD operation. Always *NULL* for delete operations.

Vertical Integration Schema

The tables below define vertical market specific associations, metadata overrides and supplementation, and custom data groupings.

The two main components of the vertical market here are TaxonomyAssociation and DocumentTypeAssociation.

TaxonomyAssociation (and supporting tables) defines the market specific taxonomy hierarchy association and is tied to a Site at the root level.

DocumentTypeAssociation defines what document types are associated at either a Site or TaxonomyAssociation level. Only available documents of these types will be displayed in any TemplatePage.

Table Name	Logged	Description
TaxonomyAssociation	Yes	This table defines any associations to any vertical market specific taxonomy level regardless of hierarchy. In the U.S.A. market this could define a Company (Fund Family), Fund or Class association. A market specific taxonomy entity (such as a specific Fund) can be associated multiple times in this table depending on where that association is being used.
TaxonomyAssociationMetaData	Yes	This table defines additional metadata for any given TaxonomyAssociation entity that is not in the vertical market.
TaxonomyAssociationGroup	Yes	This table defines a client specific grouping of TaxonomyAssociation entities. This is used primarily for grouping collections of TaxonomyAssociation entities (such as funds) in ways that make sense for the various

		product offerings for the client.
TaxonomyAssociationGroupTaxonomyAssociation	Yes	This table defines which TaxonomyAssociation entities are associated with a specific TaxonomyAssociationGroup entity.
TaxonomyAssociationHierachy	Yes	This table defines TaxonomyAssociation entity hierarchy associations that are either not part of the vertical market, or not part of the ingestion process and therefore must be defined defined outside of the market. An example use of this is for “Product Groups” where an ingested fund is really a composite of other funds. The associated funds are not part of the ingestion and are therefore defined in the client database here.
TaxonomyLevelExternalId	Yes	This table defines any external identifier the system needs to recognize for a TaxonomyAssociation entity. There must be a unique index on the *ExternalId* column.
TaxonomyAssociationClientDocument	Yes	This table defines associations between a specific TaxonomyAssociation entity and one or more ClientDocument entities.
TaxonomyAssociationClientDocumentGroup	Yes	This table defines associations between a specific TaxonomyAssociation entity and one or more ClientDocumentGroup entities. An example would be Money Market documents which often contain 6 months past history

		similar to a bank statement. Each month would be considered a client document and part of the client document group.
DocumentTypeAssociation	Yes	This table defines which document types from the vertical market can be viewed. Association can with a Site entity or TaxonomyAssociation entity.
DocumentTypeExternalId	Yes	This table defines any external identifiers the system must recognize for a DocumentType. External identifiers must be unique.
Footnote	Yes	This table defines any footnotes that might exist for a specific TaxonomyAssociation entity or TaxonomyAssociationGroup entity. These footnotes will be compiled with proper annotation on pages displaying an associated entity.
VerticalXmlImport	No	Contains the details of any vertical market related import into the system. Imports are linked to a specific xml export generated based on the import data types to allow for easy rollback of an import.
VerticalXmlExport	No	Contains the details of any vertical market integration related export of from the system. This table is also used to backup vertical market integration data immediately prior to an import.
TaxonomyAssociationGroupClientDocumentGroup		
TaxonomyAssociationGroupClientDocument		
FundDocumentClientDocumentGroup		
FundDocumentClientDocument		

TaxonomyAssociation

This table defines any associations to any vertical market specific taxonomy level regardless of hierarchy.

In the U.S.A. market this could define a Company (Fund Family), Fund or Class association.

A market specific taxonomy entity (such as a specific Fund) can be associated multiple times in this table depending on where that association is being used.

Column Name	**Data Type**	**Description**
TaxonomyAssociationId	int identity	Primary key and identifier for the row.
Level	int	Taxonomy level of the entity. Translates to a specific taxonomy hierarchy in the vertical market database.
TaxonomyId	int	Identity of the taxonomy entity from the vertical market database.
SiteId	int	If set, assigns this item as a top level item for the associated Site. This item will be displayed as a root level item on any Page meant to display Taxonomy entities that does not have a specific context set otherwise.
ParentTaxonomyAssociationId	int	If set, assigns this item as a child to a specific Taxonomy entity. This value should reflect the hierarchy defined in the vertical market. Only set if there is a need to recreate that hierarchy for the current Site.
NameOverride	nvarchar(200)	Name to override the display name for the Taxonomy entity from the vertical market. Leave null to use the default name.
DescriptionOverride	nvarchar(400)	Description to override the description for the Taxonomy entity from the vertical market. Leave null to use the default description.
CssClass	varchar(50)	CSS class name to add to the root element for any Page showing this Taxonomy entity as the root element. This allows custom stylings using the cascading feature of CSS.

UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyAssociationMetaData

This table defines additional metadata for any given TaxonomyAssociation entity that is not in the vertical market.

Column Name	**Data Type**	**Description**
TaxonomyAssociationID	int	First part composite key for the metadata. This identity binds the metadata to the Taxonomy entity.
Key	varchar(20)	Second part composite key for the metadata. This identity is the metadata key for looking up the information.
DataType	int	Enumeration representing the type of data stored for the metadata - can be integer, boolean, datetime, or string.
Order	int	Used for display order of associated metadata.
IntegerValue	int	Holds the value for integer metadata.
BooleanValue	bit	Holds the value for boolean metadata.
DateTimeValue	datetime	Holds the value for datetime metadata.
StringValue	nvarchar(max)	Holds the value for string metadata.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last

		update of this entity.

TaxonomyAssociationGroup

This table defines a client specific grouping of TaxonomyAssociation entities.

This is used primarily for grouping collections of TaxonomyAssociation entities (such as funds) in ways that make sense for the various product offerings for the client.

Column Name	**Data Type**	**Description**
TaxonomyAssociationGroupId	int identity	Primary key and identifier of the row.
Name	nvarchar(100)	Display name of the taxonomy association group.
Description	nvarchar(400)	Description of the taxonomy association group.
SiteId	int	If set, assigns this item as a top level item for the associated Site. This item will be displayed as a root level item on any Page meant to display taxonomy association groups that does not have a specific context set otherwise.
ParentTaxonomyAssociationId	int	If set, assigns this item as a child to a specific taxonomy association group. This value should reflect the hierarchy defined in the vertical market. Only set if there is a need to recreate that hierarchy for the current Site.
ParentTaxonomyAssociationGroupId		If set, assigns this item as a child to another taxonomy association group. Used when this type of nested hierarchy of custom groupings is necessary for the client.
CssClass	varchar(50)	If set, defines a css class name to inject into any element displaying this TaxonomyAssociationGroup entity. Can be used for customization of look and feel.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a

		default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyAssociationGroupTaxonomyAssociation

This table defines which TaxonomyAssociation entities are associated with a specific TaxonomyAssociationGroup entity.

Column Name	**Data Type**	**Description**
TaxonomyAssociationGroupId	int	First part composite key of the association. Defines the taxonomy association group the association belongs to.
TaxonomyAssociationId	int	Second part composite key of the association. Defines the taxonomy association that belongs to the group.
Order	int	Display order for group associations.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyAssociationHierachy

This table defines TaxonomyAssociation entity hierarchy associations that are either not part of the vertical market, or not part of the ingestion process and therefore must be defined defined outside of the market.

An example use of this is for “Product Groups” where an ingested fund is really a composite of other funds. The associated funds are not part of the ingestion and are therefore defined in the client database here.

Column Name	Data Type	Description
ParentTaxonomyAssociationId	int	First part composite key of the hierarchy - defines the parent or primary taxonomy association.
ChildTaxonomyAssociationId	int	Second part composite key of the hierarchy - defines the child or secondary taxonomy association.
RelationshipType	int	Third part composite key of the hierarchy - defines the type of association between parent and child. Example would be a product group fund.
Order	int	Display order of hierarchy associations.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyLevelExternalId

This table defines any external identifier the system needs to recognize for a TaxonomyAssociation entity.

There must be a unique index on the *ExternalId* column.

Column Name	Data Type	Description
Level	int	First part composite key identifying the taxonomy level of the entity.
TaxonomyId	int	Second part composite key identifying the vertical market identifier of the entity.
ExternalId	nvarchar(100)	Third part composite key defining the external identifier for the entity. This value must be unique across all rows (requires a unique index).
IsPrimary	bit	Boolean flag that determines which entry is the primary external id to be used for building links in a page where

		the use external id feature is turned on.
		NOTE: This column should have a default constraint setting the initial value to 0.
		NOTE 2: This column should have a unique filtered index on it that guarantees only one true (1) entry exists for each *Level* and *TaxonomyId* combination. The filter should be *WHERE IsPrimary = 1*. See (https://technet.microsoft.com/en-us/library/cc280372.aspx) for more information on filtered indexes. Something along the lines of this should work:
		`CREATE UNIQUE INDEX`
		`iux_TaxonomyLevelExternalId_Level_TaxonomyId_IsPrimary`
		`ON TaxonomyLevelExternalId`
		`(`
		`        Level,`
		`        TaxonomyId,`
		`        IsPrimary`
		`)`
		`WHERE IsPrimary = 1;`

UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyAssociationClientDocument

This table defines associations between a specific TaxonomyAssociation entity and one or more ClientDocument entities.

Column Name	**Data Type**	**Description**
TaxonomyAssociationId	int	First part composite key defining the taxonomy association.
ClientDocumentId	int	Second part composite key defining the client document associated to the taxonomy association.
Order	int	Display order for client documents associated to the taxonomy association.
UtcModifiedDate	datetime	Utc date and time of last update.

		NOTE: This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

TaxonomyAssociationClientDocumentGroup

This table defines associations between a specific TaxonomyAssociation entity and one or more ClientDocumentGroup entities.

An example would be Money Market documents which often contain 6 months past history similar to a bank statement. Each month would be considered a client document and part of the client document group.

Column Name	**Data Type**	**Description**
TaxonomyAssociationId	int	First part composite key defining the taxonomy association.
ClientDocumentGroupId	int	Second part composite key defining the client document group associated to the taxonomy association.
Order	int	Display order for client document groups associated to the taxonomy association.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

DocumentTypeAssociation

This table defines which DocumentType entities from the vertical market can be viewed. Association can with a Site entity or TaxonomyAssociation entity.

Column Name	**Data Type**	**Description**

DocumentTypeAssociationId	int identity	Primary key identifier for the DocumentType association.
DocumentTypeId	int	Vertical market identifier of the DocumentType.
SiteId	int	If set, defines this association as a global association for the Site. This will apply to any taxonomy association that does not have specific associations.
TaxonomyAssociationId	int	If set, defines this association to a specific taxonomy association. Overrides any site level associations.
Order	int	Display order of the DocumentType entities - defines column order for grids, etc.
HeaderText	nvarchar(100)	If set, defines the text to be used in place of the DocumentType name when the DocumentType is a header to a column in an html table.
LinkText	nvarchar(100)	If set, defines the text to be used in place of the DocumentType name when the DocumentType is a an html link.
DescriptionOverride	nvarchar(400)	If set, defines the text to be used in place of the DocumentType description. The description is used as both the detailed description and tooltip within the various templates..
CssClass	varchar(50)	If set, defines a css class name to inject into any element displaying this DocumentTypeAssociation entity. Can be used for customization of look and feel.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

DocumentTypeExternalId

This table defines any external identifiers the system must recognize for a DocumentType. External identifiers must be unique.

Column Name	Data Type	Description
DocumentTypeId	int	First part composite key identifying the vertical market identifier for the DocumentType.
ExternalId	nvarchar(100)	Second part composite key defining the external identifier for the entity. This value must be unique across all rows (requires a unique index).

IsPrimary	bit	Boolean flag that determines which entry is the primary external id to be used for building links in a page where the use external id feature is turned on.
		NOTE: This column should have a default constraint setting the initial value to 0.
		NOTE 2: This column should have a unique filtered index on it that guarantees only one true (1) entry exists for each *DocumentTypeId* value. The filter should be *WHERE IsPrimary = 1*. See (https://technet.microsoft.com/en-us/library/cc280372.aspx) for more information on filtered indexes. Something along the lines of this should work:
		`CREATE UNIQUE INDEX iux_DocumentTypeExternalId_DocumentTypeId`
		`ON DocumentTypeExternalId`
		`(`
		`        DocumentTypeId,`
		`        IsPrimary`
		`)`
		`WHERE IsPrimary = 1;`

UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

Footnote

This table defines any footnotes that might exist for a specific TaxonomyAssociation entity or TaxonomyAssociationGroup entity.

These footnotes will be compiled with proper annotation on pages displaying an associated entity.

Column Name	Data Type	Description
FootnoteId	int identity	Primary key identifier for the Footnote.
TaxonomyAssociationId	int	If set, defines the taxonomy association this Footnote belongs to.
TaxonomyAssociationGroupId	int	If set, defines the taxonomy association group this Footnote belongs to.
LanguageCulture	varchar(50)	If set, defines the language-culture combination this Footnote is for. If null, defines the default or primary language-culture Footnote.
Text	nvarchar(max)	Text of the Footnote.
Order	int	Order to display associated footnotes.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

VerticalXmlImport

Contains the details of any import of client data into the system. Imports are linked to a specific xml export generated based on the import data types to allow for easy rollback of an import.

Column Name	Data Type	Description
VerticalXmlImportId	int identity	Primary key identity of the VerticalXmlImport entity.
ImportTypes	int	Flag set of types of data imported with this entity. The flag set will be defined in an enumeration with the Flag attribute set. There must be an accompanying scalar UDF that will perform appropriate bitwise operations to determine which

		types are set in this flag value.
ImportXml	xml (schemabound)	Xml data defining the import. Must meet the schema definition for imports (a link will be provided when the schema is defined).
ImportDate	datetime	Date of the xml import.
ImportedBy	int	Identity of the User performing the import.
ExportBackupId	int	Identity of the XmlExport entity created as part of the import process. This entity will be used for any rollback actions of the import (which is an all or nothing action, no partial rollbacks).
ImportDescription	nvarchar(400)	Optional description of the import that the user can provide for identity purposes.

VerticalXmlExport

Contains the details of any export of client data from the system. This table is also used to backup any area of the client data immediately prior to an import.

Column Name	**Data Type**	**Description**
VerticalXmlExportId	int identity	Primary key identity of the VerticalXmlExport entity.
ExportTypes	int	Flag set of types of data exported with this entity. The flag set will be defined in an enumeration with the Flag attribute set. There must be an accompanying scalar UDF that will perform appropriate bitwise operations to determine which types are set in this flag value.
ExportXml	xml (schemabound)	Xml data defining the export. Must meet the schema definition for exports (a link will be provided when the schema is defined).
ExportDate	datetime	Date of the xml export.
ExportedBy	int	Identity of the User performing the export. In the case of a backup generated from an import, this will be set to a system identity.
ExportDescription	nvarchar(400)	Optional description of the import that the user can provide for identity purposes.

		In the case of a backup generated from an import, this will have a specific value that helps identify this entry as a backup.

Template & Design Schema

The tables below define Template associations as well as customizations for display, styling, and common text for the client.

All the customizations in this section are not tied to any vertical market entities at all - this allows for a separation of concern where the Template is not tied to a specific vertical market.

NOTE: Version tables are not logged. There should not ever be an updates to a version table record, only inserts. The version table record is itself an audit trail.

Table Name	Logged	Description
SiteText	Yes	Defines resource text to be used in the Site such as buttons and labels. This text is global to the Template. **NOTE:** There must be a unique index on the following columns: *SiteId*, *ResourceKey*, *LanguageCulture*. **NOTE 2:** This feature can be traced back to a TemplateText entity defined in the system database, identified by the Template associated to the Site identified by the *SiteId*.
SiteTextVersion	No	Contains versioned history (including current value) of text for SiteText entities.
PageText	Yes	Defines larger text blocks specific to pages within the template, such as headers and footers. **NOTE:** There must be a unique index on the following columns: *SiteId*, *PageId*, *ResourceKey*, *LanguageCulture*. **NOTE 2:** This text can be traced back to TemplatePageText entity defined in the system database, identified by the *PageId* as well as the Template associated to the Site identified with the *SiteId*.
PageTextVersion	No	Contains versioned history (including current value) of text for PageText entities.
SiteFeature	Yes	Contains configuration of Template wide features by Site. **NOTE:** This feature can be traced back to a TemplateFeature entity defined in the system database, identified by the Template associated to the Site identified by the *SiteId*.

PageFeature	Yes	Contains configuration of Page specific features by Site. **NOTE:** This feature can be traced back to a TemplatePageFeature entity defined in the system database, identified by the *PageId* as well as the Template associated to the Site identified with the *SiteId*.
SiteNavigation	Yes	Defines navigation xml for menus used throughout the Site. This xml will contain schema that allows insertion of specific menu elements as well as custom links and hierarchy. The known xml elements will be replaced with xml specific to the context when being served, then the final xml will be transformed using Template specific xslt before being injected into the outgoing response. **NOTE:** There must be a unique index on the following columns: *SiteId*, *NavigationKey*, *PageId*, *LanguageCulture*. **NOTE 2:** PageId is nullable - if it is not set the xml is global to the Site. If it is set, the xml is a Page specific override of the global site xml.
SiteNavigationVersion	No	Contains the version history (including current value) of xml for SiteNavigation entities.
PageNavigation	Yes	Defines the navigation xml for menus used on specific pages of a Site. This xml will contain schema that allows insertion of specific menu elements as well as custom links and hierarchy. The known xml elements will be replaced with xml specific to the context when being served, then the final xml will be transformed using a Template and Page specific xslt before being injected into the outgoing response. **NOTE:** There must be a unique index on the following columns: *SiteId*, *PageId*, *NavigationKey*, *LanguageCulture*. **NOTE 2:** This text can be traced back to TemplatePageNavigation entity defined in the system database, identified by the *PageId* as well as the Template associated to the Site identified with the *SiteId*.
PageNavigationVersion	No	Contains the version history (including current value) of xml for PageNavigation entities.

SiteXmlImport	No	Contains the details of any Site related import of client data into the system. Imports are linked to a specific xml export generated based on the import data types to allow for easy rollback of an import.
SiteXmlExport	No	Contains the details of any Site related export of client data from the system. This table is also used to backup Site data immediately prior to an import.
StaticResource	Yes	Contains static files as unstructured data that are referenced in CSS, such as images and fonts.

SiteText

Defines resource text to be used in the Site such as buttons and labels. This text is global to the Template.

NOTE: There must be a unique index on the following columns: *SiteId*, *ResourceKey*, *LanguageCulture*.

NOTE 2: This feature can be traced back to a TemplateText entity defined in the system database, identified by the Template associated to the Site identified by the *SiteId*.

Column Name	Data Type	Description
SiteTextId	int	Primary key identifier for the SiteText entity.
SiteId	int	Identifier or the Site the text belongs to.
ResourceKey	varchar(200)	Resource key used to look up the text.
CurrentVersion	int	Current version of the SiteText entry. Used to identify the text from the SiteTextVersion table. This value defaults to 0.
LanguageCulture	varchar(50)	If set, defines the language-culture combination this text is for. If null, defines the default or primary language-culture text.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

SiteTextVersion

Contains versioned history (including current value) of text for SiteText entities.

Column Name	**Data Type**	**Description**
SiteTextId	int	First part composite key identifying the SiteText entity the text version belongs to.
Version	int	Second part composite key identifying the version number of the text.
Text	nvarchar(max)	Text to be used throughout the Site.
UtcCreateDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*.
CreatedBy	int	Identity of the User who inserted the version record.

PageText

Defines larger text blocks specific to pages within the template, such as headers and footers.

NOTE: There must be a unique index on the following columns: *SiteId*, *PageId*, *ResourceKey*, *LanguageCulture*.

NOTE 2: This text can be traced back to TemplatePageText entity defined in the system database, identified by the *PageId* as well as the Template associated to the Site identified with the *SiteId*.

Column Name	**Data Type**	**Description**
PageTextId	int identity	Primary key identifier for the PageText entity.
SiteId	int	Identifier of the Site the text belongs to.
PageId	int	Identifier of the Page the text belongs to.
ResourceKey	varchar(200)	Resource key used to look up the text.
CurrentVersion		Current version of the PageText entry. Used to identify the text from the PageTextVersion table. This value defaults to 0
LanguageCulture	varchar(50)	If set, defines the language-culture combination this text is for. If null, defines the default or primary language-culture text.

UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

PageTextVersion

Contains versioned history (including current value) of text for PageText entities.

Column Name	Data Type	Description
PageTextId	int	First part composite key identifying the PageText entity the text version belongs to.
Version	int	Second part composite key identifying the version number of the text.
Text	nvarchar(max)	Text to be used on the associated Page for the Site associated to the PageText entity.
UtcCreateDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*.
CreatedBy	int	Identity of the User who inserted the version record.

SiteFeature

Contains configuration of Template wide features by Site.

NOTE: This feature can be traced back to a TemplateFeature entity defined in the system database, identified by the Template associated to the Site identified by the *SiteId*.

Column Name	Data Type	Description
SiteId	int	First part composite key identifying the Site this feature is configured for.
Key	varchar(200)	Second part composite key identifying the Feature by a meaningful name.
FeatureMode	int	Integer representing an enumeration defining the mode of

		the Feature (enumeration is feature specific). The default should be 0, which should mean “Disabled” for any feature.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

PageFeature

Contains configuration of Page specific features by Site.

NOTE: This feature can be traced back to a TemplatePageFeature entity defined in the system database, identified by the *PageId* as well as the Template associated to the Site identified with the *SiteId*.

Column Name	**Data Type**	**Description**
SiteId	int	First part composite key identifying the Site this feature is configured for.
PageId	int	Second part composite key identifying the Page this feature belongs to.
Key	varchar(200)	Third part composite key identifying the Feature by a meaningful name.
FeatureMode	int	Integer representing an enumeration defining the mode of the Feature. The default should be 0, which should mean “Disabled” for any feature.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

SiteNavigation

Defines navigation xml for menus used throughout the Site.

This xml will contain schema that allows insertion of specific menu elements as well as custom links and hierarchy.

The known xml elements will be replaced with xml specific to the context when being served, then the final xml will be transformed using Template specific xslt before being injected into the outgoing response.

NOTE: There must be a unique index on the following columns: *SiteId*, *NavigationKey*, *PageId*, *LanguageCulture*.

NOTE 2: PageId is nullable - if it is not set the xml is global to the Site. If it is set, the xml is a Page specific override of the global site xml.

Column Name	**Data Type**	**Description**
SiteNavigationId	int	Primary key identifier for the SiteNavigation entity.
SiteId	int	Identifier of the Site the xml navigation belongs to.
NavigationKey	varchar(200)	Key used to look up the navigation xml.
PageId	int	If set, provides a Page specific override of the navigation xml.
LanguageCulture	varchar(50)	If set, defines the language-culture combination this xml menu is for. If null, defines the default or primary language-culture text.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

SiteNavigationVersion

Contains the version history (including current value) of xml for SiteNavigation entities.

Column Name	**Data Type**	**Description**

SiteNavigationId	int	First part composite key identifying the SiteNavigation entity the version belongs to.
Version	int	Second part composite key identifying the version number of the navigation xml.
NavigationXml	xml (schemabound)	Contains the xml defining the navigation menu. Can be a combination of elements defining known menu features as well as elements defining custom menu items and hierarchies.
UtcCreateDate	datetime	Utc date and time of creation. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*.
CreatedBy	int	Identity of the User who created the version.

PageNavigation

Defines the navigation xml for menus used on specific pages of a Site.

This xml will contain schema that allows insertion of specific menu elements as well as custom links and hierarchy.

The known xml elements will be replaced with xml specific to the context when being served, then the final xml will be transformed using a Template and Page specific xslt before being injected into the outgoing response.

NOTE: There must be a unique index on the following columns: *SiteId*, *PageId*, *NavigationKey*, *LanguageCulture*.

NOTE 2: This text can be traced back to TemplatePageNavigation entity defined in the system database, identified by the *PageId* as well as the Template associated to the Site identified with the *SiteId*.

Column Name	**Data Type**	**Description**
PageNavigationId	int	Primary key identifier for the PageNavigation entity.
SiteId	int	Identifier of the Site the xml navigation belongs to.
PageId	int	Identifier of the Page the xml navigation belongs to.
NavigationKey	varchar(200)	Key used to look up the navigation xml.

LanguageCulture	varchar(50)	If set, defines the language-culture combination this xml menu is for. If null, defines the default or primary language-culture text.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who performed the last update of this entity.

PageNavigationVersion

Contains the version history (including current value) of xml for PageNavigation entities.

Column Name	Data Type	Description
PageNavigationId	int	First part composite key identifying the PageNavigation entity the version belongs to.
Version	int	Second part composite key identifying the version number of the navigation xml.
NavigationXml	xml (schemabound)	Contains the xml defining the navigation menu. Can be a combination of elements defining known menu features as well as elements defining custom menu items and hierarchies.
UtcCreateDate	datetime	Utc date and time of creation. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*.
CreatedBy	int	Identity of the User who created this version.

SiteXmlImport

Contains the details of any Site related import of client data into the system. Imports are linked to a specific xml export generated based on the import data types to allow for easy rollback of an import.

Column Name	Data Type	Description

SiteXmlImportId	int identity	Primary key identity of the SiteXmlImport entity.
ImportTypes	int	Flag set of types of data imported with this entity. The flag set will be defined in an enumeration with the Flag attribute set. There must be an accompanying scalar UDF that will perform appropriate bitwise operations to determine which types are set in this flag value.
ImportXml	xml (schemabound)	Xml data defining the import. Must meet the schema definition for imports (a link will be provided when the schema is defined).
ImportDate	datetime	Date of the xml import.
ImportedBy	int	Identity of the User performing the import.
ExportBackupId	int	Identity of the SiteXmlExport entity created as part of the import process. This entity will be used for any rollback actions of the import (which is an all or nothing action, no partial rollbacks).
ImportDescription	nvarchar(400)	Optional description of the import that the user can provide for identity purposes.

SiteXmlExport

Contains the details of any Site related export of client data from the system. This table is also used to backup Site data immediately prior to an import.

Column Name	**Data Type**	**Description**
SiteXmlExportId	int identity	Primary key identity of the SiteXmlExport entity.
ExportTypes	int	Flag set of types of data exported with this entity. The flag set will be defined in an enumeration with the Flag attribute set. There must be an accompanying scalar UDF that will perform appropriate bitwise operations to determine which types are set in this flag value.
ExportXml	xml (schemabound)	Xml data defining the export. Must meet the schema definition for exports (a link will be provided when the schema is defined).
ExportDate	datetime	Date of the xml export.

ExportedBy	int	Identity of the User performing the export. In the case of a backup generated from an import, this will be set to a system identity.
ExportDescription	nvarchar(400)	Optional description of the import that the user can provide for identity purposes. In the case of a backup generated from an import, this will have a specific value that helps identify this entry as a backup.

StaticResource

Contains static files as unstructured data that are referenced in CSS, such as images and fonts.

Column Name	**Data Type**	**Description**
StaticResourceId	int identity	Primary key identity for the StaticResource entity.
FileName	nvarchar(260)	File name of the static resource.
Size	int	Size in bytes of the static resource.
MimeType	varchar(127)	Mime type of the static resource.
Data	varbinary(max)	Actual binary content of the static resource.
UtcModifiedDate	datetime	Utc date and time of last update. **NOTE:** This column should have a default constraint setting the initial value to to *GETUTCDATE()*. It should be updated to *GETUTCDATE*() on any data modification, perhaps by a trigger.
ModifiedBy	int	Identity of the User who last performed the last update on the entity.