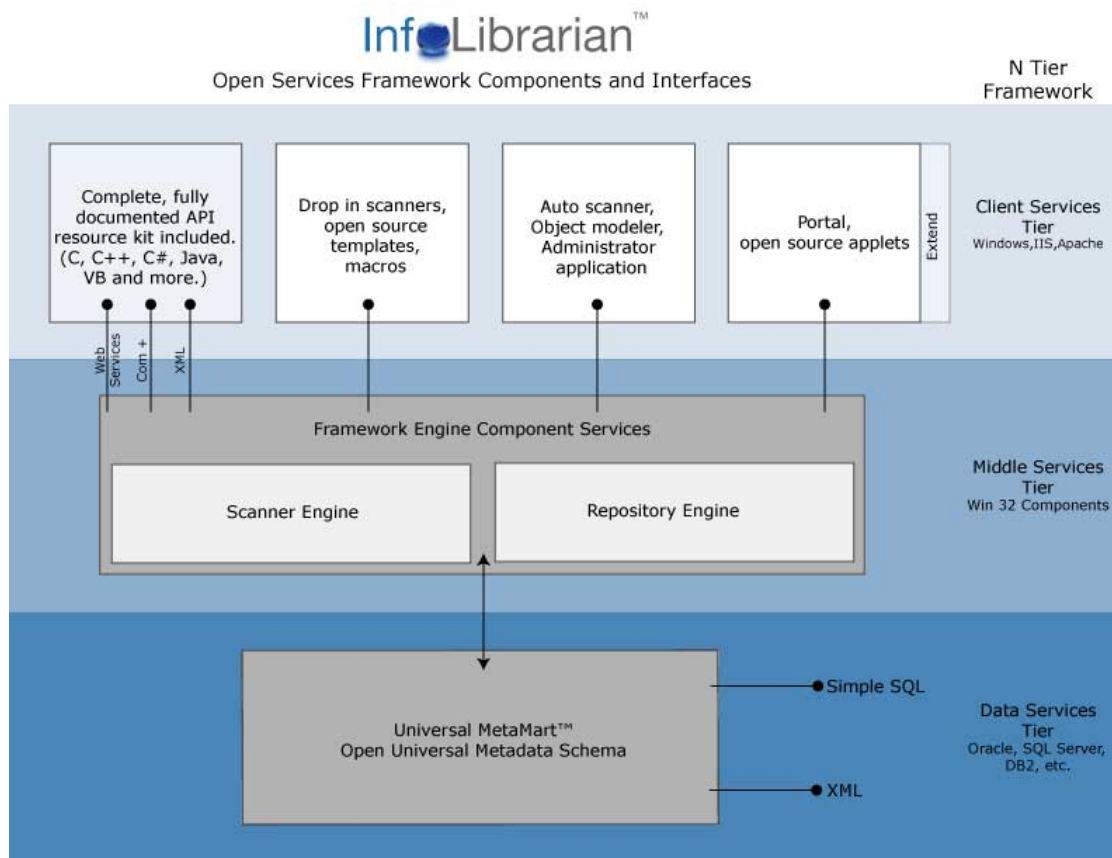


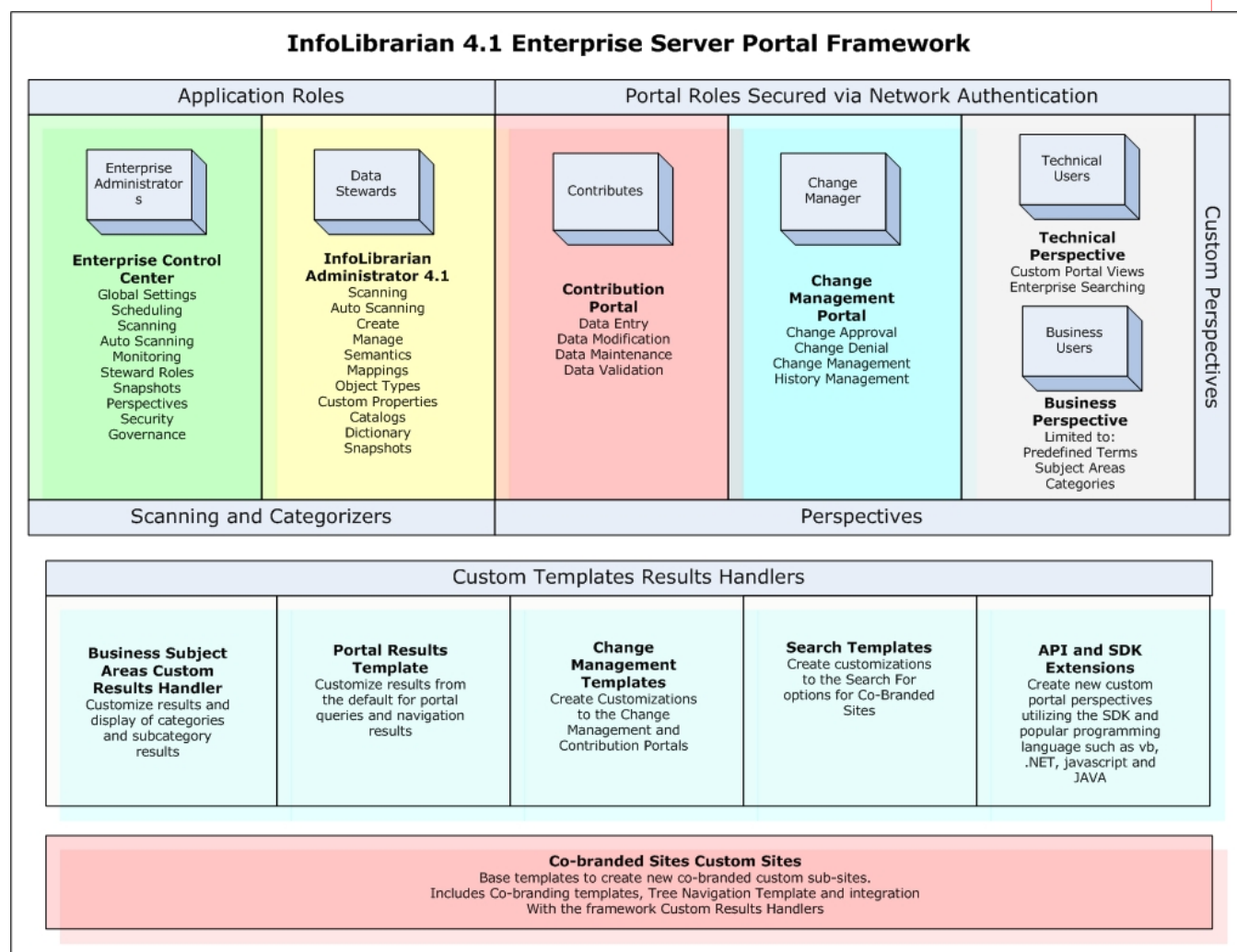
InfoLibrarian™

Metadata Framework User Guide



The Portal Framework

Portal customizations are accomplished by creating a new site from the templates provided. This provides the functionality we deliver out of the box, but affords you the ability to tailor, customize and tweak to your specific business needs. Most of the portal customization would be web development (Interface work) with some SQL Custom queries being developed to provide the data from the repository. This is not an extremely difficult task, but some training on the framework is recommended. We can and often do help companies with a prototype of a custom business portal to provide a starting point that addresses the main reporting needed and provides a good venue for cross training from us to you, that you may comfortably customize on your own moving forward.



PORTAL PERSPECTIVES

Technical Portal

<http://servername/infolib/default.htm> = Technical portal via default.htm

Control Center

<http://servername/infolib/administration> = Enterprise Control Center requires Network authentication setup on IIS virtual.

Default Business Portal

<http://servername/infolib/business> = The default business portal... less technical filtered searches and subject area navigation.

The Key difference between the business portal and the technical portal is as follows:

- The Business Portal filters object types that are searched based on the portal settings in the Control Center.
- The default view for the business portal displays Subject Areas.
- When a Business Term is mapped to a DB Column ... it is displayed as follows:
Where used.

Custom Perspectives

<http://servername/infolib/YourCustomPerspective> = Co branded business portal for customization.

Change Management

<http://servername/infolib/contribution> = Business Users Contribution (Metadata Edit) Portal. Authentication by Windows Users/Groups setup on IIS

<http://servername/infolib/mgmt> = Change managers portal for approval or denial of changes. Authentication by Windows Users/Groups setup on IIS

Portal Documentation

- Control Center portal pages themselves (Inline Documentation) click the +Links for documentation
- The Custom Templates folder on the server. (More about customization)
- Training and consulting.

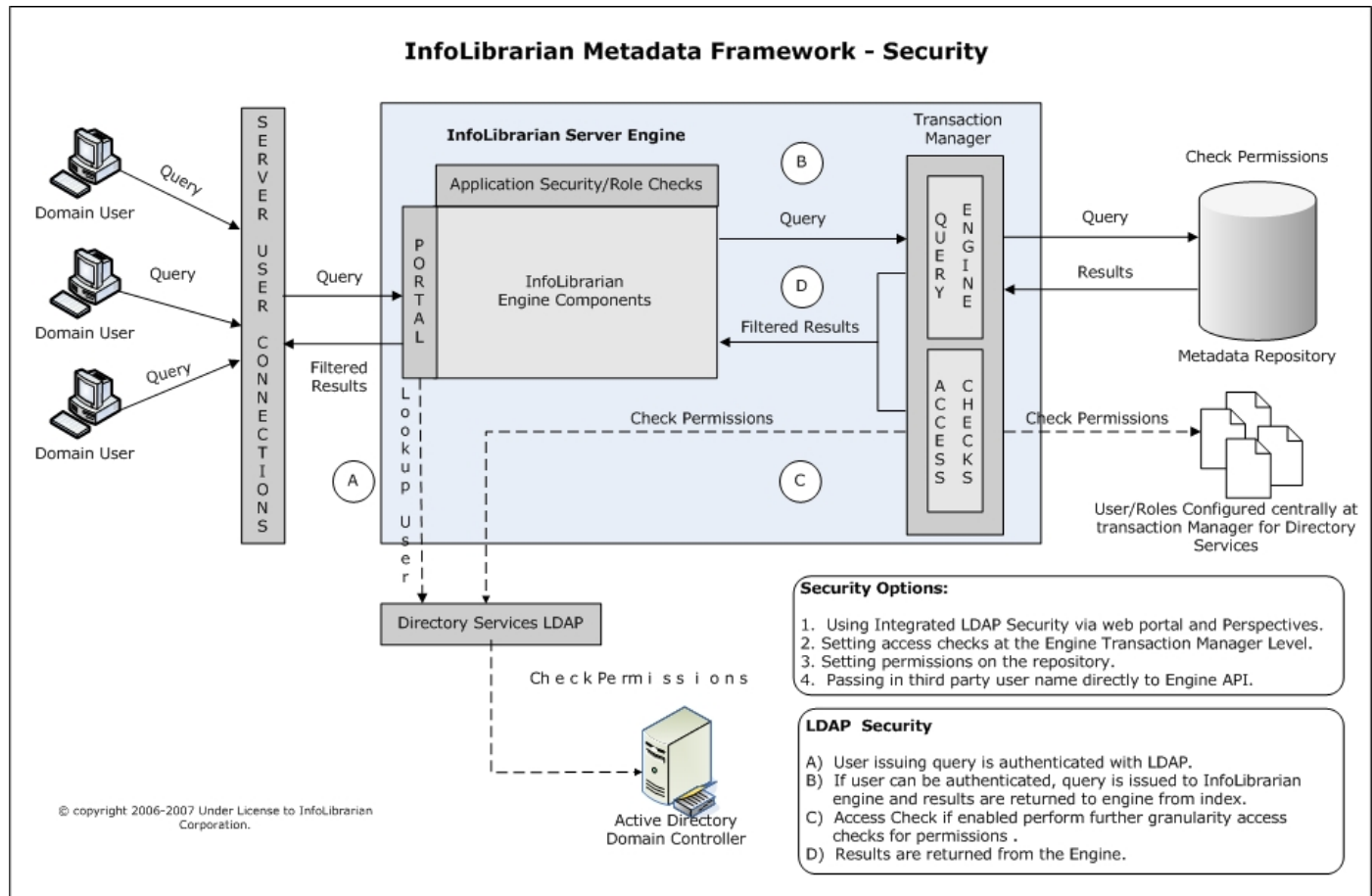
Portal Templates Language

The portal templates are provided in classic ASP and JavaScript. You may recreate new templates in virtually any web language as desired or work with the existing templates as provided.

Custom Perspectives provides the following:

1. Ability to Create and Co brand a specific view of your metadata for a particular user group.
2. Ability to Secure the perspective based on the group accessing that view with web authentication.
3. Ability to customize subject areas to meet specific display and navigation requirements.

Server Framework - Security



Default Portal Custom Templates – Simple Customization Method

Portal Settings

Global Portal Settings | Search For Settings | Search Results Settings | Business Subject Areas | Lineage Analysis Settings | Custom Portal Perspectives

Global Portal Settings

Search Service Settings
Enter the Server Name or IP Address to the InfoLibrarian Search Server that will be configured as the primary Search Service provider for the portal. This setting will direct search queries using "Document/Text" option to an InfoLibrarian Search Server Indexes containing crawls of files, web sites and databases including the Meta-Mart.

Regional Settings
Choose the regional settings... will format dates and times to your regional preference.
NOTE: Regional settings set here on the portal and for the server should be the same as for all clients running scans. This will ensure that all last modified, snapshots and log timestamps will be formatted consistently when stored in the repository and displayed in the portal.

Custom Template Language Extension
Enter the extension for the custom templates web page programming language ie:(asp,jsp,aspx).

Contribution Deleted Indicator
When deleting text fields values, this value will be used. Note: The repository does not support null or blank values. This indicator will be used when users delete text property values.

Setting	Value
InfoLibrarian Search Service. (Document/Database and Text Searching)	ILSUPPORT01
Regional Settings (Date-Time Formats for metrics)	General Unicode (US English)
Custom Template Language Extension (.asp .aspx .jsp .java etc...)	.asp
Contribution Deleted Indicator (default: Deleted)	deleted

Update

Control Center Default Portal Custom Handlers

Change the options found on this page as required. You may also create custom handlers for various sections of portal. Copy and modify custom handlers to meets your particular needs. Each tab on this page represents a section that may be customized by either changing behavior settings or by creating a custom handler that may be coded to override the default handlers. Each tab corresponds to a particular functional area of the portal. Follow the directions on these screens to apply the desired custom template as needed to over-ride the default behavior.

Note: these are global settings that will affect all portal URLs. Custom Templates only over-ride the default portal views.

See Also: Custom Perspectives for more flexibility and customization features.

How Upgrades Affect Customization

Provided that you work within the guidelines of the portal framework, Upgrades will not affect your custom portal templates or perspectives. The core portal framework provides the mechanics from which you can inherit functionalities without having to re-implement core functionalities while giving you the ability to create custom pages.

Sections of the Portal

Search For area

Corresponding settings and template handler.

Global Portal Settings
Search For Settings
Search Results Settings
Business Subject Areas
Lineage Analysis Settings
Custom Portal Perspectives

Search For Handler and Settings

Custom Template
Choose the default, or you may specify a custom results handler for customization of the search for section of the portal. A custom template may be coded to override any settings on this page.
Copy the default template located here "[Infolib/CustomTemplates/SearchTemplate](#)" to create a new template that can be selected here.

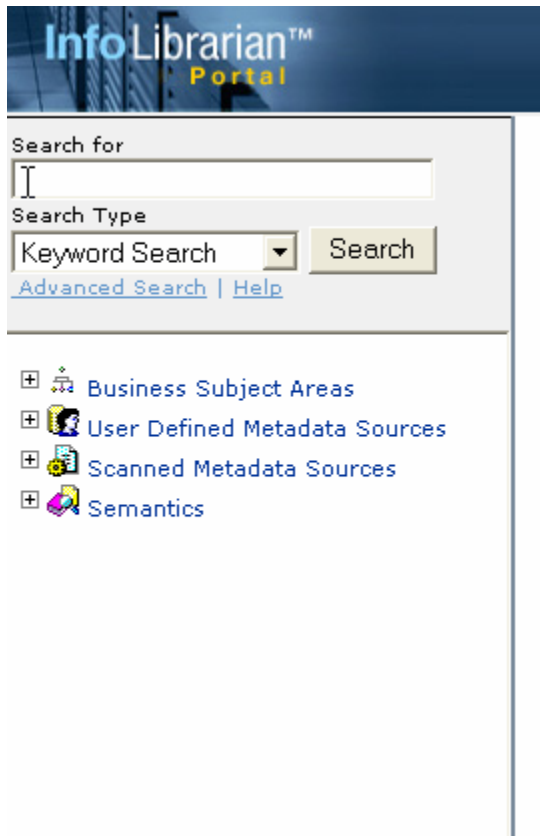
Search Options Checkboxes
Enable or disable various search options presented in the "Search For" navigation.
Choose the default template, or you may specify a custom search handler for customization of the search and results.

Business Object Types List
Specify a list of object types that represents all the object types that will be searched via the "Business Search Option". This comma delimited list will typically represent subject areas, terms and rules.

Setting	Value
Search Option - Keyword Search (Standard Keyword Search)	☒
Search Option - Impact Analysis (Impact Analysis)	☒
Search Option - Documents/Text (Documents/Text Requires InfoLibrarian Search Service)	☒
Search Option - Business Terms (Searches Business Terms)	☒
Search Option - Business Rules (Searches Business Rules)	☒
Search Option - Business Search (Searches Subject Areas/Terms and Rules)	☒
Business Repository Object Types List (Defaults: 29,28,26,8,9)	29,28,26,8,9,33556,33561,33560
Specify A Custom Template (Custom templates may be coded to override all settings on this page)	Default Template

[Repository Object Types](#) Click this link to view the repository object types list.

Left Navigation Tree



The screenshot displays the InfoLibrarian Portal search interface. At the top, there is a header with the 'InfoLibrarian™ Portal' logo. Below the header, there is a search section with a 'Search for' text input field, a 'Search Type' dropdown menu set to 'Keyword Search', and a 'Search' button. Below the search section, there is a left navigation tree with four items, each preceded by a plus icon in a square box:

- Business Subject Areas
- User Defined Metadata Sources
- Scanned Metadata Sources
- Semantics

Customization of the Left Navigation Tree

Customization of the left navigation tree requires a new portal perspective or co branded portal be created. The Toc source can then be modified to limit, filter or change the behavior of the navigation tree.

Technical Results Pane – Note the indicated “Default Metadata Results Handler”

Search for: orders

Search Type: Keyword Search

Search

Business Subject Areas

User Defined Metadata Sources

Scanned Metadata Sources

Semantics

Default Metadata Results Handler

Print this page

Keyword Search Results

orders

TIPS: Search Results Navigation

DBMS Stored Procedure(s) (8 results)

Object Name	Data Source	Catalog	Parent
cust.orders	SQLServerDemo	Northwind	Northwind
invoices.filter	Northwind Traders Sample	Northwind	Northwind
orders.qry	Northwind Traders Sample	Northwind	Northwind
quarterly.orders by product	Northwind Traders Sample	Northwind	Northwind
invoices.filter	TESTAUTO	TESTAUTO	TESTAUTO
orders.qry	TESTAUTO	TESTAUTO	TESTAUTO
quarterly.orders by product	TESTAUTO	TESTAUTO	TESTAUTO
cust.orders	SQLServerDemo	Northwind	Northwind

DBMS View(s) (2 results)

Object Name	Data Source	Catalog	Parent
orders.qry	SQLServerDemo	Northwind	Northwind
quarterly.orders	SQLServerDemo	Northwind	Northwind

DBMS Table(s) (10 results)

Object Name	Data Source	Catalog	Parent
orders	infolib	DB1	DB1

Corresponding settings and results handler template control.

Global Portal Settings | Search For Settings | Search Results Settings | Business Subject Areas | Lineage Analysis Settings | Custom Portal Perspectives

General Search Results Settings

Advanced Search Max Results Threshold

Number of results to display without warnings. This setting will determine how many results are processed before a warning is given to the user. (User can then proceed to process the results or abort) This setting is designed to prevent long running processing of results page generation. Use this setting to limit the number of results displayed when performing advanced searches and queries.

Max Results Display Threshold

Number of results to display without warnings. This setting will determine how many results are processed before a warning is given to the user. (User can then proceed to process the results or abort) This setting is designed to prevent long running processing of results page generation. Use this setting to limit the number of results displayed when expanding catalog objects tree items and keyword search results per category.

Custom Template

Choose the default, or you may specify a custom results handler for customization of the standard metadata results pages. Copy the default template located here "Infolib/CustomTemplates/PortalResults" to create a new template that can be selected here.

Setting	Value
Advanced Search Max Results (Throttles Advanced search results)	500
Max Results Display Threshold (Throttles Keyword search results)	500
Portal Metadata Results Template (Use default or choose custom template from list)	Default Template

Subject Area Results Pane– Note the indicated “Default Subject Area Handler”

Default Subject Area Handler

[Print this page](#) 

 **TIPS: Subject Areas Sub-Categories and Navigation**

Category - Documentation

 Enterprise Search

 Metadata

 Metadata Repository

Description: This is a demo subject area that was created to manage some documentation.

Comments: This subject area category contains some categorized PDF Documents pertaining to the InfoLibrarian Solution. These documents are also published on our website. [InfoLibrarian Corporation Website](#)

The Manner in which all these types of results are displayed can be customized by modifying the appropriate template. This gives you full control over how the results here are displayed.

Documentation Business Terms

Object Name	Object Type	Business Description
 Product PDF Documents	Business Term	Product PDF Documents

 **See Also: Cross Reference/Mappings**

Object Name	Object Type	Description	Data Source	Catalog	Parent	Association Type
 InfoLibrarian4.1 PDF Documentation	Document_Mapping_Object	Mapping Object for Documentation	InfoLibrarianObjects	Mapping Objects	Mapping Objects	Destination Association

 **Linked Documents**

Document Title	Business Description
 ScannerArchitecture	Scanner Architecture Diagram
 PortalFramework	Portal Framework Diagram
 Metadata_Security_Framework	Metadata Security Framework Diagram
 Metadata_Semantics	Metadata Semantics Diagram
 InfoLibrarian Solution Components	InfoLibrarian Solution Components Diagram

[Global Portal Settings](#)
[Search For Settings](#)
[Search Results Settings](#)
[Business Subject Areas](#)
[Lineage Analysis Settings](#)
[Custom Portal Perspectives](#)

Business Subject Area Handler and Custom Settings

Custom Template
Choose the default, or you may specify a custom results handler for customization of the Subject Area Results Pages. Custom templates may be coded to override all settings on this page.
Copy the default template located here "[Infolib/CustomTemplates/SubjectArea](#)" to create a new template that can be selected here.

Show Card View
This option enables the Where Used template when terms are mapped to columns.
Un-checking this option will utilize Portal Results Template.

Max Results per Subject Area Category
Switch to list view in portal Subject Area sub-categories when results exceed this setting. Set to 0 to display all subcategories as lists. By default, portal sub-categories are displayed graphically.

Reports Types List
Specify a comma delimited list of object types that represent Repository Reports. This setting will determine the types of objects that are returned when choosing "Show Reports" in the Business Portal. This setting only applies if the "Show Card View" option is turned on.

Setting	Value
 Subject Area Results Template (Use default or choose custom template from list)	Default Template 
 Show Card View (When terms mapped to columns Default=checked)	☒
 Max Results Per Category (Switch to List View in Sub-Categories when exceeded)	8
 Show Reports Object Types List (Defaults: 32)	32

Lineage Results Pane

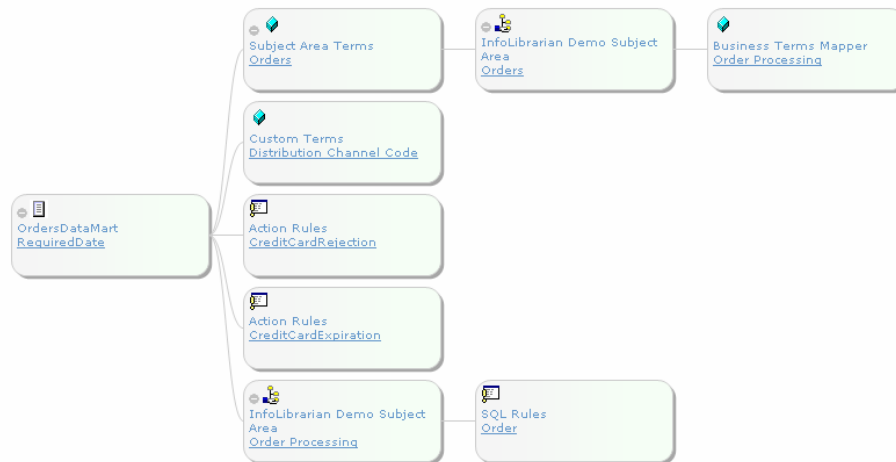
Workflow and Lineage Analysis - RequiredDate

TIPS: Lineage and Workflow

Expand All | Select All | Collapse All | Unselect All

Search: Arrange: Link type: Filter: Query: Workflow: Query:

OrdersDataMart.Orders.RequiredDate - Source lineages filtered by... 'business semantics, terms, rules and and subject areas'



Global Portal Settings | Search For Settings | Search Results Settings | Business Subject Areas | Lineage Analysis Settings | Custom Portal Perspectives

Lineage Analysis Settings

Max Levels to Display

Restricts the maximum levels of visual lineage to this setting. Increase this setting if required. However, some limit must be given to prevent circular references from endlessly processing.

ETL Objects Types List

Specify a comma delimited list of object types that represent the ETL processes in your environment. This setting will determine the object types that are included in the Lineage Filter "ETL" option. Typically ETL Packages, Transformations and DBMS Columns are indicated.

Semantics Objects Types List

Specify a comma delimited list of object types that represent Semantic(Business Subject Areas, Terms and Rules) in your environment. This setting will determine the object types that are included in the Lineage Filter "Semantics" option.

[Repository Object Types](#) Click this link to view the repository object types list.

Setting	Value
Max Levels to Display (Default: 10)	10
ETL Object Types List (Defaults: 5,21,7,27,14)	5,21,7,27,14
Semantics Object Types List (Defaults: 29,28,26,8,9)	29,28,26,8,9

Custom Portals

To achieve maximum flexibility, portal perspectives provide the simplest way to create entirely new sub-sites which inherit all of the functionality of the default site while allowing you to completely customize and enable or disable behavior to meet your needs. This too allows you to distinguish and maintain your own custom perspectives/portals.

Example #1 Co-Branding

Custom Templates Root Folder

Place the following files in a new sub site to create a co branded portal site.
That utilizes the framework default perspectives for all results and navigation pages.

- cobrandFooter.htm
 - Footer To Modify and place co branded logo etc.
- CobrandHeader.htm
 - Header To Modify and place co branded logo etc.
- default2.htm
 - Frames page displays Header and Footer
- default.htm
 - Frames page displays Header with no Footer
- TOC.asp
 - ASP Template Wrapper for Java Dynamic tree
 - Customize items displayed in tree
- TOC.htm
 - Table of Contents (Dynamic Left Tree) HTML Page

Drop them into a sub site below the root "InfoLib" to create a custom or co branded portal.

Custom Templates Sub-Folders

The Templates provided here allow you to customize the following:

- Custom Business Subject Area Handler.
- Custom Tips for Subject Area.
- Custom Results page for Metadata Details.
- Custom Business Metadata Details Views.
- Custom Table of Contents (Tree Views).
- Custom Portal Co brand Headers and Footers.
- Custom Contribution Handlers and Email notification Hooks.
- Custom Search Handlers.

FOLDER STRUCTURE FOR CUSTOM HANDLERS:

PortalResults - copy main template to create and customize new templates for Portal Results Custom Handler

SearchTemplate - copy main template to create and customize new templates for Search For Customization

SubjectArea - copy main template to create and customize new templates for Subject Area Categories Custom Results Handler

Once a template has been copied and modified it will be available via the Portal Settings within the InfoLibrarian Control Center.

NOTE: All templates and pages use the include files located in the Root Include Folder and the URLs are relative.

NOTE: All Images are located and referred to by the template pages in the Root Images folder.

WARNING!!!! DO NOT MODIFY THE CORE SITE - ONLY MAKE CHANGES TO FILES COPIED OR CREATED WITHIN THE "CUSTOM TEMPLATES" FOLDERS.
Do not overwrite the default custom handlers... make copies and rename them.
Or Create completely new sub-sites using the following methodology.

Example #2 Custom Perspective - New Sub-Site

Creating Custom Portal Perspectives

Copy the Templates from "CustomTemplates\BusinessPerspectives" to your custom co branded site (eg: infolib/CustomSite)

These templates provide all the basic pages necessary to deploy a custom portal perspective.

The Business Metadata Portal Perspectives - Main Pages

BusinessMetadata.asp

This page provides the Metadata Details view for an object when show details is clicked.

BusinessSearch.asp

Provided for Business Perspective Search For Options

Toc.asp

Provided for Custom Tree Handler for the Perspective

SearchResultsHandler.asp

Provided for navigation tree, keyword search and impact analysis results handling

header.htm

Co brand header page... replace this page with your own custom logo. Examples of co branding are found in the CustomTemplates\root folder.

Additional Template Locations for Customization based on Business Perspectives:

Custom Contribution Portals : "CustomTemplates\ContributionPerspectives" (Create a custom contribution perspective)

Custom Lineage Results : "CustomTemplates\LineagePerspectives" (Create a custom lineage handler)

Co branding Frames Pages : "CustomTemplates\Root" (Create co branded frames top and bottom)

Advanced Search Results and Pages : "CustomTemplates\AdvancedSearchPerspectives" (Create a custom advanced search)

Copy the files found within these folders to "infolib/CustomSubsite". All custom portal perspectives must be located at a single level below the "InfoLib" root site to work properly. Example: To create a custom contribution portal... Create a sub site such as "CustomContribPerspectiveA" below the "InfoLib" root and copy the contribution templates to that location for customization.

Additional Customization Style Sheets etc:

By default the templates refer to the styles and include files located in the "InfoLib/include" folder. You may copy these to your custom portal location to change these functionalities.

Please refer to the readme files and supplemental documentation provided to customize perspectives.

Note: Always document and backup the new templates/perspectives that you create. Future upgrades of InfoLibrarian may overwrite the default folders.

You will need to replace your files and folders once an upgrade is complete.

Portal Settings Affect on Custom Templates

The templates are setup to inherit settings from the Control Center for the portal. However, Each template contains commented code that can be implemented to over-ride the global and other portal settings set using the Enterprise Control Center.

Regional Settings

Regional settings are based on where scans or data entry occurs and are affected by the Windows Regional Settings for Date/Time formatting.

API's and Repository Queries – Modifying the Code

Refer to the InfoLibrarian Repository and change management API references for more details on how to call the API from the portal. Samples of the templates themselves should demonstrate most functions typically invoked via the API.

Inline SQL Queries

Inline SQL is supported from the web templates directly. Please refer to the InfoLibrarian Repository Documentation as well as the supplemental SQL Queries reference materials. Additional insight into the portal queries can be viewed directly within the templates.

Regex – Regular Expressions for syntax highlighting and code parsing

Regular expressions are supported and recommended for textual parsing for portal display purposes.

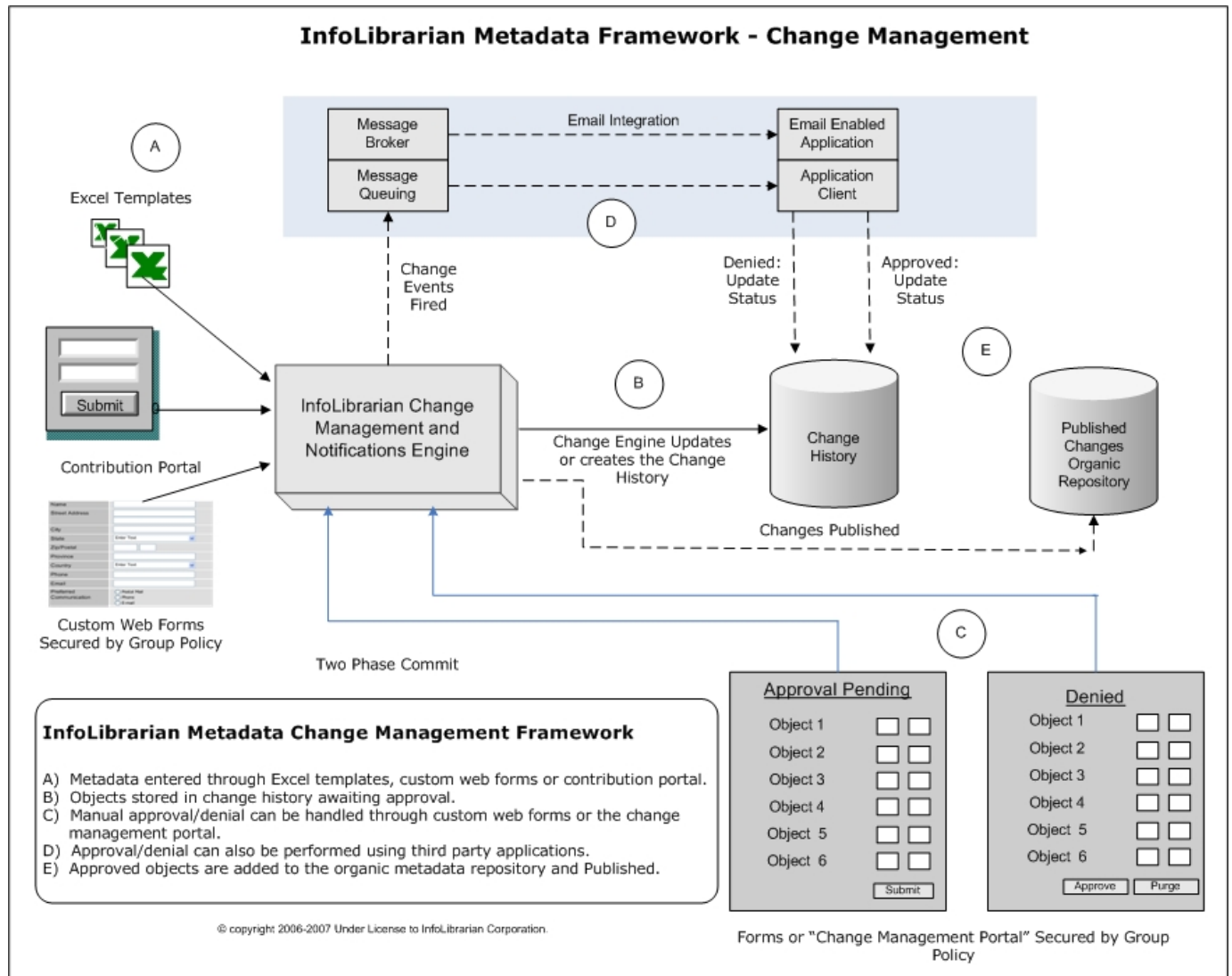
Additional code parsers can be developed by viewing the brushes/Regex java script located in the include folder. New brushes/code parsers can be developed by replicating one of the shBrushes JavaScript functions to custom parse your particular language syntax.

Best Approach

While co branding the default portal and creating handlers is the most direct and simplest form of customization for the InfoLibrarian Portal Framework, Creating a new portal perspective will give you the greatest control.

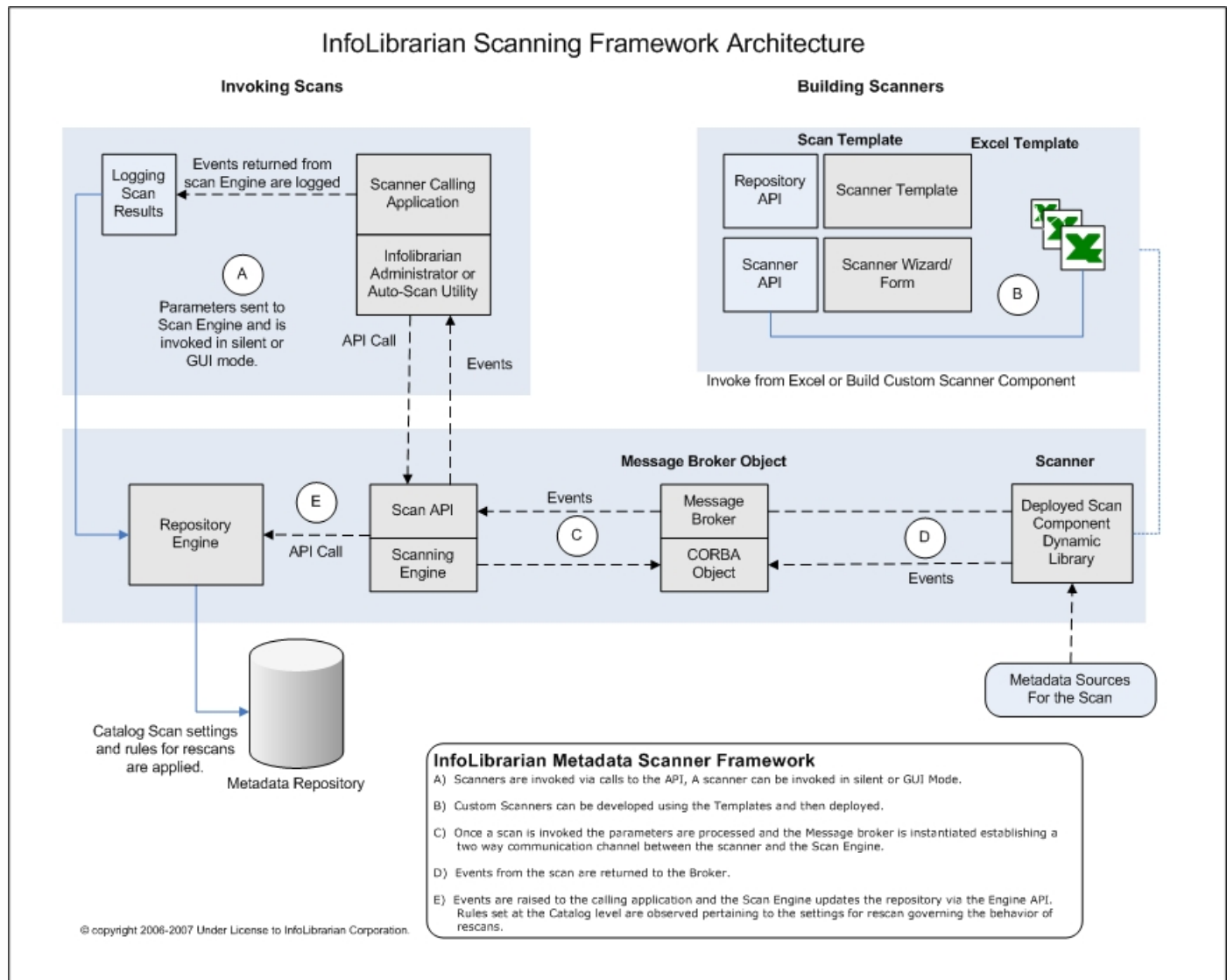
Starting with a co branded, custom perspective will allow you to replicate, and extend the existing portal functionalities in any web language and ultimately provide you with the complete code base for your own custom portal that can be maintained separately.

Portal Framework for Change Management



Custom Contribution Portals: "CustomTemplates\ContributionPerspectives" (Create a custom contribution perspective)

Scanner Templates and Loaders – Scanning Framework



InfoLibrarian Scanning Framework

How Scanning Works

Scanning,

OrganicObjects is updated including description/comments, the deleted flag is set if the source object has been dropped. (OLTP) Table or when a user uses the Administrator to delete things. This is so we don't orphan mappings, they are only flagged initially. OrganicObjects is constantly updated. New objects are added, never deleted only flagged.

Overview

The Organic Metadata (3 tables [MetadataMasterView]) is Appended to MetadataObjectsMart row by row (Historical, Read-Only, Never Updated, Summarized) Just like a data warehouse. (Hence MART) changes are calculated by the Snapshot generator while running which creates checksums of each row for fast compare.(you may notice RowGuid and szChecksum) The Changed, New and Deleted flags are also set by this process. Deleted objects are purged from organic table if "Purge deleted objects on rescan" is set. The mart is demoralized and obviously contains repeating columns depending on how many properties and parts(Splits long text to scale well) so we don't use BLOBS. The RowGuid is an artificial Surrogate key for the mart table, the szChecksum is a CRC 32 bit checksum calculated on all properties for change identification.

More Information,

The portal does contain allot of inline SQL, you can see how it all works by looking at some of this. also the advanced search, is a kind of query builder... You should notice the SQL String displayed to help you understand repository. Organic contains current view so to speak, whereas the Mart contains the history. Hence the 2 main views "MetadataMasterView" and "MetadataMartView"

Comments,

The data types in the repository are generic intentionally, so we may support Oracle, SQL Server and DB2 deployments. You will have to cast the timestamps as they are varchar's in your custom queries If you require custom queries, there's plenty of documentation and samples which should make you fairly comfortable with this.

Role of the Dictionary Table

The Dictionary Table is there to Override the scans from using source descriptions and comments. If you choose "Use Thesaurus on Rescan" the dictionary becomes the master list for descriptions and comments. If you want to set it up this way, users in Oracle for example cannot mess with your definitions by changing them in the source. Obviously, this would be done at a point in time when the initial scanning is complete.

The Spreadsheet Templates

The InfoLibrarian Metadata Spreadsheet Templates are an extremely powerful tool for loading, categorizing and building semantics and business metadata.

The Template loader has full access to the repository and scanning API's which allow you to Create Catalogs, Categories, Object types, objects and mappings. You can even perform repository lookups from the spreadsheets.

Having full programmatic control via VBA macros makes the templates easy to setup and maintain while giving you the capability to process, scrub and perform logic while loading your metadata.

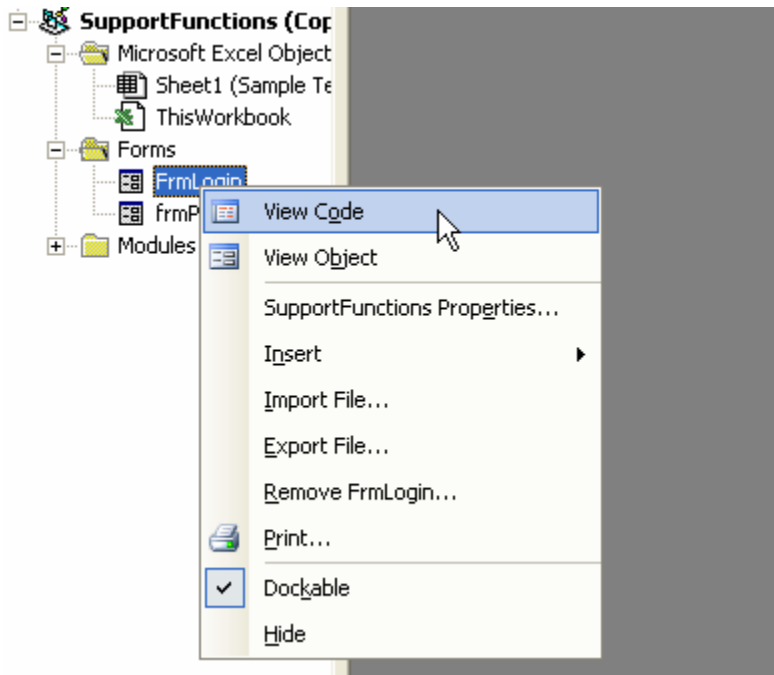
Various templates and samples are provided to do the following:

- Create business terms and their mappings
- Define semantics categories and map elements to those categories on load
- Define and map ETL processes
- Create user defined list of reports or other items from sources like the mainframe
- Perform logical calculations of existing metadata to automatically infer mappings in a batch.
- Convertible to scanners.

The Excel Templates are fully customizable.

Editing the Code - Excel Loader Templates

From the **Tools** Menu in Excel ... click **macros** and go to the **Visual Basic Editor**. Find the frmLogin and right click to **View Code**.



You will see something similar to the following:

```

*****
THIS IS THE MAIN MAPPING FUNCTION - EDIT THIS SECTION OF THE TEMPLATE TO CUSTOMIZE
*****

Private Sub MapMetadata(row As Integer)
    '-----
    ' SET UP FOR LOADING
    '-----
    MetadataMapper.CurrentObjectType = BUSINESS_TERM ' Hard Coding "Business Term" in this case
    MetadataMapper.szMetadataSource = "Business Terms" ' User Defined Metadata Source to load in
    MetadataMapper.szMetadataCatalog = "Custom Terms" ' User Defined Metadata Catalog to load in

    '-----
    ' USER DEFINED OBJECTS OR PROPERTIES
    '-----
    ' User Properties created in the Repository for metadata from Spreadsheet
    ' This creates custom property(s) for the object so that it will be editable in Administrator
    '-----
    ' If creating a Custom Object Type do this...
    ' Dim CustomObjectId As Long ' Reference for Object Type Id
    ' CustomObjectId = CreateObjectTypes("CustomObject") ' Create the Object Type if alive
    ' Reset This
    ' MetadataMapper.CurrentObjectType = CustomObjectId
    '-----
    Dim TermPropertiesArray(2) As String
    TermPropertiesArray(0) = "Examples" ' Customer editable property name.
    TermPropertiesArray(1) = "Another Property"
    ' Create the custom property(s) in the repository
    retval = CreateObjectUserDefinedPropertiesFromArray(BUSINESS_TERM, TermPropertiesArray) 'Hard
    '-----
    ' Set the Metadata Catalog and its properties here - Standard call to setup the proper values
    '-----
    retval = InfoLibRepository.SetMetadataCatalog(MetadataMapper.szMetadataSource, MetadataMapper
    MetadataMapper.SetMetadataSourceAndCatalog
    '-----
    ' MAP OBJECT BASE PROPERTIES - TEMPLATE
    '-----
    MetadataMapper.MapProperty Trim(Cells(row, 1)), OBJECT_NAME
    MetadataMapper.MapProperty ToString(Cells(row, 2)), OBJECT_DESCRIPTION
    MetadataMapper.MapProperty ToString(Cells(row, 3)), OBJECT_COMMENTS
    MetadataMapper.MetadataOrganicObject.Catalog = MetadataMapper.szMetadataCatalog
    MetadataMapper.MapProperty MetadataMapper.szMetadataCatalog, OBJECT_PARENT
    MetadataMapper.MetadataOrganicObject.DataSource = MetadataMapper.szMetadataSource

```

The Excel template contains inline samples of creating objects from data within your spreadsheet, adding properties and creating user defined custom properties and mappings.

Scanner API Function Samples

Creating or Updating an Object – Code Sample

```
MetadataMapper.MapProperty Trim(Cells(row, 1)), OBJECT_NAME
MetadataMapper.MapProperty ToString(Cells(row, 2)), OBJECT_DESCRIPTION
MetadataMapper.MapProperty ToString(Cells(row, 3)), OBJECT_COMMENTS
MetadataMapper.MetadataOrganicObject.Catalog = MetadataMapper.szMetadataCatalog
MetadataMapper.MapProperty MetadataMapper.szMetadataCatalog, OBJECT_PARENT
MetadataMapper.MetadataOrganicObject.DataSource = MetadataMapper.szMetadataSource
MetadataMapper.MapProperty CStr(MetadataMapper.CurrentObjectType), OBJECT_TYPE
```

```
'-----
' MAP ADDITIONAL CUSTOM PROPERTIES (Editable Properties) – TEMPLATE
' These User Defined properties must already be defined.
'-----
```

```
MetadataMapper.MapProperty ToString(Cells(row, 3)), REPOS_PROPERTY_TYPE, "Examples",
USER_DEFINED
'MetadataMapper.MapProperty ToString(Cells(row, 3)), REPOS_PROPERTY_TYPE, "Another
Property", USER_DEFINED
```

```
'-----
' MAP ADDITIONAL READ ONLY PROPERTIES (Read Only Properties) - TEMPLATE
'-----
```

```
MetadataMapper.MapProperty ToString(Cells(row, 3)), SOURCE_TYPE_PROPERTY, "Examples",
SOURCE_PROPERTY
MetadataMapper.MapProperty ToString(Cells(row, 3)), SOURCE_TYPE_PROPERTY, "Another
Property", SOURCE_PROPERTY
```

Creating Parent Child Object Hierarchies

When creating a parent child object instance hierarchy, create both object separately as shown above. Make sure the child objects "Parent" is set as follows to create an object as a child of another object.

```
MetadataMapper.MapProperty ParentObjectName, OBJECT_PARENT
```

Creating User Defined Properties in code

```
Dim TermPropertiesArray(2) As String
TermPropertiesArray(0) = "Examples" ' Customer editable property name.
TermPropertiesArray(1) = "Another Property"
retval = CreateObjectUserDefinedPropertiesFromArray(BUSINESS_TERM, TermPropertiesArray)
```

Note: Custom properties either needs to be created in the InfoLibrarian Administrator, Object Modeler or in code before you can map custom properties values to an object.

How to Create Target Mappings

'Obtain Serogate of target organic object using parameters that are recorded in spreadsheet cells

```
SetSourceTargetObject = InfolibRepository.GetOrganicMetadataObject(ToString(Cells(row, 7)),  
ToString(Cells(row, 8)), ToString(Cells(row, 9)), TargetType, ToString(Cells(row, 11)))
```

'Create the Association to the object.

```
MetadataMapper.MapProperty CurrentSourceObject.ReposSerogate, REPOS_PROPERTY_TYPE,  
"Association Mapping Description", OBJECT_NEXT
```

All you need is the GUID or serogate of the Target object, then map in an association "OBJECT_NEXT" which is a source mapping.

the Scanner API Reference contains the parameters of these calls

CREATING TARGET MAPPINGS – CODE SAMPLE

'Obtain Serogate of target organic object using parameters that are recorded in spreadsheet cells

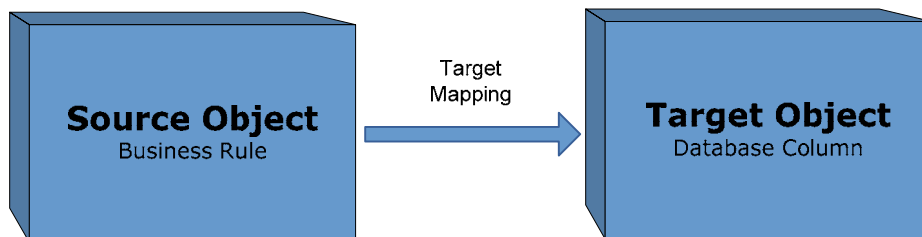
```
Set CurrentTargetObject = InfolibRepository.GetOrganicMetadataObject(ToString(Cells(row,  
7)), ToString(Cells(row, 8)), ToString(Cells(row, 9)), TargetType, ToString(Cells(row, 11)))
```

'Create the Association to the object.

```
MetadataMapper.MapProperty CurrentTargetObject.ReposSerogate, REPOS_PROPERTY_TYPE,  
"Association Mapping Description", OBJECT_NEXT
```

All you need is the GUID or serogate of the Target object, then map in an association "OBJECT_NEXT" which is a source mapping.

the Scanner API Reference contains the parameters of these calls.



How to Create Source Mappings

'Obtain Serogate of target organic object using parameters that are recorded in spreadsheet cells

```
SetSourceTargetObject = InfolibRepository.GetOrganicMetadataObject(ToString(Cells(row, 7)),  
ToString(Cells(row, 8)), ToString(Cells(row, 9)), TargetType, ToString(Cells(row, 11)))
```

'Create the Association to the object.

```
MetadataMapper.MapProperty CurrentSourceObject.ReposSerogate, REPOS_PROPERTY_TYPE,  
"Association Mapping Description", OBJECT_PREVIOUS
```

All you need is the GUID or serogate of the source object, then map in an association
"OBJECT_PREVIOUS" which is a source mapping.

the Scanner API Reference contains the parameters of these calls

CREATING SOURCE MAPPINGS – CODE SAMPLE

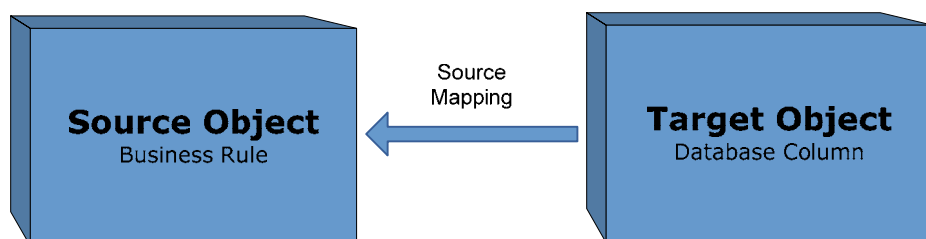
'Obtain Serogate of target organic object using parameters that are recorded in spreadsheet cells

```
Set CurrentSourceObject = InfolibRepository.GetOrganicMetadataObject(ToString(Cells(row,  
7)), ToString(Cells(row, 8)), ToString(Cells(row, 9)), TargetType, ToString(Cells(row, 11)))
```

'Create the Association to the object.

```
MetadataMapper.MapProperty CurrentSourceObject.ReposSerogate, REPOS_PROPERTY_TYPE,  
"Association Mapping Description", OBJECT_PREVIOUS
```

All you need is the GUID or serogate of the source object, then map in an association
"OBJECT_PREVIOUS" which is a source mapping.



Creating Subject Areas and Categories Using the API - Sample

```

***** SUBJECT AREA SEMANTIC MAPPINGS *****
Dim SubjectAreaLevelOne As Long
Dim SubjectAreaLevelTwo As Long
SubjectAreaLevelOne = 26 ' Default Top Level (Use the Object Type)
SubjectAreaLevelTwo = 28 ' Default Level Two (Use the Object Type)
-----
'SETTINGS TO SWITCH TO SUBJECT AREA CONTEXT
MetadataMapper.szMetadataSource = "Business Subject Areas" ' Subject Areas Section of Repository
MetadataMapper.szMetadataCatalog = "MySubjectArea" ' Subject Area to Create
MetadataMapper.CurrentObjectType = SubjectAreaLevelOne
MetadataMapper.MetadataOrganicObject.DataSource = MetadataMapper.szMetadataSource
-----
'LEVEL ONE CATEGORIES - Business Area Name
-----
MetadataMapper.MapProperty ToString(Cells(row, 11)), OBJECT_NAME 'Business Area Name
MetadataMapper.MapProperty "These categories define the business Areas perspective", OBJECT_DESCRIPTION
MetadataMapper.MetadataOrganicObject.Catalog = MetadataMapper.szMetadataCatalog
MetadataMapper.MapProperty MetadataMapper.szMetadataCatalog, OBJECT_PARENT
MetadataMapper.MapProperty CStr(SubjectAreaLevelOne), OBJECT_TYPE
MetadataMapper.CreateObjectInRepository
-----
'LEVEL TWO CATEGORIES
-----
MetadataMapper.MapProperty MyObjectName, OBJECT_NAME
MetadataMapper.MetadataOrganicObject.Catalog = MetadataMapper.szMetadataCatalog
MetadataMapper.MapProperty ToString(Cells(row, 11)), OBJECT_PARENT ' Business Area Name
MetadataMapper.MapProperty CStr(SubjectAreaLevelTwo), OBJECT_TYPE

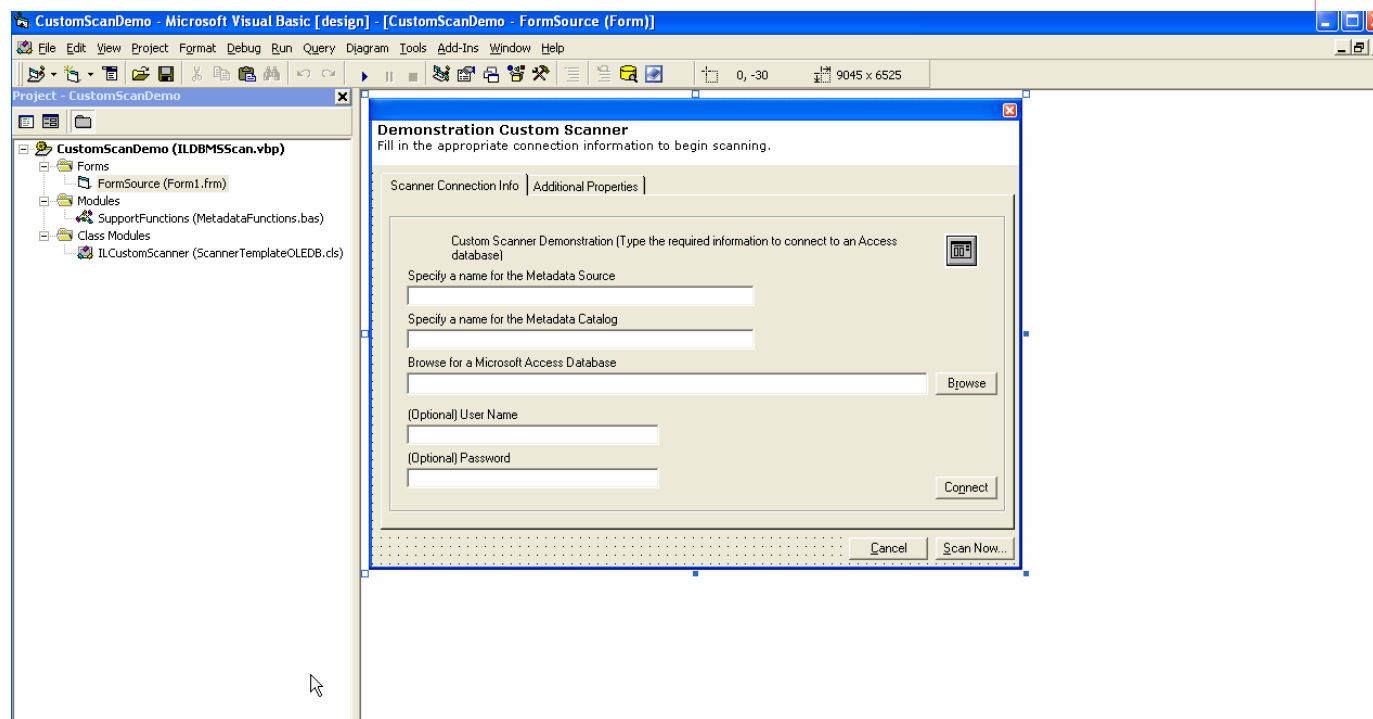
MetadataMapper.CreateObjectInRepository

'Map Terms to Subject Area Category - No Sub Category Level Three for this one.
Set CurrentTargetObject = InfolibRepository.GetOrganicMetadataObject("Business Terms",
Trim(ToString(Cells(row, 8))), Trim(ToString(Cells(row, 8))), BUSINESS_TERM, ToString(Cells(row, 1)))
'Create the Association
MetadataMapper.MapProperty CurrentTargetObject.ReposSerogate, REPOS_PROPERTY_TYPE, "Category-
Term Mapping", OBJECT_NEXT
MetadataMapper.CreateObjectInRepository
-----
'SETS THE SUBJECT AREA AND COMMITS THE CHANGES
retval = InfolibRepository.SetMetadataCatalog(MetadataMapper.szMetadataSource,
MetadataMapper.szMetadataCatalog, MetadataMapper.MetadataProvider, "", "",
MetadataMapper.MetadataProvider, Now, False, False, MetadataMapper.CurrentObjectType)
-----

```

Custom Scanner Development

Customizing a Scanner Template



Scanner Form

You will notice a form within the scanner project; this form is setup for passing the scanner parameters into the main scan function. Modify this form to customize a new scanner. In most cases you can select a template and only modify the title etc... on this form to distinguish your custom scanner in your environment. But you will not normally need to change the form itself. Note: If you do customize this form be sure to code the appropriate handlers behind the form as per the code template.

Scanner Code

The primary method of customizing a scanner involved modifying this function only:
MapMetadata()

The MapMetadata function contains the code for mapping the source metadata to the repository utilizing the Scanner and Repository API function calls. Please refer to the SDK documentation for more information.

Registering a Custom Scanner - Visual Basic Project Properties Dialog

The screenshot shows the 'CustomScanOne - Project Properties' dialog box with the 'General' tab selected. The 'Project Type' is set to 'ActiveX DLL' and the 'Startup Object' is 'Sub Main'. The 'Project Name' is 'CustomScanDemo'. The 'Project Description' is 'Our Custom Metadata Scanner - Demo'. The 'Threading Model' is set to 'Single Threaded'. The 'Upgrade ActiveX Controls' checkbox is checked. The 'Thread Pool' is set to 1 thread.

CustomScanOne - Project Properties

General | Make | Compile | Component | Debugging

Project Type: ActiveX DLL Startup Object: Sub Main

Project Name: CustomScanDemo

Help File Name: Project Help Context ID: 0

Project Description: Our Custom Metadata Scanner - Demo

☐ Unattended Execution

☒ Upgrade ActiveX Controls

☐ Require License Key

☐ Retained In Memory

Threading Model

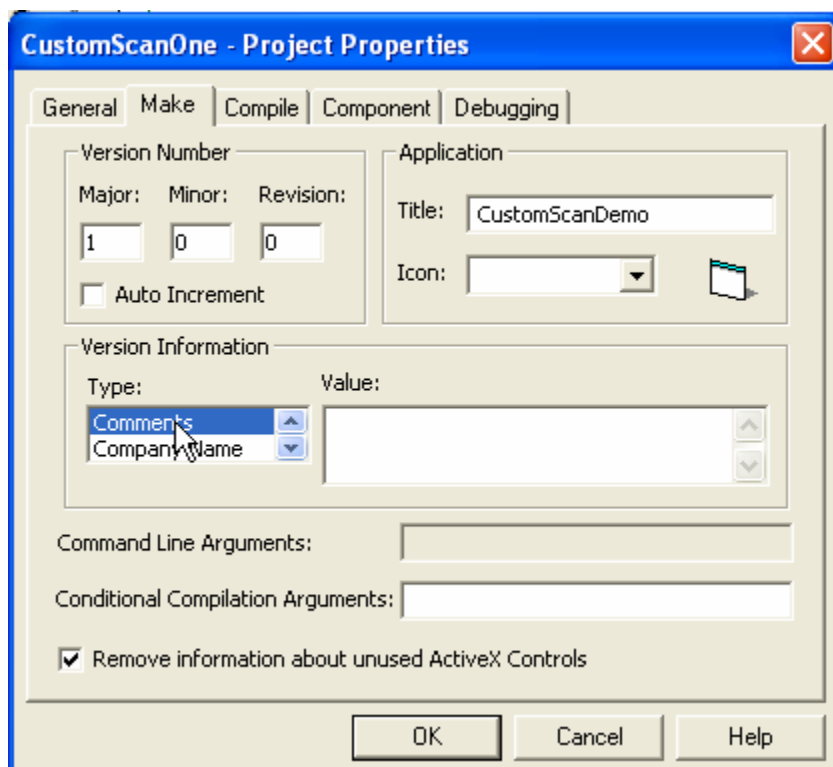
Single Threaded

☐ Thread per Object

☒ Thread Pool 1 threads

OK Cancel Help

Within the Project Properties Dialog, Provide a name and description for your custom scanner.



On the Make tab ensure that the title is the same as in the previous step.

```

CustomScanDemo - Microsoft Visual Basic [ design ] - [ILCustomScanner (Code)]
File Edit View Project Format Debug Run Query Diagram Tools Add-Ins Window Help
Ln 30, Col 50

project - CustomScanDemo
CustomScanDemo (ILD8M5Scan.vbp)
  Forms
    FormSource (Form1.frm)
  Modules
    SupportFunctions (MetadataFunctions.bas)
  Class Modules
    ILCustomScanner (ScannerTemplateOLEDB.cls)

(General) ScanMetadata

' The Product is protected by copyright and other intellectual property laws and treaties.
' InfoLibrarian Standard Scanner COM API Template ( )
'
' *** LOCAL VARIABLES ***
'
Private ObjectCount As Long 'Object Count returned by scan
Private arrCount As Long 'Used to parse array in Scanner Pump
Private SourcePathsArray() As Variant 'Array of paths to scan
Private SourceFileTypesArray() As Variant 'Array of File types filters
Private retval As Boolean

' Scanner Events Raised to the Client
Public Event ObjectCount(EventName As String, ByVal ObjectCount As Long)
Public Event ProgressPercent(EventName As String, ByVal Percent As Integer)
Public Event ResetProgress(EventName As String, ByVal Max As Integer)
Public Event ProgressMessage(EventName As String, ByVal Message As String)
Public Event cRaiseError(EventName As String, ByVal Message As String, ByVal Number As Long, ByVal Source

Private WithEvents ScannerAPI As ILScannerAPI

'
' MAIN SCANNER FUNCTION
'
Public Function ScanMetadata(ScannerArguments() As Variant) ' Standard Scanner Entry Point Function

    Set ScannerAPI = New ILScannerAPI 'Creates new instance of Scanner API
    ScannerAPI.MetadataProvider = "CustomScanDemo" 'Change this to the scanner you create here
    ScannerAPI.CurrentObjectType = 0 'Set this to the object type being scanned in the catalog
    Call ScannerPump(ScannerArguments)

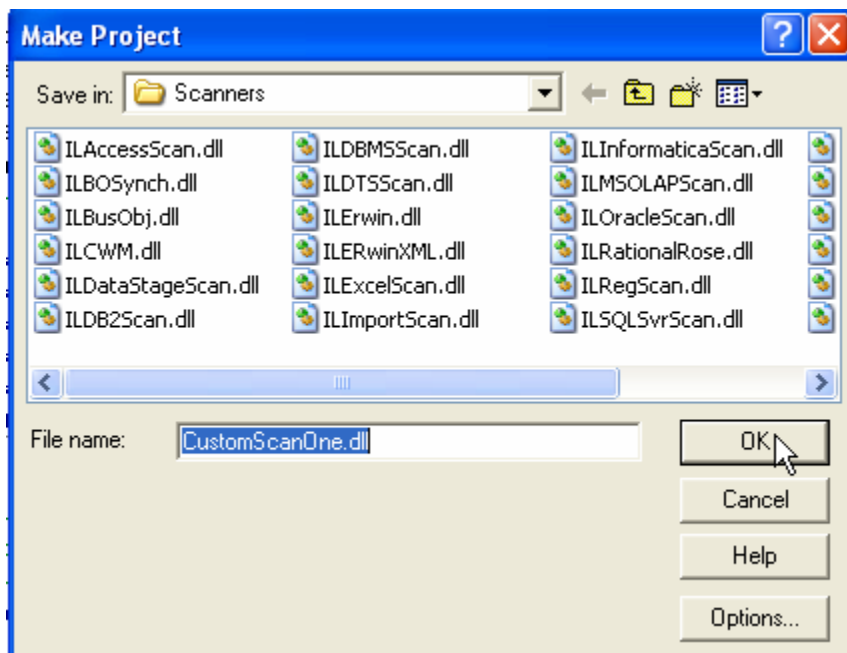
End Function

' MAIN METADATA MAPPING FUNCTION

```

Important, within the Class “CustomScanner”, change this line of code to specify the scanner name provided in the previous 2 steps. Ensure that it is identical and it is case sensitive.

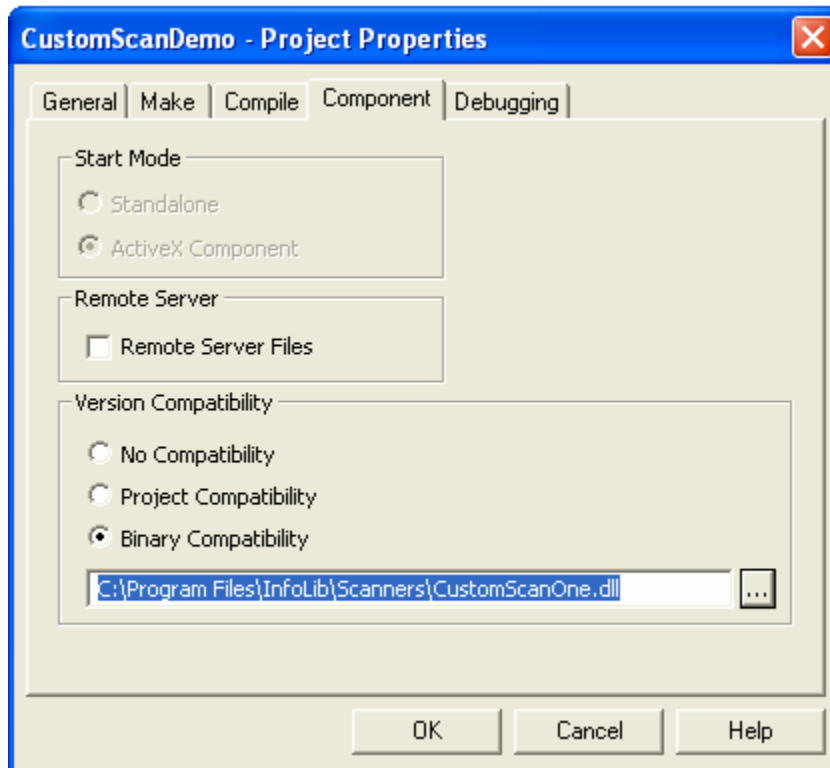
Building the DLL for the Scanner



In the file menu click "Make xxx.dll" to compile your dll.

Provide a unique dll name here.

Once the dll is initially built...



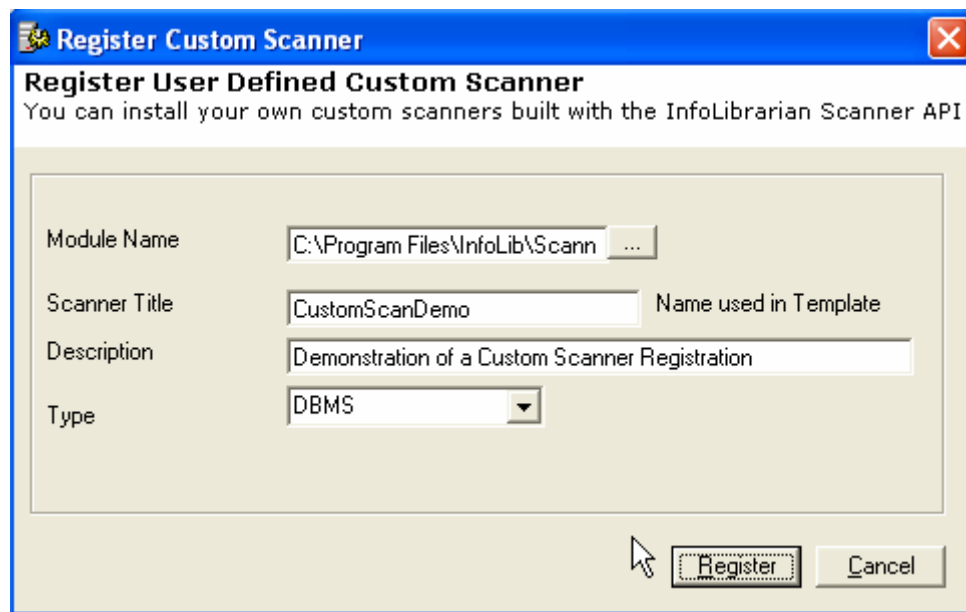
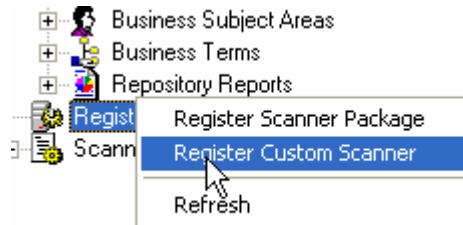
Within the project properties dialog, set the Binary Compatibility then rebuilt the dll.
This will enable debugging and proper registration of the dll.

Note : When building a custom scanner for the first time, ensure that the binary compatibility is set to "No Compatibility" Build the dll then reset to "Binary Compatibility" then rebuilt the dll again before deploying.

Registering the Scanner

1. Copy the dll to the machine on which you wish to register the custom scanner.
2. Register the new scanner dll using regsvr32.exe
3. Open the InfoLibrarian Administrator

Right click on the “Register Custom Scanner” in order to register your new custom scanner.



In this dialog enter the information as follows:

Module Name: Full path to dll created

Scanner Title: Name specified in the project dialog and template

Any description you wish

Choose the type of scanner template that was utilized for building this custom scanner.

Running the Scanner

 InfoLibrarian Metadata  Metadata Semantics  Business Rules  Business Subject Areas  Business Terms  Repository Reports  Registered Scanners  Scanning Logs	 Scanner	Description	Version	Engine Build
	 Access Metadata Scanner	Scans Access Databases for Schema Metadata	4.1 Compatible	4.1
	 Ascential DataStage ETL Scanner	Scans Ascential Data Stage ETL Metadata	4.1 Compatible	4.1
	 Business Objects Metadata Scanner	Scans Business Objects DBMS based Universes	4.1 Compatible	4.1
	 CustomScanDemo	Demonstration of a Custom Scanner Registration	CUSTOM	NA
	 DB2 Metadata Scanner	Scans DB2 RDBMS Metadata	4.1 Compatible	4.1
	 DTS Package ETL Scanner	Scans DTS Packages	4.1 Compatible	4.1
	 Erwin Native Scanner	Captures Erwin Metadata from ER1 Files	4.1 Compatible	4.1
	 ERwin XML Scanner	Scans Erwin XML Extracts for Model Metadata	4.1 Compatible	4.1
	 Excel Metadata Scanner	Scans Excel Files for Schema Metadata	4.1 Compatible	4.1
	 InfoLibrarian Business Objects Synch	Updates Terms and Descriptions to Business Objects	4.1 Compatible	4.1
	 Informatica Repository Scanner	Scans, maps metadata and lineages from Informatica	4.1 Compatible	4.1

Once you refresh the list you should see your custom scanner registered similar to the above.
You will notice the version should indicate "CUSTOM"

Double Click the scanner to run it as with any other scanner.

InfoLibrarian Repository

Please refer to the InfoLibrarian Repository Reference for more information.

Query Sample for InfoLibrarian Repository 4.1

Note:

“Serogate”, “GUID” and “Surrogate” are used throughout interchangeably and are synonymous. Queries can be on the Organic, Master views, Metamart or Change History tables.

Get the Guid or Serogate for a particular object

```
SELECT serogate,objectname from MetadataMasterView where objectname = 'ObjectName'
```

CurrentOrganicKey can be the serogate from any object.

Get all read only Properties for any object

```
SELECT * FROM MetadataMasterView where OrganicKey= CurrentOrganicKey and  
(PropertyType=1 or Propertytype=4) and deleted='0' and snapshot='snapshotValue')
```

Get Target Associations for an object

```
SELECT * FROM MetadataMasterView where OrganicKey= CurrentOrganicKey and  
(PropertyType=3) and deleted='0' and snapshot='snapshotvalue')
```

Get objects pointing to this one

```
SELECT * FROM MetadataProperties where  
MetadataProperties.PropertyValue='CurrentOrganicKey' and (PropertyType=2 or  
Propertytype=3) and deleted='0' )
```

Get SubjectAreas

```
Select MetadataCatalog from metadatacatalogs where MetadataSource = 'Business Subject  
Areas'
```

-This gives you each subject Area
then

Get SubjectArea Categories

Category Levels are actually object types.

Check object modeler for your subject area object type strucutre.

```
Select Objectname from metadataorganicobjects where  
metadatacatalog='SubjectAreaFoundAbove' and objecttype='SubjectAreaTopLevel'
```


Get SubjectArea Mappings (IE: Terms)

Get Target Associations for an object

Terms are mapped to subject Area categories so query them out.

```
SELECT * FROM MetadataOrganicObjects where OrganicKey= ObjectSerogateFromAbove and  
(PropertyType=3) and deleted='0' )
```

Get Object Children

```
select b.serogate,b.objectname,b.objecttype,b.objecttypeName  
from MetadataOrganicObjects a, MetadataOrganicObjects b  
where a.serogate=9032107  
and b.Parent=a.ObjectName  
and b.deleted='0'  
and b.objecttypename='ObjectType'
```

Get Properties for previous ie: Parent Object

```
SELECT * FROM MetadataProperties where MetadataProperties.ParentObject=9032125 and  
(PropertyType=1 or Propertytype=4) and deleted='0'
```

META MART SPECIFIC HISTORICAL QUERIES

METAMART DELTA QUERIES DIRECT

```
Select * from MetadataMartMasterView where changed='1'
```

```
Select * from MetadataMartMasterView where Deleted='1'
```

```
Select * from MetadataMartMasterView where New='1'
```

CALCULATE DELTAS BETWEEN TWO SNAPSHOTS

```
UPDATE A set A.changed='1' from MetadataObjectsMart A,MetadataObjectsMart B  
WHERE a.snapshot = " & " " & CurrentSnapshot & ""  
AND b.snapshot = " & " " & LastSnapshot & ""  
AND a.OrganicKey=b.OrganicKey  
AND b.rowguid = a.rowguid  
AND a.szChecksum <> b.szchecksum
```

CALCULATE NEW OBJECTS BETWEEN TWO SNAPSHOTS

```
UPDATE metadataobjectsmart set new ='1'  
WHERE snapshot = " & CurrentSnapshot & ""  
AND RowGuid NOT IN"  
SELECT RowGuid From MetadataObjectsMart b"  
WHERE snapshot=" & LastSnapshot & " )"
```

GET AN OBJECT AND ITS PROPERTIES FROM A POINT IN TIME

```
SELECT * FROM MetadataObjectsMart where OrganicKey= 12345  
and deleted='0' and snapshot='01/01/1984'
```

GET ASSOCIATIONS FROM THE MART FOR A POINT IN TIME

```
SELECT * FROM MetadataObjectsMart where OrganicKey= 12345  
AND (PropertyType=2 or Propertytype=3) and deleted='0' and snapshot='01/01/1984'
```

GET CHILD OBJECT TYPES FROM THE MART FOR A POINT IN TIME

```
SELECT DISTINCT ObjectType FROM MetadataObjectsMart WHERE  
MetadataSource='MetadataSourceName'  
AND MetadataCatalog='CurrentCatalogName' AND Parent='CurrentParentName' AND  
SNAPSHOT ='01/01/1994' ORDER BY ObjectName"
```

GET CHILD OBJECTS OF A PARTICULAR TYPE FROM THE MART FOR A POINT IN TIME

```
SELECT DISTINCT ObjectName,description,comments,organickey FROM MetadataObjectsMart  
WHERE MetadataSource='MetadataSourceName'  
AND MetadataCatalog='CurrentCatalogName' AND Parent='CurrentParentName' AND  
SNAPSHOT ='01/01/1994' AND ObjectType=5 ORDER BY ObjectName"
```

LOOKUP THE MART (SUBQUERY) Surrogate (GUID) for any of the above queries to query a particular instance of an object at a point in time

```
SELECT Serogate from MetadataObjectsMart where organicKey=1234 AND Snapshot =  
'01/01/2006'
```

NOTE:

Additional reference to these queries can be found in the handler results pages found in the Custom Templates Folder. These pages contain all of the various queries used to produce the results sets in the portal.

Reports

Additional third party reports can be built using these supplied queries and providing WHERE conditions and sub-queries processing to generate tailored custom reports.

Reports can then be linked as linked documents to a subject area category for categorizing and launching any custom reports from third party report generators. These reports can either be in their native tool format or web pages.

Further Information

Additional reference information can be found in the repository API Reference documentation.

Additional Help

Please refer to the InfoLibrarian Books online, Portal Documentation and other product documentation provided with the product for more information.

Metadata Semantics How To's can be found within the InfoLibrarian Books Online under the section "How To".

Training and Consulting

Additional training and consulting for the InfoLibrarian is available on request.

Contact sales@infolibcorp.com for more information