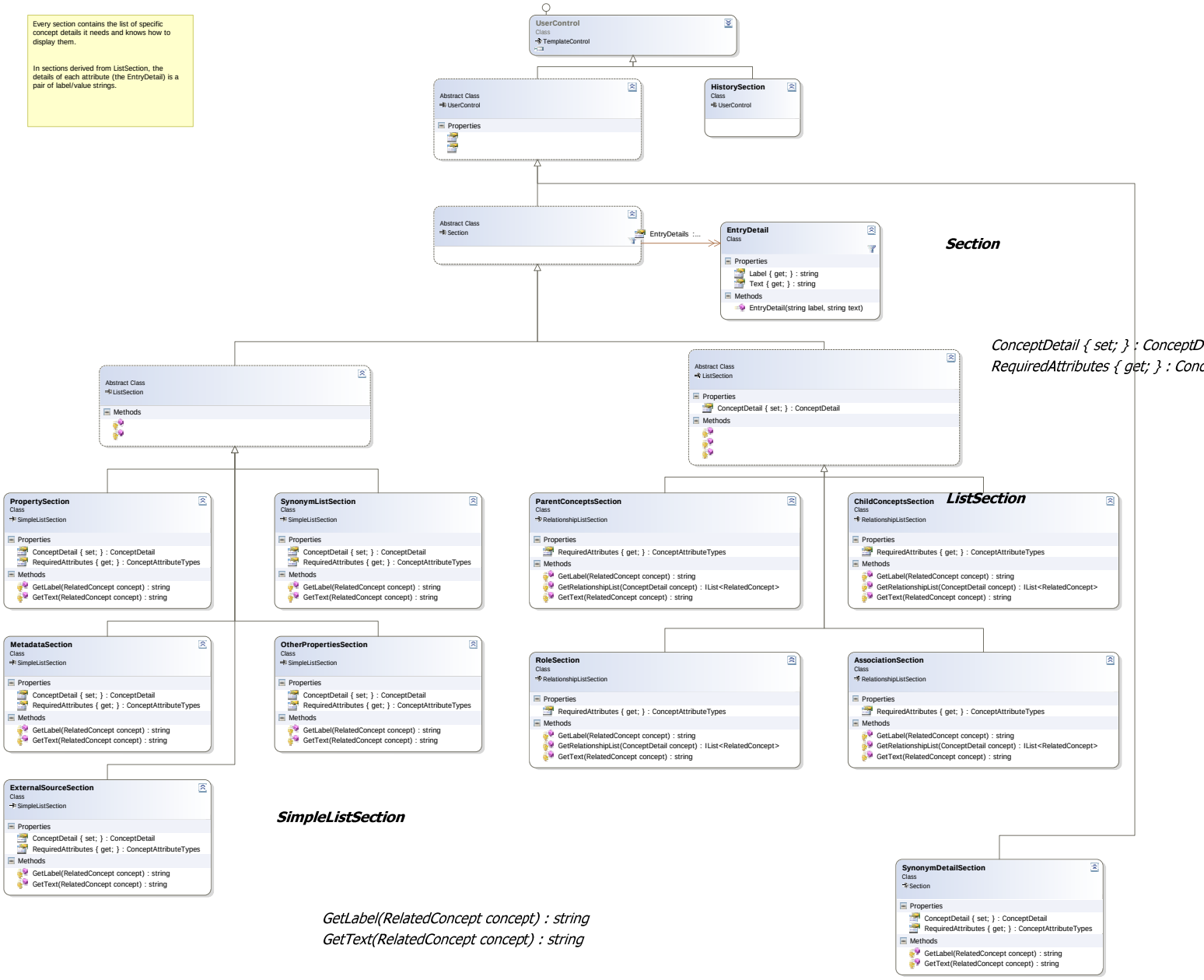
Every page contains a list of sections.
The page knows the order in which the sections are displayed.
Each section knows what concept attributes it needs and how to display them.
The section manager gathers the list of attributes needed by all the sections in the page and uses the ConceptManager to retrieve a concept with all those attributes populated.
IHttpHandler
Page
Class
TemplateControl
ConceptDetailPage
Class
Page
Methods
Display(ConceptDetail concept) : void
_sectionManager
SectionManager
Class
Properties
ConceptAttributeList { get; set; } : ConceptAttributeTypes
Methods
AddSection(Section section) : void
SectionList : List<Section>
Abstract Class
UserControl
Properties
RelationshipsPage
Class
ConceptDetailPage
Properties
AssociationSection { get; } : AssociationSection
ChildConceptsSection { get; } : ExternalSourceSection
ParentConceptsSection { get; } : ParentConceptsSection
RoleSection { get; } : RoleSection
SynonymDetailsPage
Class
ConceptDetailPage
Properties
SynonymDetailSection { get; } : SynonymDetailSection
TermsAndPropertiesPage
Class
ConceptDetailPage
Properties
ExternalSourceSection { get; } : ChildConceptsSection
MetaDataSection { get; } : MetadataSection
OtherPropertiesSection { get; } : OtherPropertiesSection
PropertySection { get; } : PropertySection
SynonymListSection { get; } : SynonymListSection
ViewAllPage
Class
ConceptDetailPage
Properties
AssociationSection { get; } : AssociationSection
ChildConceptsSection { get; } : ExternalSourceSection
ExternalSourceSection { get; } : ChildConceptsSection
MetaDataSection { get; } : MetadataSection
OtherPropertiesSection { get; } : OtherPropertiesSection
ParentConceptsSection { get; } : ParentConceptsSection
PropertySection { get; } : PropertySection
RoleSection { get; } : RoleSection
SynonymDetailSection { get; } : SynonymDetailSection
SynonymListSection { get; } : SynonymListSection
HistoryPage
Class
ConceptDetailPage
Properties
HistorySection { get; } : HistorySection


Every section contains the list of specific concept details it needs and knows how to display them.
In sections derived from ListSection, the details of each attribute (the EntryDetail) is a pair of label/value strings.
UserControl
Class
TemplateControl
Abstract Class
UserControl
Properties
HistorySection
Class
UserControl
Abstract Class
Section
EntryDetails :...
EntryDetail
Class
Properties
Label { get; } : string
Text { get; } : string
Methods
EntryDetail(string label, string text)
Section
ConceptDetail { set; } : ConceptD
RequiredAttributes { get; } : Conc
Abstract Class
ListSection
Methods
Abstract Class
ListSection
Properties
ConceptDetail { set; } : ConceptDetail
Methods
PropertySection
Class
SimpleListSection
Properties
ConceptDetail { set; } : ConceptDetail
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string
SynonymListSection
Class
SimpleListSection
Properties
ConceptDetail { set; } : ConceptDetail
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string
ParentConceptsSection
Class
RelationshipListSection
Properties
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetRelationshipList(ConceptDetail concept) : IList<RelatedConcept>
GetText(RelatedConcept concept) : string
ChildConceptsSection
Class
RelationshipListSection
Properties
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetRelationshipList(ConceptDetail concept) : IList<RelatedConcept>
GetText(RelatedConcept concept) : string
ListSection
MetadataSection
Class
SimpleListSection
Properties
ConceptDetail { set; } : ConceptDetail
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string
OtherPropertiesSection
Class
SimpleListSection
Properties
ConceptDetail { set; } : ConceptDetail
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string
RoleSection
Class
RelationshipListSection
Properties
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetRelationshipList(ConceptDetail concept) : IList<RelatedConcept>
GetText(RelatedConcept concept) : string
AssociationSection
Class
RelationshipListSection
Properties
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetRelationshipList(ConceptDetail concept) : IList<RelatedConcept>
GetText(RelatedConcept concept) : string
ExternalSourceSection
Class
SimpleListSection
Properties
ConceptDetail { set; } : ConceptDetail
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string
SimpleListSection
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string
SynonymDetailSection
Class
Section
Properties
ConceptDetail { set; } : ConceptDetail
RequiredAttributes { get; } : ConceptAttributeTypes
Methods
GetLabel(RelatedConcept concept) : string
GetText(RelatedConcept concept) : string

ConceptManager
Static Class

Methods

- FilterExternalCodes(ConceptDetail concept) : IList<Property>
- LoadBaseConceptList(int conceptId) : IList<ConceptInfo>
- LoadChildConceptList(int conceptId) : IList<ConceptInfo>
- LoadConceptDetail(int conceptId, ConceptAttributeTypes details) : ConceptDetail
- LoadExpandedTree() : void
- LoadParentConceptList(int conceptId) : IList<ConceptInfo>
- LoadTree() : void
- Search(string searchFor, SearchType searchType, int offset, int maxResults) : IList<ConceptInfo>

ConceptAttributeTypes
Enum

- None
- ConceptId
- ConceptCode
- DefinitionList
- PreferredName
- DisplayName
- MetathesaurusCui
- SynonymList
- SynonymDetailList
- ExternalCodeList
- PropertyList
- MetadataList
- ParentConceptList
- ChildConceptList
- RoleList
- AssociationList
- HistoryList

SearchType
Enum

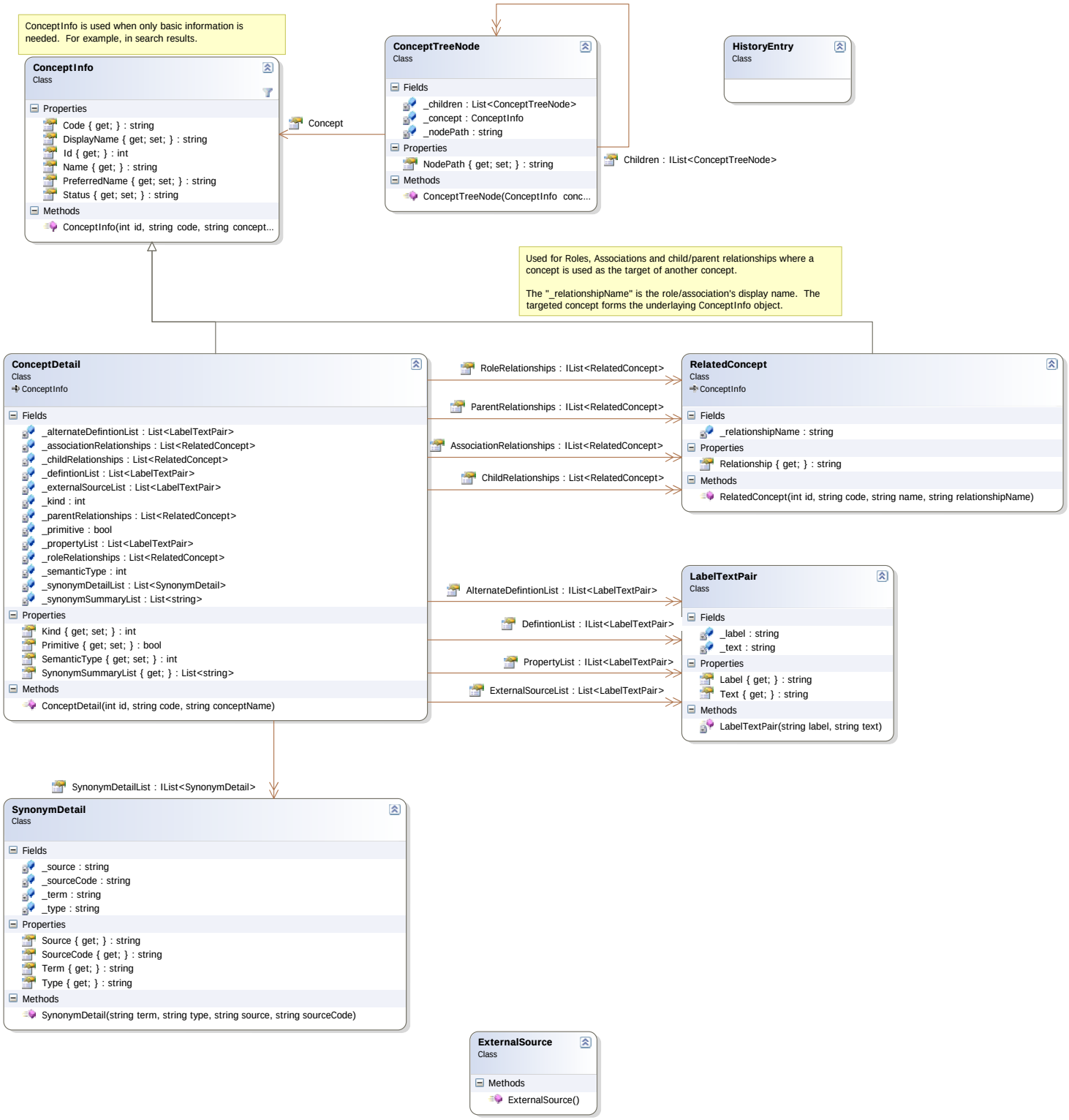
- BeginsWith
- Contains
- ExactMatch

Internal classes, only callable by ConceptManager

ConceptQuery
Static Class

Methods

- GetChildConceptList(int conceptId) : DataSet
- GetConceptDetail(int conceptId, ConceptAttributeTypes details) : DataSet
- GetSuperConceptList() : DataSet
- Search(string searchFor, SearchType searchType, int offset, int maxResults) : DataSet


ConceptInfo is used when only basic information is needed. For example, in search results.
ConceptInfo
Class
Properties
Code { get; } : string
DisplayName { get; set; } : string
Id { get; } : int
Name { get; } : string
PreferredName { get; set; } : string
Status { get; set; } : string
Methods
ConceptInfo(int id, string code, string concept...
Concept
ConceptTreeNode
Class
Fields
_children : List<ConceptTreeNode>
_concept : ConceptInfo
_nodePath : string
Properties
NodePath { get; set; } : string
Methods
ConceptTreeNode(ConceptInfo conc...
Children : IList<ConceptTreeNode>
HistoryEntry
Class
Used for Roles, Associations and child/parent relationships where a concept is used as the target of another concept.
The "_relationshipName" is the role/association's display name. The targeted concept forms the underlaying ConceptInfo object.
ConceptDetail
Class
ConceptInfo
Fields
_alternateDefintionList : List<LabelTextPair>
_associationRelationships : List<RelatedConcept>
_childRelationships : List<RelatedConcept>
_defintionList : List<LabelTextPair>
_externalSourceList : List<LabelTextPair>
_kind : int
_parentRelationships : List<RelatedConcept>
_primitive : bool
_propertyList : List<LabelTextPair>
_roleRelationships : List<RelatedConcept>
_semanticType : int
_synonymDetailList : List<SynonymDetail>
_synonymSummaryList : List<string>
Properties
Kind { get; set; } : int
Primitive { get; set; } : bool
SemanticType { get; set; } : int
SynonymSummaryList { get; } : List<string>
Methods
ConceptDetail(int id, string code, string conceptName)
RoleRelationships : IList<RelatedConcept>
ParentRelationships : IList<RelatedConcept>
AssociationRelationships : IList<RelatedConcept>
ChildRelationships : List<RelatedConcept>
RelatedConcept
Class
ConceptInfo
Fields
_relationshipName : string
Properties
Relationship { get; } : string
Methods
RelatedConcept(int id, string code, string name, string relationshipName)
AlternateDefintionList : IList<LabelTextPair>
DefintionList : IList<LabelTextPair>
PropertyList : IList<LabelTextPair>
ExternalSourceList : IList<LabelTextPair>
LabelTextPair
Class
Fields
_label : string
_text : string
Properties
Label { get; } : string
Text { get; } : string
Methods
LabelTextPair(string label, string text)
SynonymDetailList : IList<SynonymDetail>
SynonymDetail
Class
Fields
_source : string
_sourceCode : string
_term : string
_type : string
Properties
Source { get; } : string
SourceCode { get; } : string
Term { get; } : string
Type { get; } : string
Methods
SynonymDetail(string term, string type, string source, string sourceCode)
ExternalSource
Class
Methods
ExternalSource()