CCO eCoaching\_Log

**SharePOint Interface Bean (SOIBean) Detail Design**

There is guidance within this template that appears in a style named InfoBlue. This style has a hidden font attribute that allows you to toggle whether it is visible or hidden in this template. Use the Microsoft® Word® menu **Tools > Options > View > Hidden Text** check box to toggle this setting. There is also an option for printing: **Tools > Options > Print**.

# Purpose

The SharePOint Interface Bean (SOIBean) utility is to serve as a lightweight, configurable solution for downloading data from and uploading data to Microsoft SharePoint lists and libraries, which are heavily used by Contact Center Operations (CCO). The CCO eCoaching\_Log project is leveraging a that has been generalized for reusability. At this time, SOIBean will be used to interface between the eCoaching\_Log SQL Server database and SharePoint lists and libraries used by the Quality program.

# Design Structure

The utility will consist of the following software components:

* C# command line application using the Microsoft .NET 4.5 framework and Microsoft SharePoint Client-Side Object Model (CSOM).
* Microsoft SharePoint list / library
* SQL Server 2012 or newer database

# Assumptions, Decisions, Constraints, and Justifications

| **Decision, Constraint** | **Justification** |
| --- | --- |
| The application will be able to authenticate to both on-premises SharePoint and SharePoint Online (cloud-based).  **Constraints:** To connect to SharePoint Online, the application will need to be registered by the SharePoint site administrator. | Both of setups of SharePoint have been used recently by CCO, the consumer of this application. |
|  |  |
|  |  |
|  |  |

# Key abstractions

1. Create a process to download records from a Microsoft SharePoint list or library to a SQL Server database.
2. Create a process to upload records from a SQL Server database to a Microsoft SharePoint list or library
3. The upload process will have the ability to report back to the database the status of the upload to support re-sending of records that were not loaded and closure of records that have already been loaded. Record updates in SharePoint will be supported.
4. The upload process will have the ability to identify duplicate records based on specified unique fields and report duplicates back to the database.

# Modules

# SOIBean

This will be a command line program to extract records from a Microsoft SharePoint list or library and to import them into a SQL Server database table. Conversely, the program will also be able to take records from a SQL Server stored procedure and upload them to a Microsoft SharePoint list or library, reporting the upload status back to the database.

### Interface List

#### Input

* One or more standard .NET configuration files on the command line. The files will be checked in sequence for a desired configuration value, and the last value found will be used. If no file contains the configuration value, the value will be treated as undefined.
* Sample call:

SOIBean ecl.config environment.development.config ecl\_bingo\_london.config

#### Output

* For download: Data imported into SQL Server database table.
* For upload: Data uploaded to SharePoint list or library, plus affected SharePoint GUID and / or row ID if desired. Upload status updated in SQL Server database if applicable. Possible statuses returned:
  + Loaded – Successfully inserted into SharePoint
  + Updated – Successfully updated in SharePoint
  + Duplicate – Record is a duplicate of another SharePoint record
  + Load Error – Issues were encountered inserting / updating the record

#### External Interfaces

* Microsoft SharePoint list
* SQL Server database table type and stored procedure

### Interface Relationships

#### Authentication

##### Both on-premises SharePoint with Windows authentication and SharePoint Online with application registration (using Client ID and secret for authentication) will be supported.

##### SQL Server authentication will follow the method specified in the supplied connection string in the configuration file.

#### Downloads

##### Source data is pulled from a Microsoft SharePoint list.

##### Extracted data is populated in the specified Microsoft SQL Server table type and passed to the specified stored procedure at the configured SQL Server database connection.

#### Uploads

##### Source data is pulled from the specified Microsoft SQL Server stored procedure at the configured SQL Server database connection.

##### Extracted data is populated in the specified Microsoft SharePoint list. Data may be inserted as new records, or if configured, existing records may be updated.

##### Results of the insert / update to the specified SharePoint list will be posted back to the configured database via the specified status reporting stored procedure. Affected SharePoint record GUID and row ID can also be provided if desired.

### Library Usage

#### Microsoft.Sharepoint.Client – for interacting with the Microsoft SharePoint site

#### Microsoft.SharepointOnline.CSOM – Client side object model for interacting with Microsoft SharePoint Online sites

#### Newtonsoft.Json – for transforming site values

#### log4net – for logging and email capabilities

### Software Product Modules List

#### SOIBean

### Module Definition

#### SOIBean

Main executable console program that will serve as the interface to transfer records between a Microsoft SharePoint list and a SQL Server database table, transforming any specified values to the desired target text.

### Design Criteria and Constraints List

#### General

##### Configuration file sections that contain credentials or connection strings will need to be encrypted.

#### Downloads

##### Destination stored procedure will need to accept a table type parameter. The table type will need to exist and have compatible column types for the SharePoint data values.

##### Destination stored procedure will need to have the following parameters (parameters will be identified in the program by name):

* + @records – The table type of records being passed
  + @list\_identifier – The configured list identifier text / code
  + @is\_truncate – Bit flag to indicate to procedure if table(s) should be truncated or not

#### Uploads

##### Source stored procedure will not require parameters to be provided.

##### Destination status reporting stored procedure will need to accept a table type parameter. The table type will need to exist and have the exact columns returned from the source stored procedure, along with an nvarchar(50) column for returning the upload result.

##### Destination status reporting stored procedure will need to have the following parameters (parameters will be identified in the program by name):

* + @records – The table type of records being passed

### Assumptions and Dependencies List

##### The Microsoft SharePoint list being extracted from or written to must be running and accessible via a web URL to the service or account running the executable.

* + For on-premises SharePoint using Windows authentication: The account must have read permissions to the list.
  + For SharePoint Online using application registration: The application must be actively registered on the site with read permissions to the list.

##### The Microsoft SQL Server database receiving or supplying the data must be running and accessible to the service or account running the executable. The account must have permissions to execute the specified stored procedure(s).

### Detail Design

#### Executables

##### SOIBean

|  |  |
| --- | --- |
| **Component** | SOIBean |
| **Description** | Connects to a Microsoft SharePoint List URL, extracts the records, and outputs the desired fields to a database table. Alternatively, may also execute a database stored procedure to extract records and output the desired fields to a SharePoint List URL. |
| **Processing Steps** | Load the configuration files in the order received in the program arguments, overwriting earlier-supplied configuration values with values from later-supplied configuration files. |
|  | Establish a SharePoint Web ClientContext from the “ShareURL” value in the application config files and based on the “AuthMethod” effective configuration setting. |
|  | If AuthMethod is “ClientID”, use “ClientID” and “ClientSecret” to obtain an access token for the client context, using the SharePointOnline CSOM. |
|  | If AuthMethod is “Windows”, use default authentication mode. |
|  | Else throw an unsupported exception for the authentication method. |
|  | Get the SharePoint list by title from the “ShareList” value in the effective configuration. |
|  | If “ListFieldsOnly” in the configuration is true, load the collection of fields including the title and the internal column name from the SharePoint list and output the results in an email. |
|  | Else if “Direction” in the configuration is Download: |
| ***Retrieve records from SharePoint*** | Create a Caml Query and set the view XML to the “ShareViewXml” value from the effective configuration to retrieve the desired fields and records. |
|  | The results may be paged, so loop through each result page, saving the page’s results to a comprehensive list of results. |
|  | Convert all returned record list items to a list of Dictionary key/value objects. |
|  | Get the list of extract fields for the file (“ExtractFields” from the configuration) and the list of column transformations (“ColumnTransforms”). |
|  | Create a new DataTable. Number of columns should be the number of extract fields, and names should be the ColumnNames in the ExtractFields. |
|  | For each returned record |
|  | Create a new DataRow. |
|  | For each field in the extract field list |
|  | Throw an exception if the extract field key name does not exist in the returned record. |
|  | If the extract field is a lookup type field in SharePoint (user value or lookup value), get the ID for the lookup field if “LookupId” is specified as True in the configuration; otherwise get the lookup value for the field. |
|  | If the extract field should be defaulted to a value, convert a null value for that field in the returned record to the desired default value; otherwise use the string value of the record’s field. |
|  | If the extract fields target data type is a string type and the extract field value’s length is greater than the target table type column’s maximum allowed length, log a warning about the value exceeding the column length. |
|  | If the record’s field key name and field value exist in the list of column transformations, use the transform’s replacement value as the record’s field value. |
|  | Add the calculated field value to the DataRow. |
|  | Add the DataRow to the DataTable. |
|  | If the DataTable is not empty |
|  | Get the destination connection string from the effective configuration (connection string with key equal to value of “ConnectionStringKey” application setting). |
|  | Establish a database connection. |
|  | Fill a table type of “TableTypeName” from the DataTable and execute the stored procedure “ProcedureName” from the configuration, passing the table type. |
|  | Close resources. |
|  | Else if “Direction” in the configuration is Upload: |
|  | Get the destination connection string from the effective configuration (connection string with key equal to value of “ConnectionStringKey” application setting). |
|  | Establish a database connection. |
|  | Execute the stored procedure “ProcedureName” from the configuration and fill a DataSet, logging the number of records retrieved.  Number of columns returned should be at least the number of extract fields, and all ColumnNames in the ExtractFields should exist in the dataset columns. |
|  | Close resources. |
|  | If at least 1 record is retrieved: |
|  | If “Upload.DuplicateCheck” in the configuration is true, retrieve records from the SharePoint site as in the Download ***Retrieve records from SharePoint*** step. Convert all returned record list items to a Dictionary of key/value objects by item SharePoint ID and list item. |
|  | Get the list of extract fields for the file (“ExtractFields” from the configuration) and the list of column transformations (“ColumnTransforms”).  From the list of extract fields, get the list of updatable fields (any field where “CanUpdate” JSON value is true). |
|  | If “Upload.DuplicateCheck” in the configuration is true, also get:   * + - * the list of SharePoint key fields from the list of extract fields (any field where “IsSPKey” JSON value is true)       * the list of unique SharePoint fields (any field where “IsSPUnique” JSON value is true), and       * the list of unique SharePoint fields that are NOT in the list of updatable fields. |
|  | For each returned record in the DataSet: |
|  | Initialize an insert flag to true, and update flag to true only if both key fields and update fields are specified (otherwise initialize to false). |
|  | If “Upload.DuplicateCheck” in the configuration is true or the update flag is true: |
|  | For each field in the key field list |
| ***BEGIN Convert DB Value to SharePoint*** | Throw an exception if the data table does not contain the extract field column name. |
|  | If the extract field should be defaulted to a value, convert a database null value for that field in the returned record to the desired default value; otherwise use the string value of the record’s field. |
| ***END Convert DB Value to SharePoint*** | If the record’s field key name and field value exist in the list of column transformations, use the transform’s replacement value as the record’s field value. |
|  | Add the calculated field value to a key Dictionary by extract field key name. |
|  | For each field in the unique field list |
|  | Convert the returned database record values as in the ***Convert DB Value to SharePoint*** steps. |
|  | Add the calculated field value to a key Dictionary by extract field key name. |
|  | If the update flag is true |
|  | For each entry in the SharePoint list item dictionary (containing all records retrieved from SharePoint): |
|  | If the list item’s key fields match the current database record’s key field values, this is an update. Save the existing SharePoint list item and set the insert flag to false. Replace the value of any transformed unique key that is non-updateable with the original SharePoint value of the key.  Do not change any standard SharePoint fields that can’t be modified, e.g. ID, GUID, Author, Editor. |
|  | Break out of the loop. |
|  | If “Upload.DuplicateCheck” in the configuration is true |
|  | For each entry in the SharePoint list item dictionary (containing all records retrieved from SharePoint): |
|  | If an existing SharePoint list item was found and the current list item ID matches the existing item ID, continue to the next record. |
|  | Else if the current list item’s unique fields match the current database record’s unique field values, this is a duplicate. Set the insert and update flags to false. Write the record to the database as “Duplicate” status as follows: |
| ***Set SharePoint Upload Status in DB*** | Add record to a table type of “TableTypeName” from the DataTable. Set “Upload.ReportStatusField” column in the table type to the appropriate status. If provided in configuration, set “Upload.ReportIdField” column in the table type to the affected SharePoint row ID. If provided in configuration, set “Upload.ReportGuidField” column in the table type to the affected SharePoint GUID.  Execute the stored procedure “Upload.ReportProcedureName” from the configuration, passing the table type. |
|  | Break out of the loop. |
|  | If the update flag is true and an existing SharePoint list item was found: |
|  | Update the record in the SharePoint list, changing only the fields in the update fields list. For each field in the update fields list, only change the SharePoint list value if the source field value is non-null and non-blank / whitespace, or if the “AlwaysUpdate” flag for the field in the configuration is true.  Do not change any standard SharePoint fields that can’t be modified, e.g. ID, GUID, Author, Editor. |
|  | Add the updated existing list item to the Dictionary of SharePoint list items. |
|  | Write the record to the database as “Updated” status as in the ***Set SharePoint Upload Status in DB*** step. |
|  | Else if the insert flag is true: |
|  | Insert the record in the SharePoint list. Do not set any standard SharePoint fields that can’t be modified, e.g. ID, GUID, Author, Editor. |
|  | If checking for duplicates, add the new list item to the Dictionary of SharePoint list items. |
|  | Write the record to the database as “Loaded” status as in the ***Set SharePoint Upload Status in DB*** step. |
|  | Else if not checking for duplicates: |
|  | Write the record to the database as “Load Error” status as in the ***Set SharePoint Upload Status in DB*** step. |
|  | Close resources. |
|  | Send an email with the results. If “Direction” in the configuration is Upload, this should include counts of new records inserted into SharePoint, records updated on SharePoint, and records that were skipped due to duplicates or loading error. |

#### Classes

##### ColumnTransform

|  |  |
| --- | --- |
| **Component** | ColumnTransform class |
| **Description** | Contains properties to store column transform information |
| **Properties / Methods** |  |
|  | ColumnName – SharePoint key name of column to transform |
|  | Find – Field value to find for replacement / transformation |
|  | Replace – Replacement value |

##### ExtractField

|  |  |
| --- | --- |
| **Component** | ExtractField class |
| **Description** | Contains properties to store extract field information |
| **Properties / Methods** | ExtractField()  Constructor |
|  | HasDefault() – Return true if Default is not null or empty string; false otherwise. |
|  | ColumnName – Destination column name in table type for the SharePoint field. |
|  | KeyName – SharePoint key name of column to extract |
|  | Default – Default value to use if SharePoint field value is NULL. |
|  | CanUpdate – Can the field be updated? (default = false) |
|  | AlwaysUpdate – Only applicable if CanUpdate = true. Always update the field, even if the new value is blank? (default = false) |
|  | IsSPKey – Is this field a SharePoint key field? (default = false) |
|  | IsSPUnique – Is this field part of a unique key in SharePoint? (default = false) |
|  | LookupId – For lookup fields, should the lookup ID be used? If false, lookup value will be used instead. (default = false) |

##### Configuration File Settings

|  |  |
| --- | --- |
| **Key Name** | **Description** |
| <connectionStrings> **section** | |
|  | List of database connection strings – **will be encrypted** |
| <credentialAppSettings> **section** | |
|  | Custom section to hold credential strings – **will be encrypted** |
| ClientId | Only applies if AuthMethod = “ClientID”. The application’s SharePoint-registered Client ID to use for authentication. |
| ClientSecret | Only applies if AuthMethod = “ClientID”. The application’s SharePoint-registered Client Secret to use for authentication. |
| EmailUsername | Only applies if EmailAuth = “Basic”. User name for email authentication. |
| EmailPass | Only applies if EmailAuth = “Basic”. |
| <appSettings> **section** | |
| ConnectionStringKey | Key name for the connection string to use from the <connectionStrings> section |
| AuthMethod | Specifies the method to use to authenticate to SharePoint  Supported Options: Windows, ClientID |
| Direction | Direction of data with regards to the SharePoint List. Indicates if data is being uploaded to the list or downloaded from the list.  Supported options: Upload, Download |
| ProcedureName | Download: Name of the stored procedure to call to import records into database.  Upload: Name of the stored procedure to call to receive records to upload to SharePoint.  May include schema name.  Ex: schema.spSampleProcedure |
| TableTypeName | Name of the table type for passing records to stored procedure. May include schema name.  Ex: schema.sampleTableType |
| ProcedureListID | Download: List identifier string to be used by the stored procedure to identify the source of the records.  Upload: N/A  Ex: Bingo Logs Upload |
| ProcedureTruncate | Download: True / False flag to pass to the stored procedure to indicate whether or not the destination table(s) should be truncated.  Upload: N/A |
| Upload.CheckForDuplicates | Download: N/A  Upload: True / False flag indicating whether the upload process should check to see if the supplied records from the database would create duplicates in the SharePoint list. |
| Upload.ReportProcedureName | Download: N/A  Upload: Name of the stored procedure to call to report SharePoint record upload status back to the database.  May include schema name.  Ex: schema.spSampleProcedure |
| Upload.ReportStatusField | Download: N/A  Upload: Name of the field in [TableTypeName] that should be populated with the SharePoint record upload status. |
| Upload.ReportIdField | Download: N/A  Upload: Name of the field in [TableTypeName] that should be populated with the affected SharePoint record ID. Optional. |
| Upload.ReportGuidField | Download: N/A  Upload: Name of the field in [TableTypeName] that should be populated with the affected SharePoint record GUID. Optional. |
| ShareURL | URL location of SharePoint site. This is the base point to the SharePoint physical site URL, not an actual List.  Ex: https://maximus365.sharepoint.com/sites/CCO/Support/QA-OPS/Calibration/ |
| ShareList | Title of the SharePoint list or library, e.g. “BINGO\_London” |
| ShareViewXml | XML Collaborative Application Markup Language (CAML) query to apply to the list to filter it and control record display. Must be escaped for XML.  Ex: &lt;View&gt;&lt;/View&gt; |
| ListFieldsOnly | True / False flag to indicate whether or not the program should do nothing more than print the available fields in the SharePoint List in the email. Intended for debugging / initial development purposes, a value of “true” will not execute the [ShareViewXml] query and will not call the database stored procedure. |
| ColumnTransforms | JSON list of column transformations for data values.  Components:   * + - * ColumnName – SharePoint key name for the field to be searched.       * Find – Value in the field that needs to be replaced if it exists       * Replace – Replacement value   Ex: [{'ColumnName':'Site','Find':'Winchester','Replace':'Winchester - GDIT'}] |
| ExtractFields | JSON list of fields to extract.  Components:   * + - * ColumnName – Destination column name in table type for the SharePoint field.       * KeyName – SharePoint key name for the field.       * Default – Default value for a field if it has the null value. If not supplied or blank, null values are not changed. Download only.       * IsSPKey – Is this field a key field in the SharePoint list? Used to determine updates vs. inserts. TRUE / FALSE. Default = false.       * IsSPUnique – Does this field need to be unique (may be part of a set)? Used to determine duplicate records. TRUE / FALSE. Default = false.       * CanUpdate – Can this field be updated? TRUE / FALSE. Default = false.       * AlwaysUpdate – Only applies if CanUpdate is TRUE for the field. Should the field always be updated, even if the new value is null or blank?       * LookupId – When providing the value of this field, should the lookup identifier be returned as the value instead of the actual lookup value? TRUE / FALSE. Default = false (return the actual lookup value).   Ex: [{'ColumnName':'GUID','KeyName':'GUID','Default':'False'},{'ColumnName':'Job Title','KeyName':'Job\_x0020\_Title','Default':'False', 'IsSPKey':'True','CanUpdate':'False'}] |
| EmailHost | SMTP email host server |
| EmailAuth | Authentication method for SMTP email server. Corresponds to log4net supported SMTP authentication values: Ntlm, Basic, None. |
| EmailFrom | From address for email summary. |
| EmailTo | To address(es) for email summary. |
| ConvertEmptyStringToNull | True to convert empty strings in the SharePoint list to a “null” value; False to leave as empty strings. |
| FailureToken | Unique token string that will appear in the body of failure emails only. This is intended to facilitate email recipient inbox rules where processing failures may need to be routed or handled separately within the recipient’s inbox. |

# Change History

| **Revision** | **Date** | **Change Description** | **Author** |
| --- | --- | --- | --- |
| 1.0 | 12/21/2020 | Initial Revision for CCO eCoaching (Adapted from original design CROP DD CCO\_CROP\_SOIBean\_DD.docx created by Lola Treinen) | Susmitha Palacherla |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |