Word 2010 DepositMO plugin - API reference and source guide

0.1

Generated by Doxygen 1.7.1

Wed Jun 15 2011 08:14:20

Contents

Chapter 1

Namespace Index

1.1 Package List

Here are the packages with brief descriptions (if available):

Chapter 2

Class Index

2.1 Class List

Here are the classes, structs, unions and interfaces with brief descriptions:

Chapter 3

Namespace Documentation

3.1 Package uk

Packages

- package ac

3.2 Package uk.ac

Packages

- package soton

3.3 Package uk.ac.soton

Packages

- package ses

3.4 Package uk.ac.soton.ses

Packages

- package Word2010DepositMOAddIn

  Namespace for Microsoft Word 2010 DepositMO add-in.

Classes

- class HtmlScraper

  Limited basic HTML parsing routines.

- class SwordAtomReader

  SWORD 2 Atom XML reader.

- class SwordClientHandler

  SWORD 2 client implementation for .NET.

- class SwordClientHandlerTests

  A series of basic tests for the .NET API.

- class SwordListAuthor

  SWORD author.

- class SwordListEntry

  SWORD entry for the collection list.

- class SwordListReader

  SWORD list reader.

3.5 Package uk.ac.soton.ses.Word2010DepositMOAddIn

Namespace for Microsoft Word 2010 DepositMO add-in.

Packages

- package Properties

Classes

- class AboutBox

  About Box for Word 2010 DepositMO add-in.

- class DepositMORepositoryControl

  Custom Windows Forms user control for Microsoft Word 2010 DepositMO add-in.

- class LogViewerForm

  Internal log viewer for Microsoft Word 2010 DepositMO add-in.

- class QuickSubmitForm

  Quick submission form for the Microsoft Word 2010 DepositMO add-in.

- class RepositoryGroupUserControl

  Custom Windows Forms user control for repository groups in Microsoft Word 2010 DepositMO add-in.

- class Word2010DepositMOAddIn

  Entry point for the Microsoft Word 2010 DepositMO add-in.

- class Globals
- class ThisRibbonCollection
- class Word2010DepositMORibbon

  Ribbon interface for Microsoft Word 2010 DepositMO add-in.

3.5.1 Detailed Description

Namespace for Microsoft Word 2010 DepositMO add-in.

3.6 Package uk.ac.soton.ses.Word2010DepositMOAddIn.Properties

Classes

- class Resources

  A strongly-typed resource class, for looking up localized strings, etc.

- class Settings

Chapter 4

Class Documentation

4.1 uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox Class Reference

About Box for Word 2010 DepositMO add-in.

Public Member Functions

- AboutBox ()

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Properties

- string AssemblyTitle `[get]`
- string AssemblyVersion `[get]`
- string AssemblyDescription `[get]`
- string AssemblyProduct `[get]`
- string AssemblyCopyright `[get]`
- string AssemblyCompany `[get]`

Private Member Functions

- void okButton_Click (object sender, EventArgs e)
- void InitializeComponent ()

  Required method for Designer support - do not modify the contents of this method with the code editor.

Private Attributes

- System.ComponentModel.IContainer components = null

  Required designer variable.

- System.Windows.Forms.TableLayoutPanel tableLayoutPanel
- System.Windows.Forms.PictureBox logoPictureBox
- System.Windows.Forms.Label labelProductName
- System.Windows.Forms.Label labelVersion
- System.Windows.Forms.Label labelCopyright
- System.Windows.Forms.Label labelCompanyName
- System.Windows.Forms.TextBox textBoxDescription
- System.Windows.Forms.Button okButton

4.1.1 Detailed Description

About Box for Word 2010 DepositMO add-in.

Definition at line 30 of file AboutBox.cs.

4.1.2 Constructor & Destructor Documentation

4.1.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AboutBox ( )

Definition at line 32 of file AboutBox.cs.

```
        {
            InitializeComponent();
            this.Text = String.Format("About {0}", AssemblyTitle);
            this.labelProductName.Text = AssemblyProduct;
            this.labelVersion.Text = String.Format("Version {0}",
  AssemblyVersion);
            this.labelCopyright.Text = AssemblyCopyright;
            this.labelCompanyName.Text = AssemblyCompany;
            this.textBoxDescription.Text = AssemblyDescription;
        }
```

4.1.3 Member Function Documentation

4.1.3.1 override void uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Definition at line 29 of file AboutBox.Designer.cs.

```
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
```

4.1.3.2 void uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.InitializeComponent ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 44 of file AboutBox.Designer.cs.

```
  {
      System.ComponentModel.ComponentResourceManager resources = new System
.ComponentModel.ComponentResourceManager(typeof(AboutBox));
      this.tableLayoutPanel = new System.Windows.Forms.TableLayoutPanel();
      this.logoPictureBox = new System.Windows.Forms.PictureBox();
      this.labelProductName = new System.Windows.Forms.Label();
      this.labelVersion = new System.Windows.Forms.Label();
      this.labelCopyright = new System.Windows.Forms.Label();
      this.labelCompanyName = new System.Windows.Forms.Label();
      this.textBoxDescription = new System.Windows.Forms.TextBox();
      this.okButton = new System.Windows.Forms.Button();
      this.tableLayoutPanel.SuspendLayout();
      ((System.ComponentModel.ISupportInitialize)(this.logoPictureBox)).Beg
inInit();
      this.SuspendLayout();
      //
      // tableLayoutPanel
      //
      this.tableLayoutPanel.ColumnCount = 2;
      this.tableLayoutPanel.ColumnStyles.Add(new System.Windows.Forms.Colum
nStyle(System.Windows.Forms.SizeType.Percent, 33F));
      this.tableLayoutPanel.ColumnStyles.Add(new System.Windows.Forms.Colum
nStyle(System.Windows.Forms.SizeType.Percent, 67F));
      this.tableLayoutPanel.Controls.Add(this.logoPictureBox, 0, 0);
      this.tableLayoutPanel.Controls.Add(this.labelProductName, 1, 0);
      this.tableLayoutPanel.Controls.Add(this.labelVersion, 1, 1);
      this.tableLayoutPanel.Controls.Add(this.labelCopyright, 1, 2);
      this.tableLayoutPanel.Controls.Add(this.labelCompanyName, 1, 3);
      this.tableLayoutPanel.Controls.Add(this.textBoxDescription, 1, 4);
      this.tableLayoutPanel.Controls.Add(this.okButton, 1, 5);
      this.tableLayoutPanel.Dock = System.Windows.Forms.DockStyle.Fill;
      this.tableLayoutPanel.Location = new System.Drawing.Point(9, 9);
      this.tableLayoutPanel.Name = "tableLayoutPanel";
      this.tableLayoutPanel.RowCount = 6;
      this.tableLayoutPanel.RowStyles.Add(new System.Windows.Forms.RowStyle
(System.Windows.Forms.SizeType.Percent, 10F));
      this.tableLayoutPanel.RowStyles.Add(new System.Windows.Forms.RowStyle
(System.Windows.Forms.SizeType.Percent, 10F));
      this.tableLayoutPanel.RowStyles.Add(new System.Windows.Forms.RowStyle
(System.Windows.Forms.SizeType.Percent, 10F));
      this.tableLayoutPanel.RowStyles.Add(new System.Windows.Forms.RowStyle
(System.Windows.Forms.SizeType.Percent, 10F));
      this.tableLayoutPanel.RowStyles.Add(new System.Windows.Forms.RowStyle
(System.Windows.Forms.SizeType.Percent, 50F));
      this.tableLayoutPanel.RowStyles.Add(new System.Windows.Forms.RowStyle
(System.Windows.Forms.SizeType.Percent, 10F));
      this.tableLayoutPanel.Size = new System.Drawing.Size(417, 265);
      this.tableLayoutPanel.TabIndex = 0;
      //
      // logoPictureBox
      //
      this.logoPictureBox.Dock = System.Windows.Forms.DockStyle.Fill;
      this.logoPictureBox.Image = ((System.Drawing.Image)(resources.GetObje
ct("logoPictureBox.Image")));
      this.logoPictureBox.Location = new System.Drawing.Point(3, 3);
      this.logoPictureBox.Name = "logoPictureBox";
      this.tableLayoutPanel.SetRowSpan(this.logoPictureBox, 6);
      this.logoPictureBox.Size = new System.Drawing.Size(131, 259);
      this.logoPictureBox.SizeMode = System.Windows.Forms.PictureBoxSizeMod
e.StretchImage;
      this.logoPictureBox.TabIndex = 12;
      this.logoPictureBox.TabStop = false;
      //
      // labelProductName
      //
      this.labelProductName.Dock = System.Windows.Forms.DockStyle.Fill;
      this.labelProductName.Location = new System.Drawing.Point(143, 0);
      this.labelProductName.Margin = new System.Windows.Forms.Padding(6, 0,
 3, 0);
      this.labelProductName.MaximumSize = new System.Drawing.Size(0, 17);
      this.labelProductName.Name = "labelProductName";
      this.labelProductName.Size = new System.Drawing.Size(271, 17);
      this.labelProductName.TabIndex = 19;
      this.labelProductName.Text = "Product Name";
      this.labelProductName.TextAlign = System.Drawing.ContentAlignment.Mid
dleLeft;
      //
      // labelVersion
      //
      this.labelVersion.Dock = System.Windows.Forms.DockStyle.Fill;
      this.labelVersion.Location = new System.Drawing.Point(143, 26);
      this.labelVersion.Margin = new System.Windows.Forms.Padding(6, 0, 3,
0);
      this.labelVersion.MaximumSize = new System.Drawing.Size(0, 17);
      this.labelVersion.Name = "labelVersion";
      this.labelVersion.Size = new System.Drawing.Size(271, 17);
      this.labelVersion.TabIndex = 0;
      this.labelVersion.Text = "Version";
      this.labelVersion.TextAlign = System.Drawing.ContentAlignment.MiddleL
eft;
      //
      // labelCopyright
      //
      this.labelCopyright.Dock = System.Windows.Forms.DockStyle.Fill;
      this.labelCopyright.Location = new System.Drawing.Point(143, 52);
      this.labelCopyright.Margin = new System.Windows.Forms.Padding(6, 0, 3
, 0);
      this.labelCopyright.MaximumSize = new System.Drawing.Size(0, 17);
      this.labelCopyright.Name = "labelCopyright";
      this.labelCopyright.Size = new System.Drawing.Size(271, 17);
      this.labelCopyright.TabIndex = 21;
      this.labelCopyright.Text = "Copyright";
      this.labelCopyright.TextAlign = System.Drawing.ContentAlignment.Middl
eLeft;
      //
      // labelCompanyName
      //
      this.labelCompanyName.Dock = System.Windows.Forms.DockStyle.Fill;
      this.labelCompanyName.Location = new System.Drawing.Point(143, 78);
      this.labelCompanyName.Margin = new System.Windows.Forms.Padding(6, 0,
 3, 0);
      this.labelCompanyName.MaximumSize = new System.Drawing.Size(0, 17);
      this.labelCompanyName.Name = "labelCompanyName";
      this.labelCompanyName.Size = new System.Drawing.Size(271, 17);
      this.labelCompanyName.TabIndex = 22;
      this.labelCompanyName.Text = "Company Name";
      this.labelCompanyName.TextAlign = System.Drawing.ContentAlignment.Mid
dleLeft;
      //
      // textBoxDescription
      //
      this.textBoxDescription.Dock = System.Windows.Forms.DockStyle.Fill;
      this.textBoxDescription.Location = new System.Drawing.Point(143, 107)
;
      this.textBoxDescription.Margin = new System.Windows.Forms.Padding(6,
3, 3, 3);
      this.textBoxDescription.Multiline = true;
      this.textBoxDescription.Name = "textBoxDescription";
      this.textBoxDescription.ReadOnly = true;
      this.textBoxDescription.ScrollBars = System.Windows.Forms.ScrollBars.
Both;
      this.textBoxDescription.Size = new System.Drawing.Size(271, 126);
      this.textBoxDescription.TabIndex = 23;
      this.textBoxDescription.TabStop = false;
      this.textBoxDescription.Text = "Description";
      //
      // okButton
      //
      this.okButton.Anchor = ((System.Windows.Forms.AnchorStyles)((System.W
indows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Right)));
      this.okButton.DialogResult = System.Windows.Forms.DialogResult.Cancel
;
      this.okButton.Location = new System.Drawing.Point(339, 239);
      this.okButton.Name = "okButton";
      this.okButton.Size = new System.Drawing.Size(75, 23);
      this.okButton.TabIndex = 24;
      this.okButton.Text = "&OK";
      this.okButton.Click += new System.EventHandler(this.okButton_Click);
      //
      // AboutBox
      //
      this.AcceptButton = this.okButton;
      this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
      this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
      this.ClientSize = new System.Drawing.Size(435, 283);
      this.Controls.Add(this.tableLayoutPanel);
      this.FormBorderStyle = System.Windows.Forms.FormBorderStyle.FixedDial
og;
      this.MaximizeBox = false;
      this.MinimizeBox = false;
      this.Name = "AboutBox";
      this.Padding = new System.Windows.Forms.Padding(9);
      this.ShowIcon = false;
      this.ShowInTaskbar = false;
      this.StartPosition = System.Windows.Forms.FormStartPosition.CenterPar
ent;
      this.Text = "AboutBox";
      this.tableLayoutPanel.ResumeLayout(false);
      this.tableLayoutPanel.PerformLayout();
      ((System.ComponentModel.ISupportInitialize)(this.logoPictureBox)).End
Init();
      this.ResumeLayout(false);

  }
```

4.1.3.3 void uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.okButton_Click (object *sender,* EventArgs *e*) `[private]`

Definition at line 123 of file AboutBox.cs.

```
  {
      this.Close();
      this.Dispose();
  }
```

4.1.4 Member Data Documentation

4.1.4.1 System.ComponentModel.IContainer uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.components = null `[private]`

Required designer variable.

Definition at line 24 of file AboutBox.Designer.cs.

4.1.4.2 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.labelCompanyName `[private]`

Definition at line 199 of file AboutBox.Designer.cs.

4.1.4.3 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.labelCopyright `[private]`

Definition at line 198 of file AboutBox.Designer.cs.

4.1.4.4 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.labelProductName `[private]`

Definition at line 196 of file AboutBox.Designer.cs.

4.1.4.5 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.labelVersion `[private]`

Definition at line 197 of file AboutBox.Designer.cs.

4.1.4.6 System.Windows.Forms.PictureBox uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.logoPictureBox `[private]`

Definition at line 195 of file AboutBox.Designer.cs.

4.1.4.7 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.okButton `[private]`

Definition at line 201 of file AboutBox.Designer.cs.

4.1.4.8 System.Windows.Forms.TableLayoutPanel uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.tableLayoutPanel `[private]`

Definition at line 194 of file AboutBox.Designer.cs.

4.1.4.9 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.textBoxDescription `[private]`

Definition at line 200 of file AboutBox.Designer.cs.

4.1.5 Property Documentation

4.1.5.1 string uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AssemblyCompany `[get]`

Definition at line 110 of file AboutBox.cs.

4.1.5.2 string uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AssemblyCopyright `[get]`

Definition at line 97 of file AboutBox.cs.

4.1.5.3 string uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AssemblyDescription `[get]`

Definition at line 71 of file AboutBox.cs.

4.1.5.4 string uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AssemblyProduct `[get]`

Definition at line 84 of file AboutBox.cs.

4.1.5.5 string uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AssemblyTitle `[get]`

Definition at line 46 of file AboutBox.cs.

4.1.5.6 string uk.ac.soton.ses.Word2010DepositMOAddIn.AboutBox.AssemblyVersion `[get]`

Definition at line 63 of file AboutBox.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/AboutBox.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/AboutBox.Designer.cs

4.2 uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryCont Class Reference

Custom Windows Forms user control for Microsoft Word 2010 DepositMO add-in.

Public Member Functions

- DepositMORepositoryControl ()

  Constructor.

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Properties

- internal string username `[get]`

  The username for the repository at the endpoint.

- internal string password `[get]`

  The password for the repository.

- internal string endpoint `[get]`

  The repository's endpoint.

Private Member Functions

- void DepositMORepositoryControl_Load (object sender, EventArgs e)

  OnLoad handler; adds tooltips.

- void submitToThisRepositoryButton_Click (object sender, EventArgs e)

  Event handler for clicking the submit to this repository button.

- void SubmitDocument (bool updateExisting)

  Submits the document to the endpoint.

- void logMessage (string message)

  Logs the supplied message to the submission log text box and the global log message stack.

- internal void SubmitDocument (bool updateExisting, string username, string password, string endpoint)

  Fires the document submission on a separate thread to avoid locking Word's UI whilst the deposition is being negotiated.

- bool _SubmitDocument (bool updateExisting, string username, string password, string endpoint)

  Submits the document to the target endpoint, or updates the document at the target endpoint using the supplied credentials.

- void logException (Exception ex)

Logs the exception
`ex`
to the global log.

- void DumpListingToLog ()

  Dumps the user's repository contents to the log.

- void DumpListingToLog (string username, string password, string endpoint)

  Dumps the user's repository contents to the log.

- void updateEprintButton_Click (object sender, EventArgs e)

  Event handler for the update ePrint button.

- void WarningBox (string message)

  Displays a warning dialogue box with the message
  `message`

- void addRepository (string username, string password, string endpoint)

  Adds a repository to the group.

- void addToGroupButton_Click (object sender, EventArgs e)

  Event handler when the add to group button has been clicked.

- void removeFromGroupButton_Click (object sender, EventArgs e)

  Removes the currently selected tab page from the tab control.

- void submitToAllButton_Click (object sender, EventArgs e)

  Click event handler for submit to all repositories.

- void submitToAll ()

  Iterates across all the tab pages, takes the repository/username/password triples from there and attempts to submit the document to each repository.

- void InitializeComponent ()

  Required method for Designer support - do not modify the contents of this method with the code editor.

Private Attributes

- System.ComponentModel.IContainer components = null

  Required designer variable.

- System.Windows.Forms.GroupBox repositoryControlGroupBox
- System.Windows.Forms.TextBox repositoryLocationTextBox
- System.Windows.Forms.Label repositoryLocationLabel
- System.Windows.Forms.Button submitToThisRepositoryButton
- System.Windows.Forms.TextBox passwordTextBox
- System.Windows.Forms.Label passwordLabel
- System.Windows.Forms.TextBox usernameTextBox
- System.Windows.Forms.Label usernameLabel
- System.Windows.Forms.Button updateEprintButton

- System.Windows.Forms.Button addToGroupButton
- System.Windows.Forms.GroupBox groupRepositoryGroupBox
- System.Windows.Forms.Button removeFromGroupButton
- System.Windows.Forms.TabControl repositoryCollectionTabControl
- System.Windows.Forms.Button submitToAllButton
- System.Windows.Forms.TextBox submissionLogTextBox
- System.Windows.Forms.TextBox documentLocationTextBox
- System.Windows.Forms.Label documentLocationLabel
- System.Windows.Forms.GroupBox messagesGroupBox

Static Private Attributes

- static readonly object lockobj = new object()

  Lock object for threading.

4.2.1 Detailed Description

Custom Windows Forms user control for Microsoft Word 2010 DepositMO add-in.

Definition at line 31 of file DepositMORepositoryControl.cs.

4.2.2 Constructor & Destructor Documentation

4.2.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DepositMORepositoryCont ( )

Constructor.

Sets the default endpoint, username and password

Definition at line 56 of file DepositMORepositoryControl.cs.

```
  {
      InitializeComponent();
      this.repositoryLocationTextBox.Text = Globals.Word2010DepositMOAddIn.
DefaultEndpoint;
      this.usernameTextBox.Text = Globals.Word2010DepositMOAddIn.DefaultUse
rname;
      this.passwordTextBox.Text = Globals.Word2010DepositMOAddIn.DefaultPas
sword;
  }
```

4.2.3 Member Function Documentation

4.2.3.1 bool uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl._SubmitDocument (bool *updateExisting*, string *username*, string *password*, string *endpoint*) `[private]`

Submits the document to the target endpoint, or updates the document at the target endpoint using the supplied credentials.

Parameters

- ***updateExisting*** `true`
  to update an existing document, else
  `false`
- ***username*** The username of the user at the repository
- ***password*** The password of the user at the repository
- ***endpoint*** The endpoint of the repository

Returns

Definition at line 143 of file DepositMORepositoryControl.cs.

References uk.ac.soton.ses.SwordClientHandler.PostDocxToRepository(), and uk.ac.soton.ses.SwordClientHandler.PutToDocument().

```
        {
            // we might not technically have come in on a new thread but to keep
 things consistent we'll
            // increment the counter here, and decrement it at every handled exit
  point of the method
            Globals.Word2010DepositMOAddIn.IncrementSubmissionThreadCount();

            #region Pre-submission checks
            // pre-submission checks
            if (String.IsNullOrWhiteSpace(endpoint))
            {
                this.WarningBox("Attempting to submit document to empty endpoint;
  cowardly refusing to continue");
                Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadCount();
                return false;
            }
            #endregion

            // grab a reference to the active document outside the lock so if som
 eone
            // changes the active document whilst another thread is active it wil
 l
            // be the correct document
            Document d = Globals.Word2010DepositMOAddIn.Application.ActiveDocumen
 t;

            // ensure the user has saved their document before continuing (see
            // read-lock comments below)
            try
            {
                d.Save();
            }
            catch
            {
                // for some reason the standard flow control of Word means that
                // if the user cancels the save dialogue, an exception will be th
 rown
                // --- this is contrary to standard Windows Forms behaviour

                this.WarningBox("Didn't save the document this time; deposition a
 borted");
                Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadCount();
                return false;
            }
```

```
      // this could be spun off on a separate (singleton-type) thread
      // to avoid UI glitches if the remote isn’t responding quickly
      lock (lockobj)
      {
          SwordClientHandler sch = null;
          string originalDocumentName = null;
          string tempFileName = null;
          bool updateResult = false;

          // get the full path and filename of the document and remember th
is
          originalDocumentName = d.FullName;

          // save the file to the temporary path
          tempFileName += Path.GetTempPath() + d.Name;

          try
          {
              // we need to ’save as’ under a different filename as Word ha
s an exclusive
              // read-lock on the current document; ’Type.Missing’ through
the call until
              // we can set the ’add to recent files’ flag to false
              d.SaveAs2(tempFileName, Type.Missing, Type.Missing, Type.Miss
ing, false);

              // we then need to ’save as’ the original filename to restore
 the control
              // to the user
              d.SaveAs2(originalDocumentName);
          }
          catch (COMException cex)
          {
              this.logMessage("A problem occurred with " + cex.Source + " w
hilst trying to save the document");
              this.logException(cex);
              this.logMessage("Couldn’t submit document " + originalDocumen
tName + " this time as " + cex.Source + " will not allow it");

              Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadCount
();
              return false;
          }

          // initialise the SWORD handler
          this.logMessage("Connecting to repository at " + endpoint);
          sch = new SwordClientHandler(username, password, endpoint);

          // use the PUT parts of SWORD if we’re updating
          if (updateExisting)
          {
              // ’endpoint’ in this context should now be the location of t
he document
              #region Update existing document
              try { updateResult = sch.PutToDocument(tempFileName,
endpoint); }
              catch (Exception ex)
              {
                  this.logMessage("Could not update document at " +
endpoint);
                  this.logException(ex);
              }
              finally { File.Delete(tempFileName); }

              if (updateResult)
              {
                  this.logMessage("Successfully updated document at " +
endpoint);
                  Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadC
ount();
                  return true;
              }
              else
              {
                  this.logMessage("Couldn't update document at " +
endpoint);
                  Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadC
ount();
                  return false;
              }
              #endregion
          }
          else
          {
              #region Create new document
              // create new document
              string documentUri = null;

              try { documentUri = sch.PostDocxToRepository(tempFileName); }

              catch (Exception ex)
              {
                  this.logMessage("Could not deposit document to " +
endpoint);
                  this.logException(ex);
              }
              finally { File.Delete(tempFileName); }
              #endregion

              // check for null response. Behaviour at this point is undefi
ned
              if (documentUri == null)
              {
                  this.logMessage("Couldn't understand response from server
 at endpoint " + endpoint + "; please check deposition manually");
                  Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadC
ount();
                  return false;
              }

              #region Response handling on new deposition
              // attempt to parse the response
              if (!String.IsNullOrEmpty(documentUri))
              {
                  this.logMessage("Document deposited at " + documentUri);
                  Globals.Word2010DepositMOAddIn.SetSingleLineTextBox(docum
entUri, this.documentLocationTextBox);
              }
              else
              {
                  this.logMessage("Warning: response from endpoint " +
endpoint + " did not contain a reference to the submitted document");
                  Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadC
ount();
                  return false;
              }
              Globals.Word2010DepositMOAddIn.DecrementSubmissionThreadCount
();
              return true;
              #endregion
          }

      }
  }
```

4.2.3.2 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addRepository (string *username,* string *password,* string *endpoint*) `[private]`

Adds a repository to the group.

Parameters

username The username of the user at the repository

password The password of the user at the repository

endpoint The endpoint of the repository

Definition at line 362 of file DepositMORepositoryControl.cs.

References uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.endpoint, uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.id, uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.password, and uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.username.

```
        {
            // get a unique number for this session so we can grab a handle back
      to it later
            int high_id = 0;
            foreach (TabPage t in this.repositoryCollectionTabControl.TabPages)
            {
                int tabvalue = int.Parse(t.Text);
                if (tabvalue > high_id)
                {
                    high_id = tabvalue;
                }
            }
            int id = high_id + 1;
            string idstring = id.ToString();

            // create a new repository control
            RepositoryGroupUserControl rguc = new RepositoryGroupUserControl();
            rguc.Name = idstring;
            rguc.id = id;
            rguc.username = username;
            rguc.password = password;
            rguc.endpoint = endpoint;

            // create a new tab page with the ID...
            TabPage tp = new TabPage(idstring);

            // ... add the repository control to the page ...
            tp.Controls.Add(rguc);

            // ... and add it to the tab control
            this.repositoryCollectionTabControl.TabPages.Add(tp);
            this.logMessage("Added repository " + rguc.endpoint + " to collection
       with ID " + idstring);
        }
```

4.2.3.3 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addToGroupButton_-Click (object *sender,* EventArgs *e*) `[private]`

Event handler when the add to group button has been clicked.

Parameters

sender Event sender

e Event arguments

Definition at line 401 of file DepositMORepositoryControl.cs.

```
  {
      this.addRepository(this.username, this.password, this.endpoint);
  }
```

4.2.3.4 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DepositMORepositoryCont Load (object *sender,* EventArgs *e*) `[private]`

OnLoad handler; adds tooltips.

Parameters

sender Event sender

e Event parameters

Definition at line 69 of file DepositMORepositoryControl.cs.

```
  {
      new ToolTip().SetToolTip(this.addToGroupButton, "Add these repository
 details to the group to submit documents en masse");
      new ToolTip().SetToolTip(this.documentLocationTextBox, "The location
of the document");
      new ToolTip().SetToolTip(this.passwordTextBox, "The password of the u
ser at the repository");
      new ToolTip().SetToolTip(this.usernameTextBox, "The username of the u
ser at the repository");
      new ToolTip().SetToolTip(this.removeFromGroupButton, "Removes the sel
ected repository’s details from the group");
      new ToolTip().SetToolTip(this.repositoryLocationTextBox, "The locatio
n (endpoint) of the repository");
      new ToolTip().SetToolTip(this.submitToAllButton, "Submits the current
 document to all endpoints in the group, using the credentials specific to each r
epository");
      new ToolTip().SetToolTip(this.submissionLogTextBox, "Information abou
t the submissions (successes, failures, counts and so on)");
      new ToolTip().SetToolTip(this.submitToThisRepositoryButton, "Submits
the current document to the specified endpoint");
      new ToolTip().SetToolTip(this.updateEprintButton, "Updates the docume
nt at the specified endpoint");
  }
```

4.2.3.5 override void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing true if managed resources should be disposed; otherwise, false.

Definition at line 30 of file DepositMORepositoryControl.Designer.cs.

```
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
```

4.2.3.6 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DumpListingToLog ( ) [private]

Dumps the user's repository contents to the log.

Definition at line 303 of file DepositMORepositoryControl.cs.

```
        {
            this.DumpListingToLog(this.username, this.password, this.endpoint);
        }
```

4.2.3.7 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DumpListingToLog (string *username*, string *password*, string *endpoint*) [private]

Dumps the user's repository contents to the log.

Parameters

username Repository username

password Repository password

endpoint Repository endpoint

Definition at line 314 of file DepositMORepositoryControl.cs.

References uk.ac.soton.ses.SwordClientHandler.ListRecords(), uk.ac.soton.ses.SwordListEntry.Title, and uk.ac.soton.ses.SwordListEntry.Updated.

```
        {
            SwordClientHandler sch = new SwordClientHandler(username, password,
      endpoint);
            XmlDocument recordsXml = sch.ListRecords(true);
            if (recordsXml != null)
            {
                SwordListReader slr = new SwordListReader(recordsXml);
                foreach (SwordListEntry sle in slr.Entries)
                {
                    this.logMessage(sle.Title + " [ " + sle.Updated.ToLocalTime()
      .ToLongDateString() + "]");
                }
            }
        }
```

4.2.3.8 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.InitializeComponent ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 45 of file DepositMORepositoryControl.Designer.cs.

```
        {
            this.repositoryControlGroupBox = new System.Windows.Forms.GroupBox();

            this.documentLocationTextBox = new System.Windows.Forms.TextBox();
            this.documentLocationLabel = new System.Windows.Forms.Label();
            this.addToGroupButton = new System.Windows.Forms.Button();
            this.updateEprintButton = new System.Windows.Forms.Button();
            this.passwordTextBox = new System.Windows.Forms.TextBox();
            this.passwordLabel = new System.Windows.Forms.Label();
            this.usernameTextBox = new System.Windows.Forms.TextBox();
            this.usernameLabel = new System.Windows.Forms.Label();
            this.repositoryLocationTextBox = new System.Windows.Forms.TextBox();
            this.repositoryLocationLabel = new System.Windows.Forms.Label();
            this.submitToThisRepositoryButton = new System.Windows.Forms.Button()
;
            this.groupRepositoryGroupBox = new System.Windows.Forms.GroupBox();
            this.submitToAllButton = new System.Windows.Forms.Button();
            this.removeFromGroupButton = new System.Windows.Forms.Button();
            this.repositoryCollectionTabControl = new System.Windows.Forms.TabCon
trol();
            this.submissionLogTextBox = new System.Windows.Forms.TextBox();
            this.messagesGroupBox = new System.Windows.Forms.GroupBox();
            this.repositoryControlGroupBox.SuspendLayout();
            this.groupRepositoryGroupBox.SuspendLayout();
            this.messagesGroupBox.SuspendLayout();
            this.SuspendLayout();
            //
            // repositoryControlGroupBox
            //
            this.repositoryControlGroupBox.Anchor = ((System.Windows.Forms.Anchor
Styles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyl
es.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.repositoryControlGroupBox.Controls.Add(this.
documentLocationTextBox);
            this.repositoryControlGroupBox.Controls.Add(this.
documentLocationLabel);
            this.repositoryControlGroupBox.Controls.Add(this.addToGroupButton);
            this.repositoryControlGroupBox.Controls.Add(this.updateEprintButton);

            this.repositoryControlGroupBox.Controls.Add(this.passwordTextBox);
            this.repositoryControlGroupBox.Controls.Add(this.passwordLabel);
            this.repositoryControlGroupBox.Controls.Add(this.usernameTextBox);
            this.repositoryControlGroupBox.Controls.Add(this.usernameLabel);
            this.repositoryControlGroupBox.Controls.Add(this.
repositoryLocationTextBox);
            this.repositoryControlGroupBox.Controls.Add(this.
repositoryLocationLabel);
            this.repositoryControlGroupBox.Controls.Add(this.
submitToThisRepositoryButton);
            this.repositoryControlGroupBox.Font = new System.Drawing.Font("Tahoma
", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((
byte)(0)));
            this.repositoryControlGroupBox.Location = new System.Drawing.Point(4,
 4);
            this.repositoryControlGroupBox.Name = "repositoryControlGroupBox";
            this.repositoryControlGroupBox.Size = new System.Drawing.Size(200, 22
3);
            this.repositoryControlGroupBox.TabIndex = 0;
            this.repositoryControlGroupBox.TabStop = false;
            this.repositoryControlGroupBox.Text = "Repository Control";
            //
            // documentLocationTextBox
            //
            this.documentLocationTextBox.Anchor = ((System.Windows.Forms.AnchorSt
yles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles
.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.documentLocationTextBox.Font = new System.Drawing.Font("Tahoma",
 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((by
te)(0)));
            this.documentLocationTextBox.Location = new System.Drawing.Point(7, 1
33);
            this.documentLocationTextBox.Name = "documentLocationTextBox";
            this.documentLocationTextBox.Size = new System.Drawing.Size(187, 21);

            this.documentLocationTextBox.TabIndex = 10;
            //
            // documentLocationLabel
            //
            this.documentLocationLabel.AutoSize = true;
            this.documentLocationLabel.Font = new System.Drawing.Font("Tahoma", 8
.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte
)(0)));
            this.documentLocationLabel.Location = new System.Drawing.Point(7, 117
);
            this.documentLocationLabel.Name = "documentLocationLabel";
            this.documentLocationLabel.Size = new System.Drawing.Size(95, 13);
            this.documentLocationLabel.TabIndex = 9;
            this.documentLocationLabel.Text = "Document location";
            //
            // addToGroupButton
            //
            this.addToGroupButton.Font = new System.Drawing.Font("Tahoma", 8.25F,
 System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0))
);
            this.addToGroupButton.Location = new System.Drawing.Point(94, 159);
            this.addToGroupButton.Name = "addToGroupButton";
            this.addToGroupButton.Size = new System.Drawing.Size(82, 52);
            this.addToGroupButton.TabIndex = 8;
            this.addToGroupButton.Text = "Add to repository collection";
            this.addToGroupButton.UseVisualStyleBackColor = true;
            this.addToGroupButton.Click += new System.EventHandler(this.
addToGroupButton_Click);
            //
            // updateEprintButton
            //
            this.updateEprintButton.Font = new System.Drawing.Font("Tahoma", 8.25
F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0
)));
            this.updateEprintButton.Location = new System.Drawing.Point(7, 188);
            this.updateEprintButton.Name = "updateEprintButton";
            this.updateEprintButton.Size = new System.Drawing.Size(76, 23);
            this.updateEprintButton.TabIndex = 7;
            this.updateEprintButton.Text = "Update";
            this.updateEprintButton.UseVisualStyleBackColor = true;
            this.updateEprintButton.Click += new System.EventHandler(this.
updateEprintButton_Click);
            //
            // passwordTextBox
            //
            this.passwordTextBox.Anchor = ((System.Windows.Forms.AnchorStyles)(((
System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.passwordTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
      System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)))
      ;
            this.passwordTextBox.Location = new System.Drawing.Point(69, 89);
            this.passwordTextBox.Name = "passwordTextBox";
            this.passwordTextBox.PasswordChar = '*';
            this.passwordTextBox.Size = new System.Drawing.Size(125, 21);
            this.passwordTextBox.TabIndex = 6;
            //
            // passwordLabel
            //
            this.passwordLabel.AutoSize = true;
            this.passwordLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Sy
      stem.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.passwordLabel.Location = new System.Drawing.Point(9, 92);
            this.passwordLabel.Name = "passwordLabel";
            this.passwordLabel.Size = new System.Drawing.Size(53, 13);
            this.passwordLabel.TabIndex = 5;
            this.passwordLabel.Text = "Password";
            //
            // usernameTextBox
            //
            this.usernameTextBox.Anchor = ((System.Windows.Forms.AnchorStyles)(((
      System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.usernameTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
      System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)))
      ;
            this.usernameTextBox.Location = new System.Drawing.Point(69, 63);
            this.usernameTextBox.Name = "usernameTextBox";
            this.usernameTextBox.Size = new System.Drawing.Size(125, 21);
            this.usernameTextBox.TabIndex = 4;
            //
            // usernameLabel
            //
            this.usernameLabel.AutoSize = true;
            this.usernameLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Sy
      stem.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.usernameLabel.Location = new System.Drawing.Point(8, 66);
            this.usernameLabel.Name = "usernameLabel";
            this.usernameLabel.Size = new System.Drawing.Size(55, 13);
            this.usernameLabel.TabIndex = 3;
            this.usernameLabel.Text = "Username";
            //
            // repositoryLocationTextBox
            //
            this.repositoryLocationTextBox.Anchor = ((System.Windows.Forms.Anchor
      Styles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyl
      es.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.repositoryLocationTextBox.Font = new System.Drawing.Font("Tahoma
      ", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((
      byte)(0)));
            this.repositoryLocationTextBox.Location = new System.Drawing.Point(7,
       37);
            this.repositoryLocationTextBox.Name = "repositoryLocationTextBox";
            this.repositoryLocationTextBox.Size = new System.Drawing.Size(187, 21
      );
            this.repositoryLocationTextBox.TabIndex = 2;
            //
            // repositoryLocationLabel
            //
            this.repositoryLocationLabel.AutoSize = true;
            this.repositoryLocationLabel.Font = new System.Drawing.Font("Tahoma",
       8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((by
      te)(0)));
            this.repositoryLocationLabel.Location = new System.Drawing.Point(7, 2
      0);
            this.repositoryLocationLabel.Name = "repositoryLocationLabel";
            this.repositoryLocationLabel.Size = new System.Drawing.Size(99, 13);
            this.repositoryLocationLabel.TabIndex = 1;
            this.repositoryLocationLabel.Text = "Repository location";
            //
            // submitToThisRepositoryButton
            //
            this.submitToThisRepositoryButton.Font = new System.Drawing.Font("Tah
      oma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point,
       ((byte)(0)));
            this.submitToThisRepositoryButton.Location = new System.Drawing.Point
      (7, 159);
            this.submitToThisRepositoryButton.Name = "submitToThisRepositoryButto
      n";
            this.submitToThisRepositoryButton.Size = new System.Drawing.Size(76,
      23);
            this.submitToThisRepositoryButton.TabIndex = 0;
            this.submitToThisRepositoryButton.Tag = "";
            this.submitToThisRepositoryButton.Text = "Submit";
            this.submitToThisRepositoryButton.UseVisualStyleBackColor = true;
            this.submitToThisRepositoryButton.Click += new System.EventHandler(th
      is.submitToThisRepositoryButton_Click);
            //
            // groupRepositoryGroupBox
            //
            this.groupRepositoryGroupBox.Anchor = ((System.Windows.Forms.AnchorSt
      yles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles
      .Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.groupRepositoryGroupBox.Controls.Add(this.submitToAllButton);
            this.groupRepositoryGroupBox.Controls.Add(this.removeFromGroupButton)
      ;
            this.groupRepositoryGroupBox.Controls.Add(this.
      repositoryCollectionTabControl);
            this.groupRepositoryGroupBox.Font = new System.Drawing.Font("Tahoma",
       8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((by
      te)(0)));
            this.groupRepositoryGroupBox.Location = new System.Drawing.Point(4, 2
      34);
            this.groupRepositoryGroupBox.Name = "groupRepositoryGroupBox";
            this.groupRepositoryGroupBox.Size = new System.Drawing.Size(200, 206)
      ;
            this.groupRepositoryGroupBox.TabIndex = 1;
            this.groupRepositoryGroupBox.TabStop = false;
            this.groupRepositoryGroupBox.Text = "Repository Collection";
            //
            // submitToAllButton
            //
            this.submitToAllButton.Font = new System.Drawing.Font("Tahoma", 8.25F
      , System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)
      ));
            this.submitToAllButton.Location = new System.Drawing.Point(7, 177);
            this.submitToAllButton.Name = "submitToAllButton";
            this.submitToAllButton.Size = new System.Drawing.Size(89, 23);
            this.submitToAllButton.TabIndex = 10;
            this.submitToAllButton.Text = "Submit to all";
            this.submitToAllButton.UseVisualStyleBackColor = true;
            this.submitToAllButton.Click += new System.EventHandler(this.
      submitToAllButton_Click);
            //
            // removeFromGroupButton
            //
            this.removeFromGroupButton.Font = new System.Drawing.Font("Tahoma", 8
      .25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte
      )(0)));
            this.removeFromGroupButton.Location = new System.Drawing.Point(102, 1
      77);
            this.removeFromGroupButton.Name = "removeFromGroupButton";
            this.removeFromGroupButton.Size = new System.Drawing.Size(91, 23);
            this.removeFromGroupButton.TabIndex = 9;
            this.removeFromGroupButton.Text = "Remove";
            this.removeFromGroupButton.UseVisualStyleBackColor = true;
            this.removeFromGroupButton.Click += new System.EventHandler(this.
      removeFromGroupButton_Click);
            //
            // repositoryCollectionTabControl
            //
            this.repositoryCollectionTabControl.Anchor = ((System.Windows.Forms.A
      nchorStyles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.Ancho
      rStyles.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.repositoryCollectionTabControl.Font = new System.Drawing.Font("T
      ahoma", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Poin
      t, ((byte)(0)));
            this.repositoryCollectionTabControl.Location = new System.Drawing.Poi
      nt(7, 20);
            this.repositoryCollectionTabControl.Name = "repositoryCollectionTabCo
      ntrol";
            this.repositoryCollectionTabControl.SelectedIndex = 0;
            this.repositoryCollectionTabControl.Size = new System.Drawing.Size(18
      7, 151);
            this.repositoryCollectionTabControl.TabIndex = 0;
            //
            // submissionLogTextBox
            //
            this.submissionLogTextBox.BorderStyle = System.Windows.Forms.BorderSt
      yle.None;
            this.submissionLogTextBox.Dock = System.Windows.Forms.DockStyle.Fill;

            this.submissionLogTextBox.Font = new System.Drawing.Font("Tahoma", 8.
      25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)
      (0)));
            this.submissionLogTextBox.Location = new System.Drawing.Point(3, 17);

            this.submissionLogTextBox.Multiline = true;
            this.submissionLogTextBox.Name = "submissionLogTextBox";
            this.submissionLogTextBox.ReadOnly = true;
            this.submissionLogTextBox.ScrollBars = System.Windows.Forms.ScrollBar
      s.Vertical;
            this.submissionLogTextBox.Size = new System.Drawing.Size(194, 80);
            this.submissionLogTextBox.TabIndex = 2;
            //
            // messagesGroupBox
            //
            this.messagesGroupBox.Anchor = ((System.Windows.Forms.AnchorStyles)((
      (System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Left)
                        | System.Windows.Forms.AnchorStyles.Right)));
            this.messagesGroupBox.Controls.Add(this.submissionLogTextBox);
            this.messagesGroupBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
       System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0))
      );
            this.messagesGroupBox.Location = new System.Drawing.Point(4, 446);
            this.messagesGroupBox.Name = "messagesGroupBox";
            this.messagesGroupBox.Size = new System.Drawing.Size(200, 100);
            this.messagesGroupBox.TabIndex = 3;
            this.messagesGroupBox.TabStop = false;
            this.messagesGroupBox.Text = "Messages";
            //
            // DepositMORepositoryControl
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.Controls.Add(this.messagesGroupBox);
```

```
            this.Controls.Add(this.groupRepositoryGroupBox);
            this.Controls.Add(this.repositoryControlGroupBox);
            this.Name = "DepositMORepositoryControl";
            this.Size = new System.Drawing.Size(207, 559);
            this.Load += new System.EventHandler(this.
      DepositMORepositoryControl_Load);
            this.repositoryControlGroupBox.ResumeLayout(false);
            this.repositoryControlGroupBox.PerformLayout();
            this.groupRepositoryGroupBox.ResumeLayout(false);
            this.messagesGroupBox.ResumeLayout(false);
            this.messagesGroupBox.PerformLayout();
            this.ResumeLayout(false);

        }
```

4.2.3.9 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.logException (Exception *ex*) `[private]`

Logs the exception

```
ex
```

to the global log.

Parameters

ex Exception to log

Definition at line 294 of file DepositMORepositoryControl.cs.

```
        {
            this.logMessage("Full error was: " + ex.ToString());
            this.logMessage("Please contact your administrator or report this if
      you believe it to be a bug");
        }
```

4.2.3.10 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.logMessage (string *message*) `[private]`

Logs the supplied message to the submission log text box and the global log message stack.

Parameters

message The message to log

Definition at line 113 of file DepositMORepositoryControl.cs.

```
        {
            Globals.Word2010DepositMOAddIn.LogMessage(message, this.
      submissionLogTextBox);
        }
```

4.2.3.11 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.removeFromGroupButton_Click (object *sender*, EventArgs *e*) [private]

Removes the currently selected tab page from the tab control.

Note that at the moment we don't do anything with the repository control underneath (GC?)

Parameters

sender Event sender

e Event arguments

Definition at line 412 of file DepositMORepositoryControl.cs.

```
        {
            if (this.repositoryCollectionTabControl.SelectedTab != null)
            {
                this.repositoryCollectionTabControl.TabPages.Remove(this.
      repositoryCollectionTabControl.SelectedTab);
            }
        }
```

4.2.3.12 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.SubmitDocument (bool *updateExisting*) [private]

Submits the document to the endpoint.

If

```
updateExisting
```

is true, then the endpoint will be taken to be that in the document location text area, otherwise the general repository endpoint will be used

Parameters

updateExisting

Definition at line 99 of file DepositMORepositoryControl.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.submitButton_Click().

```
        {
            string documentEndpoint = this.endpoint;
            if (updateExisting)
            {
                documentEndpoint = this.documentLocationTextBox.Text;
            }
            this.SubmitDocument(updateExisting, this.username, this.password, doc
      umentEndpoint); //this.endpoint);
        }
```

4.2.3.13 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.SubmitDocument (bool *updateExisting*, string *username*, string *password*, string *endpoint*) `[private]`

Fires the document submission on a separate thread to avoid locking Word's UI whilst the deposition is being negotiated.

The thread is issued via a delegate to provide the underlying submission method (

```
_SubmitDocument
```

) with multiple parameters

Parameters

updateExisting `true`
if the endpoint represents an existing document, else

```
false
```

for a new submission

username The username for the repository at

```
endpoint
```

password The password associated with

```
username
```

endpoint The endpoint against which submission should occur

Definition at line 128 of file DepositMORepositoryControl.cs.

```
        {
            Thread submitThread = new Thread(() => this._SubmitDocument(updateExi
    sting, username, password, endpoint));
            submitThread.Start();
        }
```

4.2.3.14 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToAll ( ) `[private]`

Iterates across all the tab pages, takes the repository/username/password triples from there and attempts to submit the document to each repository.

Definition at line 434 of file DepositMORepositoryControl.cs.

References uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.endpoint, uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.password, and uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.username.

```
        {
            int submissionCount = 0;
            foreach (TabPage tabPage in this.repositoryCollectionTabControl.TabPa
    ges)
            {
                RepositoryGroupUserControl rguc = (RepositoryGroupUserControl)tab
    Page.Controls[0];
```

```
            this.SubmitDocument(false, rguc.username, rguc.password, rguc.end
     point);
            submissionCount++;
        }
        this.logMessage("Submitted document to " + submissionCount.ToString()
      + " repositories");
    }
```

4.2.3.15 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToAllButton_Click (object *sender,* EventArgs *e*) [private]

Click event handler for submit to all repositories.

Parameters

sender

e

Definition at line 425 of file DepositMORepositoryControl.cs.

```
    {
        this.submitToAll();
    }
```

4.2.3.16 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToThisRepositoryB Click (object *sender,* EventArgs *e*) [private]

Event handler for clicking the submit to this repository button.

Parameters

sender Event sender

e Event arguments

Definition at line 88 of file DepositMORepositoryControl.cs.

```
    {
        this.SubmitDocument(false);
    }
```

4.2.3.17 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.updateEprintButton_Click (object *sender,* EventArgs *e*) [private]

Event handler for the update ePrint button.

Parameters

sender Event sender

e Event arguments

Definition at line 333 of file DepositMORepositoryControl.cs.

```
    {
        if (this.documentLocationTextBox.Text == String.Empty || this.documen
tLocationTextBox.Text == "")
        {
            //this.logMessage("No document location provided; refusing to upd
ate");
            this.WarningBox("No document location has been provided against w
hich to update. Perhaps you meant to submit rather than update?");
        }
        else
        {
            this.SubmitDocument(true);
        }
    }
```

4.2.3.18 void uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.WarningBox (string *message*) [private]

Displays a warning dialogue box with the message

`message`

Parameters

message The message to display

Definition at line 350 of file DepositMORepositoryControl.cs.

```
    {
        MessageBox.Show(message, "DepositMO", MessageBoxButtons.OK, MessageBo
xIcon.Warning);
        this.logMessage(message);
    }
```

4.2.4 Member Data Documentation

4.2.4.1 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addToGroupButton [private]

Definition at line 312 of file DepositMORepositoryControl.Designer.cs.

4.2.4.2 System.ComponentModel.IContainer uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.components = null [private]

Required designer variable.

Definition at line 24 of file DepositMORepositoryControl.Designer.cs.

4.2.4.3 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.documentLocationLabel `[private]`

Definition at line 319 of file DepositMORepositoryControl.Designer.cs.

4.2.4.4 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.documentLocationTextBox `[private]`

Definition at line 318 of file DepositMORepositoryControl.Designer.cs.

4.2.4.5 System.Windows.Forms.GroupBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.groupRepositoryGroupBox `[private]`

Definition at line 313 of file DepositMORepositoryControl.Designer.cs.

4.2.4.6 readonly object uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.lockobj = new object() `[static, private]`

Lock object for threading.

Definition at line 51 of file DepositMORepositoryControl.cs.

4.2.4.7 System.Windows.Forms.GroupBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.messagesGroupBox `[private]`

Definition at line 320 of file DepositMORepositoryControl.Designer.cs.

4.2.4.8 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.passwordLabel `[private]`

Definition at line 308 of file DepositMORepositoryControl.Designer.cs.

4.2.4.9 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.passwordTextBox `[private]`

Definition at line 307 of file DepositMORepositoryControl.Designer.cs.

4.2.4.10 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.removeFromGroupButton `[private]`

Definition at line 314 of file DepositMORepositoryControl.Designer.cs.

4.2.4.11 System.Windows.Forms.TabControl uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.repositoryCollectionTabCo `[private]`

Definition at line 315 of file DepositMORepositoryControl.Designer.cs.

4.2.4.12 System.Windows.Forms.GroupBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.repositoryControlGroupBo `[private]`

Definition at line 303 of file DepositMORepositoryControl.Designer.cs.

4.2.4.13 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.repositoryLocationLabel `[private]`

Definition at line 305 of file DepositMORepositoryControl.Designer.cs.

4.2.4.14 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.repositoryLocationTextBox `[private]`

Definition at line 304 of file DepositMORepositoryControl.Designer.cs.

4.2.4.15 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submissionLogTextBox `[private]`

Definition at line 317 of file DepositMORepositoryControl.Designer.cs.

4.2.4.16 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToAllButton `[private]`

Definition at line 316 of file DepositMORepositoryControl.Designer.cs.

4.2.4.17 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToThisRepositoryB `[private]`

Definition at line 306 of file DepositMORepositoryControl.Designer.cs.

4.2.4.18 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.updateEprintButton `[private]`

Definition at line 311 of file DepositMORepositoryControl.Designer.cs.

4.2.4.19 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.usernameLabel `[private]`

Definition at line 310 of file DepositMORepositoryControl.Designer.cs.

4.2.4.20 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.usernameTextBox `[private]`

Definition at line 309 of file DepositMORepositoryControl.Designer.cs.

4.2.5 Property Documentation

4.2.5.1 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.endpoint `[get, private]`

The repository's endpoint.

Definition at line 46 of file DepositMORepositoryControl.cs.

4.2.5.2 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.password `[get, private]`

The password for the repository.

Definition at line 41 of file DepositMORepositoryControl.cs.

4.2.5.3 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.username `[get, private]`

The username for the repository at the endpoint.

Definition at line 36 of file DepositMORepositoryControl.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/DepositMORepositoryControl.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/DepositMORepositoryControl.Desig

4.3 uk.ac.soton.ses.Word2010DepositMOAddIn.Globals Class Reference

Properties

- static internal Word2010DepositMOAddIn Word2010DepositMOAddIn `[get, set]`

- static internal global::Microsoft.Office.Tools.Word.ApplicationFactory Factory [get, set]
- static internal ThisRibbonCollection Ribbons [get]

Private Member Functions

- Globals ()

Static Private Attributes

- static Word2010DepositMOAddIn _Word2010DepositMOAddIn
- static global::Microsoft.Office.Tools.Word.ApplicationFactory _factory
- static ThisRibbonCollection _ThisRibbonCollection

4.3.1 Detailed Description

Definition at line 177 of file Word2010DepositMOAddIn.Designer.cs.

4.3.2 Constructor & Destructor Documentation

4.3.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.Globals.Globals ( ) [private]

Definition at line 180 of file Word2010DepositMOAddIn.Designer.cs.

```
                                        {
        }
```

4.3.3 Member Data Documentation

4.3.3.1 global::Microsoft.Office.Tools.Word.ApplicationFactory uk.ac.soton.ses.Word2010DepositMOAddIn.Globals._factory [static, private]

Definition at line 185 of file Word2010DepositMOAddIn.Designer.cs.

4.3.3.2 ThisRibbonCollection uk.ac.soton.ses.Word2010DepositMOAddIn.Globals._ThisRibbonCollection [static, private]

Definition at line 187 of file Word2010DepositMOAddIn.Designer.cs.

4.3.3.3 Word2010DepositMOAddIn uk.ac.soton.ses.Word2010DepositMOAddIn.Globals._Word2010DepositMOAddIn [static, private]

Definition at line 183 of file Word2010DepositMOAddIn.Designer.cs.

4.3.4 Property Documentation

4.3.4.1 internal global::Microsoft.Office.Tools.Word.ApplicationFactory uk.ac.soton.ses.Word2010DepositMOAddIn.Globals.Factory `[static, get, set, private]`

Definition at line 204 of file Word2010DepositMOAddIn.Designer.cs.

4.3.4.2 internal ThisRibbonCollection uk.ac.soton.ses.Word2010DepositMOAddIn.Globals.Ribbons `[static, get, private]`

Definition at line 218 of file Word2010DepositMOAddIn.Designer.cs.

4.3.4.3 internal Word2010DepositMOAddIn uk.ac.soton.ses.Word2010DepositMOAddIn.Globals.Word2010DepositMOAddIn `[static, get, set, private]`

Definition at line 190 of file Word2010DepositMOAddIn.Designer.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMOAddIn.Designe

4.4 uk.ac.soton.ses.HtmlScraper Class Reference

Limited basic HTML parsing routines.

Static Public Member Functions

- static string GetMetaContent (string address, string name)
  Returns the content metatag value for the supplied parameter name and web address.
- static string GetMetaContent (string address, string name, string target)
  Returns the value of the named attribute from a metatag if this exists.
- static string GetAttributeContent (string address, string type, string namelabel, string name, string target)
  Returns the value of the named attribute from the supplied address.
- static string GetAttributeContent (HttpWebRequest request, string type, string namelabel, string name, string target)
  Gathers values from an HTML file's attribute where another named attribute matches another value.

4.4.1 Detailed Description

Limited basic HTML parsing routines.

Definition at line 30 of file HtmlScraper.cs.

4.4.2 Member Function Documentation

4.4.2.1 static string uk.ac.soton.ses.HtmlScraper.GetAttributeContent (string *address*, string *type*, string *namelabel*, string *name*, string *target*) `[static]`

Returns the value of the named attribute from the supplied address.

For example, given:

link rel="foo" href="bar"

Parameters

address Web address containing HTML to be parsed

type Type of tag, e.g. link

namelabel Label of the name, e.g. rel

name Name, e.g. foo

target Target label, e.g. href

Returns

Target value, e.g. bar

Definition at line 66 of file HtmlScraper.cs.

```
        {
            return GetAttributeContent((HttpWebRequest)HttpWebRequest.Create(addr
      ess), type, namelabel, name, target);
        }
```

4.4.2.2 static string uk.ac.soton.ses.HtmlScraper.GetAttributeContent (HttpWebRequest *request*, string *type*, string *namelabel*, string *name*, string *target*) `[static]`

Gathers values from an HTML file's attribute where another named attribute matches another value.

For example, given:

link rel="foo" href="bar"

as HTML content from `http://www.example.com,` one can retrieve the value 'bar' by calling:

GetAttributeContent((HttpWebRequest)HttpWebRequest.Create("http://www.example.com"), "link", "rel", "foo", "href").

Note that this method hasn't been exhaustively tested but appears to work fairly well even for relatively broken HTML 4 documents (as well as nice, strict XHTML ones). Additionally, as the whole HTML document is loaded into RAM whilst parsing (in order to ditch and close the request/responses early) a truly massive HTML document could∗ cause a

```
OutOfMemoryException
```

.

All exceptions are caught, outputted to any debug listener, and rethrown

It is not optimised and could contain bugs, so please use with caution!

Parameters

request The HTTP web request from which to gather an HTML response
type The type of tag (e.g. "meta" or "link")
namelabel The label of the tag to match
name The value of the labelled tag to match
target The label of the target tag

Returns

The value of the target tag

Definition at line 95 of file HtmlScraper.cs.

```
    {
        string contentvalue = null;
        string responsestring = null;

        // return early if the input is bad
        if (type == null || namelabel == null || name == null || target == nu
  ll || request == null)
        {
            return null;
        }

        string targettagtype = type.ToLower();
        string targetnamelabel = namelabel.ToLower();

        try
        {
            // first, get the page code into a string for parsing

            using (HttpWebResponse resp = (HttpWebResponse)request.GetRespons
  e())
            {
                using (TextReader tr = new StreamReader(resp.GetResponseStrea
  m()))
                {
                    responsestring = tr.ReadToEnd();
                }
            }

            // request doesn't close and response should be already closed

            for (int l = 0; l < responsestring.Length - targettagtype.Length;
   l++)
            {
                // do we have an opening tag?
                if (responsestring[l].Equals('<'))
                {
                    // gather the substring to see if this tag is a meta
                    string substring = responsestring.Substring(l, targettagt
  ype.Length + 1);
                    if (substring.ToLower().Equals("<" + targettagtype))
                    {
                        // find end
                        int endpoint = -1;
                        for (int e = l; e < responsestring.Length; e++)
                        {
                            if (responsestring[e].Equals('>'))
                            {
                                endpoint = e;
                                break;
                            }
```

```
                      }

                      // we successfully found the end of the meta line(s)
                      if (endpoint > -1)
                      {
                          // isolate this meta line
                          string metaline = responsestring.Substring(l, end
point - l + 1);

                          // find the ’name’ part
                          int nameIndex = metaline.ToLower().IndexOf(" " +
targetnamelabel);
                          if (nameIndex > -1)
                          {
                              int startchar = -1;
                              int endchar = -1;
                              for (int i = nameIndex; i < metaline.Length;
i++)
                              {
                                  if (metaline[i] == ’"’ || metaline[i] ==
’\’’)
                                  {
                                      if (startchar == -1)
                                      {
                                          startchar = i;
                                      }
                                      else
                                      {
                                          endchar = i;
                                      }
                                  }

                                  if (startchar > -1 && endchar > -1)
                                  {
                                      break;
                                  }
                              }

                              // this should be the name of the meta tag
                              string namevalue = metaline.Substring(startch
ar + 1, endchar - (startchar + 1));
                              Debug.WriteLine(String.Format("Found tag type
 {0} with name {1}", targetnamelabel, namevalue));

                              // is it the one we want?
                              if (namevalue.ToLower().Equals(name.ToLower()
))
                              {
                                  Debug.WriteLine(String.Format("Correct li
ne found: {0}", metaline));

                                  int contentstart = -1;
                                  int contentend = -1;
                                  int contentindex = -1;

                                  // ok, now look for the attribute we want
 the value for inside this meta
                                  contentindex = metaline.IndexOf(" " + tar
get);
                                  if (contentindex > -1)
                                  {
                                      for (int i = contentindex; i < metali
ne.Length; i++)
                                      {
                                          if (metaline[i] == ’"’ || metalin
e[i] == ’\’’)
                                          {
                                              if (contentstart == -1)
                                              {
                                                  contentstart = i;
                                              }
                                              else
                                              {
                                                  contentend = i;
                                              }
                                          }

                                          if (contentstart > -1 && contente
nd > -1)
                                          {
                                              break;
                                          }
                                      }

                                      if (contentstart != -1 && contentend
!= -1)
                                      {
                                          // match found for the contents o
f the attribute
                                          contentvalue = metaline.Substring
(contentstart + 1, contentend - (contentstart + 1));
                                          Debug.WriteLine(String.Format("Co
ntent value is {0}", contentvalue));
                                          break;
                                      }
                                  }
                              }
                          }
                      }
                  }
              }
          }
      }
      catch (Exception ex)
      {
          Debug.WriteLine(ex);
          throw ex;
      }

      return contentvalue == null ? null : contentvalue.Trim();
  }
```

4.4.2.3 static string uk.ac.soton.ses.HtmlScraper.GetMetaContent (string *address,* string *name*) `[static]`

Returns the content metatag value for the supplied parameter name and web address.

Parameters

address Web address containing HTML to be parsed

name Name of the content parameter

Returns

Definition at line 38 of file HtmlScraper.cs.

```
        {
            return GetMetaContent(address, name, "content");
        }
```

4.4.2.4 static string uk.ac.soton.ses.HtmlScraper.GetMetaContent (string *address,* string *name,* string *target*) `[static]`

Returns the value of the named attribute from a metatag if this exists.

Parameters

address Web address containing HTML to be parsed

name Name of the content parameter (e.g. foo if name="foo")

target Name of the target attribute (e.g. value if value="bar")

Returns

Value of the target attribute (e.g. bar if value="bar")

Definition at line 50 of file HtmlScraper.cs.

```
        {
            return GetAttributeContent((HttpWebRequest)HttpWebRequest.Create(addr
      ess), "meta", "name", name, target);
        }
```

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/HtmlScraper.cs

4.5 uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm Class Reference

Internal log viewer for Microsoft Word 2010 DepositMO add-in.

Public Member Functions

- LogViewerForm ()

  Windows form for viewing the log.

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Private Member Functions

- void closeButton_Click (object sender, EventArgs e)

  Event triggered by clicking the log viewer's close button.

- void copyButton_Click (object sender, EventArgs e)

  Copies the contents of the log text box to the clipboard.

- void InitializeComponent ()

  Required method for Designer support - do not modify the contents of this method with the code editor.

Private Attributes

- System.ComponentModel.IContainer components = null

  Required designer variable.

- System.Windows.Forms.Button closeButton
- System.Windows.Forms.TextBox logTextBox
- System.Windows.Forms.Button copyButton

4.5.1 Detailed Description

Internal log viewer for Microsoft Word 2010 DepositMO add-in.

Definition at line 25 of file LogViewerForm.cs.

4.5.2 Constructor & Destructor Documentation

4.5.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.LogViewerForm ( )

Windows form for viewing the log.

Definition at line 30 of file LogViewerForm.cs.

```
        {
            InitializeComponent();
            // set the contents of the text box to be the singleton's log message
 stack
            this.logTextBox.Lines = Globals.Word2010DepositMOAddIn.GetAllLogMessa
ges();
        }
```

4.5.3 Member Function Documentation

4.5.3.1 void uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.closeButton_Click (object *sender*, EventArgs *e*) `[private]`

Event triggered by clicking the log viewer's close button.

Parameters

sender Event sender

e Event arguments

Definition at line 42 of file LogViewerForm.cs.

```
{
    this.Close();
    this.Dispose();
}
```

4.5.3.2 void uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.copyButton_Click (object *sender*, EventArgs *e*) `[private]`

Copies the contents of the log text box to the clipboard.

Parameters

sender

e

Definition at line 53 of file LogViewerForm.cs.

```
{
    Clipboard.SetText(this.logTextBox.Text);
}
```

4.5.3.3 override void uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing true if managed resources should be disposed; otherwise, false.

Definition at line 30 of file LogViewerForm.Designer.cs.

```
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}
```

4.5.3.4 void uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.InitializeComponent ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 45 of file LogViewerForm.Designer.cs.

```
        {
            this.closeButton = new System.Windows.Forms.Button();
            this.logTextBox = new System.Windows.Forms.TextBox();
            this.copyButton = new System.Windows.Forms.Button();
            this.SuspendLayout();
            //
            // closeButton
            //
            this.closeButton.Anchor = ((System.Windows.Forms.AnchorStyles)((Syste
m.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Right)));

            this.closeButton.Font = new System.Drawing.Font("Tahoma", 8.25F);
            this.closeButton.Location = new System.Drawing.Point(364, 264);
            this.closeButton.Name = "closeButton";
            this.closeButton.Size = new System.Drawing.Size(75, 23);
            this.closeButton.TabIndex = 0;
            this.closeButton.Text = "Close";
            this.closeButton.UseVisualStyleBackColor = true;
            this.closeButton.Click += new System.EventHandler(this.
closeButton_Click);
            //
            // logTextBox
            //
            this.logTextBox.Anchor = ((System.Windows.Forms.AnchorStyles)((((Syst
em.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyles.Bottom)
                  | System.Windows.Forms.AnchorStyles.Left)
                  | System.Windows.Forms.AnchorStyles.Right)));
            this.logTextBox.BorderStyle = System.Windows.Forms.BorderStyle.None;
            this.logTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F);
            this.logTextBox.Location = new System.Drawing.Point(12, 12);
            this.logTextBox.Multiline = true;
            this.logTextBox.Name = "logTextBox";
            this.logTextBox.ReadOnly = true;
            this.logTextBox.ScrollBars = System.Windows.Forms.ScrollBars.Vertical
;
            this.logTextBox.Size = new System.Drawing.Size(427, 246);
            this.logTextBox.TabIndex = 1;
            //
            // copyButton
            //
            this.copyButton.Anchor = ((System.Windows.Forms.AnchorStyles)((System
.Windows.Forms.AnchorStyles.Bottom | System.Windows.Forms.AnchorStyles.Left)));
            this.copyButton.Font = new System.Drawing.Font("Tahoma", 8.25F);
            this.copyButton.Location = new System.Drawing.Point(12, 265);
            this.copyButton.Name = "copyButton";
            this.copyButton.Size = new System.Drawing.Size(75, 23);
            this.copyButton.TabIndex = 2;
            this.copyButton.Text = "Copy";
            this.copyButton.UseVisualStyleBackColor = true;
            this.copyButton.Click += new System.EventHandler(this.
copyButton_Click);
            //
            // LogViewerForm
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.ClientSize = new System.Drawing.Size(451, 299);
            this.Controls.Add(this.copyButton);
            this.Controls.Add(this.logTextBox);
            this.Controls.Add(this.closeButton);
            this.Name = "LogViewerForm";
            this.Text = "DepositMO message stack";
            this.ResumeLayout(false);
            this.PerformLayout();

        }
```

4.5.4 Member Data Documentation

4.5.4.1 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.closeButton [private]

Definition at line 108 of file LogViewerForm.Designer.cs.

4.5.4.2 System.ComponentModel.IContainer uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.components = null [private]

Required designer variable.

Definition at line 24 of file LogViewerForm.Designer.cs.

4.5.4.3 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.copyButton [private]

Definition at line 110 of file LogViewerForm.Designer.cs.

4.5.4.4 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.LogViewerForm.logTextBox [private]

Definition at line 109 of file LogViewerForm.Designer.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/LogViewerForm.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/LogViewerForm.Designer.cs

4.6 uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm Class Reference

Quick submission form for the Microsoft Word 2010 DepositMO add-in.

Public Member Functions

- QuickSubmitForm ()

    Constructor for quick submission Windows form.

Protected Member Functions

- override void Dispose (bool disposing)

    Clean up any resources being used.

Private Member Functions

- void cancelButton_Click (object sender, EventArgs e)
- void submitButton_Click (object sender, EventArgs e)
- void InitializeComponent ()

    Required method for Designer support - do not modify the contents of this method with the code editor.

Private Attributes

- System.ComponentModel.IContainer components = null

    Required designer variable.

- System.Windows.Forms.Button cancelButton
- System.Windows.Forms.TextBox passwordTextBox
- System.Windows.Forms.Label passwordLabel
- System.Windows.Forms.TextBox usernameTextBox
- System.Windows.Forms.Label usernameLabel
- System.Windows.Forms.TextBox repositoryLocationTextBox
- System.Windows.Forms.Label repositoryLocationLabel
- System.Windows.Forms.Button submitButton

4.6.1 Detailed Description

Quick submission form for the Microsoft Word 2010 DepositMO add-in.

Definition at line 32 of file QuickSubmitForm.cs.

4.6.2 Constructor & Destructor Documentation

4.6.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.QuickSubmitForm ( )

Constructor for quick submission Windows form.

Definition at line 37 of file QuickSubmitForm.cs.

```
    {
        InitializeComponent();
        this.repositoryLocationTextBox.Text = Globals.Ribbons.GetRibbon<Word2
010DepositMORibbon>().endpointEditBox.Text;

        if (!String.IsNullOrEmpty(Globals.Ribbons.GetRibbon<Word2010DepositMO
Ribbon>().quickUsername))
        {
            this.usernameTextBox.Text = Globals.Ribbons.GetRibbon<Word2010Dep
ositMORibbon>().quickUsername;
        }

        if (!String.IsNullOrEmpty(Globals.Ribbons.GetRibbon<Word2010DepositMO
Ribbon>().quickPassword))
        {
            this.passwordTextBox.Text = Globals.Ribbons.GetRibbon<Word2010Dep
ositMORibbon>().quickPassword;
        }
    }
```

4.6.3 Member Function Documentation

4.6.3.1 void uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.cancelButton_Click (object *sender,* EventArgs *e*) [private]

Definition at line 53 of file QuickSubmitForm.cs.

```
    {
        this.Close();
        this.Dispose();
    }
```

4.6.3.2 override void uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.Dispose (bool *disposing*) [protected]

Clean up any resources being used.

Parameters

disposing true if managed resources should be disposed; otherwise, false.

Definition at line 29 of file QuickSubmitForm.Designer.cs.

```
    {
        if (disposing && (components != null))
        {
            components.Dispose();
        }
        base.Dispose(disposing);
    }
```

4.6.3.3 void uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.InitializeComponent ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 44 of file QuickSubmitForm.Designer.cs.

```
  {
      this.cancelButton = new System.Windows.Forms.Button();
      this.passwordTextBox = new System.Windows.Forms.TextBox();
      this.passwordLabel = new System.Windows.Forms.Label();
      this.usernameTextBox = new System.Windows.Forms.TextBox();
      this.usernameLabel = new System.Windows.Forms.Label();
      this.repositoryLocationTextBox = new System.Windows.Forms.TextBox();
      this.repositoryLocationLabel = new System.Windows.Forms.Label();
      this.submitButton = new System.Windows.Forms.Button();
      this.SuspendLayout();
      //
      // cancelButton
      //
      this.cancelButton.Location = new System.Drawing.Point(402, 55);
      this.cancelButton.Name = "cancelButton";
      this.cancelButton.Size = new System.Drawing.Size(75, 23);
      this.cancelButton.TabIndex = 0;
      this.cancelButton.Text = "Cancel";
      this.cancelButton.UseVisualStyleBackColor = true;
      this.cancelButton.Click += new System.EventHandler(this.
cancelButton_Click);
      //
      // passwordTextBox
      //
      this.passwordTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)))
;
      this.passwordTextBox.Location = new System.Drawing.Point(125, 62);
      this.passwordTextBox.Name = "passwordTextBox";
      this.passwordTextBox.PasswordChar = ’*’;
      this.passwordTextBox.Size = new System.Drawing.Size(100, 21);
      this.passwordTextBox.TabIndex = 12;
      //
      // passwordLabel
      //
      this.passwordLabel.AutoSize = true;
      this.passwordLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Sy
stem.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
      this.passwordLabel.Location = new System.Drawing.Point(65, 65);
      this.passwordLabel.Name = "passwordLabel";
      this.passwordLabel.Size = new System.Drawing.Size(53, 13);
      this.passwordLabel.TabIndex = 11;
      this.passwordLabel.Text = "Password";
      //
      // usernameTextBox
      //
      this.usernameTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)))
;
      this.usernameTextBox.Location = new System.Drawing.Point(125, 36);
      this.usernameTextBox.Name = "usernameTextBox";
      this.usernameTextBox.Size = new System.Drawing.Size(100, 21);
      this.usernameTextBox.TabIndex = 10;
      //
      // usernameLabel
      //
      this.usernameLabel.AutoSize = true;
      this.usernameLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Sy
stem.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
      this.usernameLabel.Location = new System.Drawing.Point(64, 39);
      this.usernameLabel.Name = "usernameLabel";
      this.usernameLabel.Size = new System.Drawing.Size(55, 13);
      this.usernameLabel.TabIndex = 9;
      this.usernameLabel.Text = "Username";
      //
      // repositoryLocationTextBox
      //
      this.repositoryLocationTextBox.Anchor = ((System.Windows.Forms.Anchor
Styles)(((System.Windows.Forms.AnchorStyles.Top | System.Windows.Forms.AnchorStyl
es.Left)
                  | System.Windows.Forms.AnchorStyles.Right)));
      this.repositoryLocationTextBox.Font = new System.Drawing.Font("Tahoma
", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((
byte)(0)));
      this.repositoryLocationTextBox.Location = new System.Drawing.Point(12
5, 9);
      this.repositoryLocationTextBox.Name = "repositoryLocationTextBox";
      this.repositoryLocationTextBox.Size = new System.Drawing.Size(354, 21
);
      this.repositoryLocationTextBox.TabIndex = 8;
      //
      // repositoryLocationLabel
      //
      this.repositoryLocationLabel.AutoSize = true;
      this.repositoryLocationLabel.Font = new System.Drawing.Font("Tahoma",
 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((by
te)(0)));
      this.repositoryLocationLabel.Location = new System.Drawing.Point(20,
12);
      this.repositoryLocationLabel.Name = "repositoryLocationLabel";
      this.repositoryLocationLabel.Size = new System.Drawing.Size(99, 13);
      this.repositoryLocationLabel.TabIndex = 7;
      this.repositoryLocationLabel.Text = "Repository location";
      //
      // submitButton
      //
      this.submitButton.Location = new System.Drawing.Point(321, 55);
      this.submitButton.Name = "submitButton";
      this.submitButton.Size = new System.Drawing.Size(75, 23);
      this.submitButton.TabIndex = 13;
      this.submitButton.Text = "Submit";
      this.submitButton.UseVisualStyleBackColor = true;
      this.submitButton.Click += new System.EventHandler(this.
submitButton_Click);
      //
      // QuickSubmitForm
      //
      this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
      this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
      this.ClientSize = new System.Drawing.Size(491, 95);
      this.Controls.Add(this.submitButton);
      this.Controls.Add(this.passwordTextBox);
      this.Controls.Add(this.passwordLabel);
      this.Controls.Add(this.usernameTextBox);
      this.Controls.Add(this.usernameLabel);
      this.Controls.Add(this.repositoryLocationTextBox);
      this.Controls.Add(this.repositoryLocationLabel);
      this.Controls.Add(this.cancelButton);
      this.Name = "QuickSubmitForm";
      this.Text = "DepositMO quick submission";
      this.ResumeLayout(false);
      this.PerformLayout();

  }
```

4.6.3.4 void uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.submitButton_Click (object *sender,* EventArgs *e*) `[private]`

Definition at line 59 of file QuickSubmitForm.cs.

References uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.SubmitDocument().

```
  {
      CustomTaskPaneCollection ctps = Globals.Word2010DepositMOAddIn.Custom
TaskPanes;
      if (ctps.Count == 0)
      {
          // abort
          this.Close();
          this.Dispose();
          return;
      }

      CustomTaskPane currentTaskPane = null;

      foreach (CustomTaskPane ctp in ctps)
      {
          if (ctp.Window == Globals.Word2010DepositMOAddIn.Application.Acti
veWindow)
          {
              currentTaskPane = ctp;
          }
      }

      if (currentTaskPane == null)
      {
          // abort
          MessageBox.Show("Could not get the correct task pane");
          this.Close();
          this.Dispose();
          return;
      }

      if(currentTaskPane.Control is DepositMORepositoryControl)

      {
          DepositMORepositoryControl dmrControl = (DepositMORepositoryContr
ol)currentTaskPane.Control;
          dmrControl.SubmitDocument(false, this.usernameTextBox.Text, this.
passwordTextBox.Text, this.repositoryLocationTextBox.Text);
      }
      else
      {
          MessageBox.Show("Couldn’t get control over the repository panel")
;
      }

      // update the internal fields on the ribbon if applicable

      if (!String.IsNullOrEmpty(this.usernameTextBox.Text))
      {
          Globals.Ribbons.GetRibbon<Word2010DepositMORibbon>().quickUsernam
e = this.usernameTextBox.Text;
      }

      if (!String.IsNullOrEmpty(this.passwordTextBox.Text))
      {
          Globals.Ribbons.GetRibbon<Word2010DepositMORibbon>().quickPasswor
d = this.passwordTextBox.Text;
      }

      this.Close();
```

```
            this.Dispose();
        }
```

4.6.4 Member Data Documentation

4.6.4.1 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.cancelButton `[private]`

Definition at line 155 of file QuickSubmitForm.Designer.cs.

4.6.4.2 System.ComponentModel.IContainer uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.components = null `[private]`

Required designer variable.

Definition at line 23 of file QuickSubmitForm.Designer.cs.

4.6.4.3 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.passwordLabel `[private]`

Definition at line 157 of file QuickSubmitForm.Designer.cs.

4.6.4.4 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.passwordTextBox `[private]`

Definition at line 156 of file QuickSubmitForm.Designer.cs.

4.6.4.5 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.repositoryLocationLabel `[private]`

Definition at line 161 of file QuickSubmitForm.Designer.cs.

4.6.4.6 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.repositoryLocationTextBox `[private]`

Definition at line 160 of file QuickSubmitForm.Designer.cs.

4.6.4.7 System.Windows.Forms.Button uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.submitButton `[private]`

Definition at line 162 of file QuickSubmitForm.Designer.cs.

4.6.4.8 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.usernameLabel `[private]`

Definition at line 159 of file QuickSubmitForm.Designer.cs.

4.6.4.9 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.QuickSubmitForm.usernameTextBox `[private]`

Definition at line 158 of file QuickSubmitForm.Designer.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/QuickSubmitForm.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/QuickSubmitForm.Designer.cs

4.7 uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserContr Class Reference

Custom Windows Forms user control for repository groups in Microsoft Word 2010 DepositMO add-in.

Public Member Functions

- RepositoryGroupUserControl ()
  Default constructor.

Protected Member Functions

- override void Dispose (bool disposing)
  Clean up any resources being used.

Properties

- string endpoint `[get, set]`
  Endpoint.
- string username `[get, set]`
  Username.
- string password `[get, set]`
  Password.
- int id `[get, set]`
  Control ID.

Private Member Functions

- void InitializeComponent ()

  Required method for Designer support - do not modify the contents of this method with the code editor.

Private Attributes

- System.ComponentModel.IContainer components = null

  Required designer variable.

- System.Windows.Forms.TextBox passwordTextBox
- System.Windows.Forms.Label passwordLabel
- System.Windows.Forms.TextBox usernameTextBox
- System.Windows.Forms.Label usernameLabel
- System.Windows.Forms.TextBox repositoryLocationTextBox
- System.Windows.Forms.Label repositoryLocationLabel
- System.Windows.Forms.Label identLabel

4.7.1 Detailed Description

Custom Windows Forms user control for repository groups in Microsoft Word 2010 DepositMO add-in.

Definition at line 31 of file RepositoryGroupUserControl.cs.

4.7.2 Constructor & Destructor Documentation

4.7.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.RepositoryGroupUserContr ( )

Default constructor.

Sets endpoint, username and password

Definition at line 56 of file RepositoryGroupUserControl.cs.

```
{
    InitializeComponent();
    this.endpoint = Globals.Word2010DepositMOAddIn.DefaultEndpoint;
    this.username = Globals.Word2010DepositMOAddIn.DefaultUsername;
    this.password = Globals.Word2010DepositMOAddIn.DefaultPassword;
}
```

4.7.3 Member Function Documentation

4.7.3.1 override void uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.Dispose (bool *disposing*) `[protected]`

Clean up any resources being used.

Parameters

disposing true if managed resources should be disposed; otherwise, false.

Definition at line 30 of file RepositoryGroupUserControl.Designer.cs.

```
{
    if (disposing && (components != null))
    {
        components.Dispose();
    }
    base.Dispose(disposing);
}
```

4.7.3.2 void uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.InitializeComponent ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 45 of file RepositoryGroupUserControl.Designer.cs.

```
      {
          this.passwordTextBox = new System.Windows.Forms.TextBox();
          this.passwordLabel = new System.Windows.Forms.Label();
          this.usernameTextBox = new System.Windows.Forms.TextBox();
          this.usernameLabel = new System.Windows.Forms.Label();
          this.repositoryLocationTextBox = new System.Windows.Forms.TextBox();
          this.repositoryLocationLabel = new System.Windows.Forms.Label();
          this.identLabel = new System.Windows.Forms.Label();
          this.SuspendLayout();
          //
          // passwordTextBox
          //
          this.passwordTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)))
;
          this.passwordTextBox.Location = new System.Drawing.Point(66, 75);
          this.passwordTextBox.Name = "passwordTextBox";
          this.passwordTextBox.PasswordChar = '*';
          this.passwordTextBox.Size = new System.Drawing.Size(92, 21);
          this.passwordTextBox.TabIndex = 12;
          //
          // passwordLabel
          //
          this.passwordLabel.AutoSize = true;
          this.passwordLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Sy
stem.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
          this.passwordLabel.Location = new System.Drawing.Point(6, 78);
          this.passwordLabel.Name = "passwordLabel";
          this.passwordLabel.Size = new System.Drawing.Size(53, 13);
          this.passwordLabel.TabIndex = 11;
          this.passwordLabel.Text = "Password";
          //
          // usernameTextBox
          //
          this.usernameTextBox.Font = new System.Drawing.Font("Tahoma", 8.25F,
System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)))
;
          this.usernameTextBox.Location = new System.Drawing.Point(66, 49);
          this.usernameTextBox.Name = "usernameTextBox";
```

```
            this.usernameTextBox.Size = new System.Drawing.Size(92, 21);
            this.usernameTextBox.TabIndex = 10;
            //
            // usernameLabel
            //
            this.usernameLabel.AutoSize = true;
            this.usernameLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Sy
stem.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.usernameLabel.Location = new System.Drawing.Point(5, 52);
            this.usernameLabel.Name = "usernameLabel";
            this.usernameLabel.Size = new System.Drawing.Size(55, 13);
            this.usernameLabel.TabIndex = 9;
            this.usernameLabel.Text = "Username";
            //
            // repositoryLocationTextBox
            //
            this.repositoryLocationTextBox.Font = new System.Drawing.Font("Tahoma
", 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((
byte)(0)));
            this.repositoryLocationTextBox.Location = new System.Drawing.Point(4,
 23);
            this.repositoryLocationTextBox.Name = "repositoryLocationTextBox";
            this.repositoryLocationTextBox.Size = new System.Drawing.Size(179, 21
);
            this.repositoryLocationTextBox.TabIndex = 8;
            //
            // repositoryLocationLabel
            //
            this.repositoryLocationLabel.AutoSize = true;
            this.repositoryLocationLabel.Font = new System.Drawing.Font("Tahoma",
 8.25F, System.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((by
te)(0)));
            this.repositoryLocationLabel.Location = new System.Drawing.Point(4, 6
);
            this.repositoryLocationLabel.Name = "repositoryLocationLabel";
            this.repositoryLocationLabel.Size = new System.Drawing.Size(99, 13);
            this.repositoryLocationLabel.TabIndex = 7;
            this.repositoryLocationLabel.Text = "Repository location";
            //
            // identLabel
            //
            this.identLabel.AutoSize = true;
            this.identLabel.Font = new System.Drawing.Font("Tahoma", 8.25F, Syste
m.Drawing.FontStyle.Regular, System.Drawing.GraphicsUnit.Point, ((byte)(0)));
            this.identLabel.Location = new System.Drawing.Point(98, 6);
            this.identLabel.Name = "identLabel";
            this.identLabel.Size = new System.Drawing.Size(13, 13);
            this.identLabel.TabIndex = 13;
            this.identLabel.Text = "0";
            this.identLabel.Visible = false;
            //
            // RepositoryGroupUserControl
            //
            this.AutoScaleDimensions = new System.Drawing.SizeF(6F, 13F);
            this.AutoScaleMode = System.Windows.Forms.AutoScaleMode.Font;
            this.Controls.Add(this.identLabel);
            this.Controls.Add(this.passwordTextBox);
            this.Controls.Add(this.passwordLabel);
            this.Controls.Add(this.usernameTextBox);
            this.Controls.Add(this.usernameLabel);
            this.Controls.Add(this.repositoryLocationTextBox);
            this.Controls.Add(this.repositoryLocationLabel);
            this.Name = "RepositoryGroupUserControl";
            this.Size = new System.Drawing.Size(186, 196);
            this.ResumeLayout(false);
            this.PerformLayout();
```

```
        }
```

4.7.4 Member Data Documentation

4.7.4.1 System.ComponentModel.IContainer uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.components = null `[private]`

Required designer variable.

Definition at line 24 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.2 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.identLabel `[private]`

Definition at line 148 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.3 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.passwordLabel `[private]`

Definition at line 143 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.4 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.passwordTextBox `[private]`

Definition at line 142 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.5 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.repositoryLocationLabel `[private]`

Definition at line 147 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.6 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.repositoryLocationTextBox `[private]`

Definition at line 146 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.7 System.Windows.Forms.Label uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.usernameLabel `[private]`

Definition at line 145 of file RepositoryGroupUserControl.Designer.cs.

4.7.4.8 System.Windows.Forms.TextBox uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.usernameTextBox `[private]`

Definition at line 144 of file RepositoryGroupUserControl.Designer.cs.

4.7.5 Property Documentation

4.7.5.1 string uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.endpoint `[get, set]`

Endpoint.

Definition at line 36 of file RepositoryGroupUserControl.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addRepository(), and uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToAll().

4.7.5.2 int uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.id `[get, set]`

Control ID.

Definition at line 51 of file RepositoryGroupUserControl.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addRepository().

4.7.5.3 string uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.password `[get, set]`

Password.

Definition at line 46 of file RepositoryGroupUserControl.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addRepository(), and uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToAll().

4.7.5.4 string uk.ac.soton.ses.Word2010DepositMOAddIn.RepositoryGroupUserControl.username `[get, set]`

Username.

Definition at line 41 of file RepositoryGroupUserControl.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.addRepository(), and uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.submitToAll().

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/RepositoryGroupUserControl.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/RepositoryGroupUserControl.Desig

4.8 uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources Class Reference

A strongly-typed resource class, for looking up localized strings, etc.

Properties

- static internal global::System.Resources.ResourceManager ResourceManager `[get]`
  Returns the cached ResourceManager instance used by this class.
- static internal global::System.Globalization.CultureInfo Culture `[get, set]`
  Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.
- static internal System.Drawing.Bitmap DocumentImage `[get]`

Private Member Functions

- internal Resources ()

Static Private Attributes

- static global::System.Resources.ResourceManager resourceMan
- static global::System.Globalization.CultureInfo resourceCulture

4.8.1 Detailed Description

A strongly-typed resource class, for looking up localized strings, etc.

Definition at line 25 of file Resources.Designer.cs.

4.8.2 Constructor & Destructor Documentation

4.8.2.1 internal uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources.Resources ( ) `[private]`

Definition at line 32 of file Resources.Designer.cs.

```
                                                     {
        }
```

4.8.3 Member Data Documentation

4.8.3.1 global::System.Globalization.CultureInfo uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources.resourceCulture `[static, private]`

Definition at line 29 of file Resources.Designer.cs.

4.8.3.2 global::System.Resources.ResourceManager uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources.resourceMan `[static, private]`

Definition at line 27 of file Resources.Designer.cs.

4.8.4 Property Documentation

4.8.4.1 internal global::System.Globalization.CultureInfo uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources.Culture `[static, get, set, private]`

Overrides the current thread's CurrentUICulture property for all resource lookups using this strongly typed resource class.

Definition at line 54 of file Resources.Designer.cs.

4.8.4.2 internal System.Drawing.Bitmap uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources.DocumentImage `[static, get, private]`

Definition at line 63 of file Resources.Designer.cs.

4.8.4.3 internal global::System.Resources.ResourceManager uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Resources.ResourceManager `[static, get, private]`

Returns the cached ResourceManager instance used by this class.

Definition at line 39 of file Resources.Designer.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Properties/Resources.Designer.cs

4.9 uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Settings Class Reference

Properties

- static Settings Default `[get]`

Static Private Attributes

- static Settings defaultInstance = ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronize Settings())))

4.9.1 Detailed Description

Definition at line 16 of file Settings.Designer.cs.

4.9.2 Member Data Documentation

4.9.2.1 Settings uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Settings.defaultInstance = ((Settings)(global::System.Configuration.ApplicationSettingsBase.Synchronized(new Settings()))) `[static, private]`

Definition at line 18 of file Settings.Designer.cs.

4.9.3 Property Documentation

4.9.3.1 Settings uk.ac.soton.ses.Word2010DepositMOAddIn.Properties.Settings.Default `[static, get]`

Definition at line 20 of file Settings.Designer.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Properties/Settings.Designer.cs

4.10 uk.ac.soton.ses.SwordAtomReader Class Reference

SWORD 2 Atom XML reader.

Public Member Functions

- SwordAtomReader (XmlDocument atomXml)

  Creates a new SWORD Atom reader from the supplied XML document.

Properties

- XmlDocument AtomXml `[get, set]`

  Gets or sets the Atom response on this instance.

- string EditMediaHref `[get]`

  Edit media href.

- string EditHref `[get]`

  Edit href.

- string ContentsHref `[get]`

  Contents href.

- string AtomId `[get]`

  Atom ID.

- string AtomTitle `[get]`

  Atom title.

- string AtomGenerator `[get]`

  Atom generator.

- string AtomGeneratorUri `[get]`

  Atom generator URI.

- string AtomGeneratorVersion `[get]`

  Atom generator version.

- string AtomSummary `[get]`

  Atom summary.

- string AtomContentType `[get]`

  Atom content type.

- string AtomContentSrc `[get]`

  Atom content source.

- string SwordTreatment `[get]`

  SWORD treatment.

- string SwordPackaging `[get]`

  SWORD packaging.

- string SwordNoop `[get]`

  SWORD noop.

- DateTime AtomUpdated `[get]`

  Atom updated.

- DateTime AtomPublished `[get]`

  Atom published.

Private Member Functions

- string GetXPathValue (string xPathQuery)
- string GetXPathText (string xPathQuery)
- string[ ] GetXPathTextArray (string xPathQuery)
- XmlNode GetSingleNode (string xPathQuery)
- XmlNodeList GetMultipleNodes (string xPathQuery)

Private Attributes

- XmlDocument atomXml = null

  The Atom XML document.

- XmlNamespaceManager xnm = null

  XML namespace manager for Atom XML document.

4.10.1 Detailed Description

SWORD 2 Atom XML reader.

Definition at line 28 of file SwordAtomReader.cs.

4.10.2 Constructor & Destructor Documentation

4.10.2.1 uk.ac.soton.ses.SwordAtomReader.SwordAtomReader (XmlDocument *atomXml*)

Creates a new SWORD Atom reader from the supplied XML document.

Parameters

atomXml Atom XML

Definition at line 44 of file SwordAtomReader.cs.

```
{
    this.atomXml = atomXml;
    if (this.atomXml != null)
    {
        this.xnm = new XmlNamespaceManager(this.atomXml.NameTable);
        this.xnm.AddNamespace("atom", "http://www.w3.org/2005/Atom");
        this.xnm.AddNamespace("sword", "http://purl.org/net/sword/");
    }
}
```

4.10.3 Member Function Documentation

4.10.3.1 XmlNodeList uk.ac.soton.ses.SwordAtomReader.GetMultipleNodes (string *xPathQuery*) `[private]`

Definition at line 99 of file SwordAtomReader.cs.

```
{
    if (this.atomXml == null)
    {
        return null;
    }
    return this.atomXml.SelectNodes(xPathQuery, this.xnm);
}
```

4.10.3.2 XmlNode uk.ac.soton.ses.SwordAtomReader.GetSingleNode (string *xPathQuery*) `[private]`

Definition at line 90 of file SwordAtomReader.cs.

```
{
    if (this.atomXml == null)
    {
        return null;
    }
    return this.atomXml.SelectSingleNode(xPathQuery, this.xnm);

}
```

4.10.3.3 string uk.ac.soton.ses.SwordAtomReader.GetXPathText (string *xPathQuery*) `[private]`

Definition at line 65 of file SwordAtomReader.cs.

```
{
    XmlNode singleNode = this.GetSingleNode(xPathQuery);
    if (singleNode == null)
    {
        return null;
    }
    return singleNode.InnerText;
}
```

4.10.3.4 string [ ] uk.ac.soton.ses.SwordAtomReader.GetXPathTextArray (string *xPathQuery*) `[private]`

Definition at line 75 of file SwordAtomReader.cs.

```
{
    XmlNodeList multipleNodes = this.GetMultipleNodes(xPathQuery);
    if (multipleNodes == null)
    {
        return null;
    }
    string[] nodetexts = new string[multipleNodes.Count];
    for (int i = 0; i < multipleNodes.Count; i++)
    {
        nodetexts[i] = multipleNodes[i].InnerText;
    }
    return nodetexts;
}
```

4.10.3.5 string uk.ac.soton.ses.SwordAtomReader.GetXPathValue (string *xPathQuery*) `[private]`

Definition at line 55 of file SwordAtomReader.cs.

```
{
    XmlNode singleNode = this.GetSingleNode(xPathQuery);
    if (singleNode == null)
```

```
            {
                return null;
            }
            return singleNode.Value;
        }
```

4.10.4 Member Data Documentation

4.10.4.1 XmlDocument uk.ac.soton.ses.SwordAtomReader.atomXml = null [private]

The Atom XML document.

Definition at line 33 of file SwordAtomReader.cs.

4.10.4.2 XmlNamespaceManager uk.ac.soton.ses.SwordAtomReader.xnm = null [private]

XML namespace manager for Atom XML document.

Definition at line 38 of file SwordAtomReader.cs.

4.10.5 Property Documentation

4.10.5.1 string uk.ac.soton.ses.SwordAtomReader.AtomContentSrc [get]

Atom content source.

Definition at line 174 of file SwordAtomReader.cs.

4.10.5.2 string uk.ac.soton.ses.SwordAtomReader.AtomContentType [get]

Atom content type.

Definition at line 169 of file SwordAtomReader.cs.

4.10.5.3 string uk.ac.soton.ses.SwordAtomReader.AtomGenerator [get]

Atom generator.

Definition at line 149 of file SwordAtomReader.cs.

4.10.5.4 string uk.ac.soton.ses.SwordAtomReader.AtomGeneratorUri [get]

Atom generator URI.

Definition at line 154 of file SwordAtomReader.cs.

4.10.5.5 string uk.ac.soton.ses.SwordAtomReader.AtomGeneratorVersion [get]

Atom generator version.

Definition at line 159 of file SwordAtomReader.cs.

4.10.5.6 string uk.ac.soton.ses.SwordAtomReader.AtomId [get]

Atom ID.

Definition at line 139 of file SwordAtomReader.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestAtomReader(), uk.ac.soton.ses.SwordClientHandlerTests.TestDepositEprintsXml(), and uk.ac.soton.ses.SwordClientHandlerTests.TestDepositZipToEprint().

4.10.5.7 DateTime uk.ac.soton.ses.SwordAtomReader.AtomPublished [get]

Atom published.

Definition at line 199 of file SwordAtomReader.cs.

4.10.5.8 string uk.ac.soton.ses.SwordAtomReader.AtomSummary [get]

Atom summary.

Definition at line 164 of file SwordAtomReader.cs.

4.10.5.9 string uk.ac.soton.ses.SwordAtomReader.AtomTitle [get]

Atom title.

Definition at line 144 of file SwordAtomReader.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestGetInfo().

4.10.5.10 DateTime uk.ac.soton.ses.SwordAtomReader.AtomUpdated [get]

Atom updated.

Definition at line 194 of file SwordAtomReader.cs.

4.10.5.11 XmlDocument uk.ac.soton.ses.SwordAtomReader.AtomXml [get, set]

Gets or sets the Atom response on this instance.

Definition at line 112 of file SwordAtomReader.cs.

4.10.5.12 string uk.ac.soton.ses.SwordAtomReader.ContentsHref [get]

Contents href.

Definition at line 134 of file SwordAtomReader.cs.

Referenced by uk.ac.soton.ses.SwordClientHandler.PostDocxToContainer(), and uk.ac.soton.ses.SwordClientHandlerTests.TestGetInfo().

4.10.5.13 string uk.ac.soton.ses.SwordAtomReader.EditHref `[get]`

Edit href.

Definition at line 129 of file SwordAtomReader.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestAtomReader().

4.10.5.14 string uk.ac.soton.ses.SwordAtomReader.EditMediaHref `[get]`

Edit media href.

Definition at line 124 of file SwordAtomReader.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestAtomReader().

4.10.5.15 string uk.ac.soton.ses.SwordAtomReader.SwordNoop `[get]`

SWORD noop.

Definition at line 189 of file SwordAtomReader.cs.

4.10.5.16 string uk.ac.soton.ses.SwordAtomReader.SwordPackaging `[get]`

SWORD packaging.

Definition at line 184 of file SwordAtomReader.cs.

4.10.5.17 string uk.ac.soton.ses.SwordAtomReader.SwordTreatment `[get]`

SWORD treatment.

Definition at line 179 of file SwordAtomReader.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/SwordAtomReader.cs

4.11 uk.ac.soton.ses.SwordClientHandler Class Reference

SWORD 2 client implementation for .NET.

Public Member Functions

- SwordClientHandler (string username, string password, string endpoint)

  Creates a new SWORD 2 handler instance.

- SwordClientHandler (string username, string password, string endpoint, bool resolveInbox)

  Creates a new SWORD 2 handler instance.

- string PostDocxToContainer (string docxFilename)

Deposits the .docx from the supplied filename to the initialised endpoint with the default content type.

- string PostDocxToContainer (string docxFilename, string contentType)

  Deposits the .docx from the supplied filename to the initialised endpoint with the supplied content type.

- string DepositDocxToContents (string docxFilename, string contentType, string contentUri)

  Deposits the .docx from the supplied filename to the supplied endpoint with the supplied content type.

- string PostDocxToRepository (string docxFilename, string contentType, string endpoint)

  Creates a container on the repository and posts the supplied docx file to that container.

- string PostDocxToRepository (string docxFilename)

  Creates a container on the repository and posts the supplied docx file to that container, using the default docx content type and the initialised endpoint on the instance.

- string PostDocxToContainer (string docxFilename, string contentType, string containerUri)

  Posts the supplied docx file to the supplied container.

- string CreateContainer ()

  Creates an empty container on the initialised repository.

- string CreateContainer (string endpoint)

  Creates an empty container on the repository at the supplied endpoint.

- string GetResponseLocation (HttpWebRequest wr)

  Gets the location of the response from the provided
  `HttpWebRequest`

- XmlDocument GetEprintInfo (string uri)

  Gets information about the ePrint at the provided URI.

- bool PutToDocument (string documentFileName, string endpoint)

  Performs a PUT to the supplied endpoint with the supplied document filename using
  `DOCX_CONTENTTYPE`
  .

- bool PutToDocument (string documentFileName, string endpoint, string contentType)

  Performs a PUT to the supplied endpoint with the supplied document filename using the supplied content type.

- XmlDocument DepositZipToEprint (string zipFileName)

  Deposits the zip file in the supplied filename to the initialised endpoint with a gzip content type (to be formally standardised at the server side).

- XmlDocument DepositZipToEprint (string zipFileName, string contentType)

  Deposits the zip file in the supplied filename to the initalised endpoint with the supplied content type.

- XmlDocument DepositEprintsXml (XmlDocument xmlDocument)

  Deposits the EP2 XML document to the initialised endpoint with X-Packaging type
  `EP2_PACKAGING`

- XmlDocument DepositEprintsXml (string xmlFileName)

  Deposits the EP2 XML document in the supplied filename to the initialised endpoint with X-Packaging type
  `EP2_PACKAGING`

- XmlDocument DepositXml (string xmlFileName, string xPackaging)

  Deposits the XML document in the supplied filename to the initialised endpoint with the supplied X-Packaging type.

- XmlDocument DepositXml (XmlDocument xmlDocument, string xPackaging)

  Deposits the XML document to the initialised endpoint with the supplied X-Packaging type.

- XmlDocument Delete (string swordUrl)

  Deletes the document at the target address
  `swordUrl`
  .

- XmlDocument ListRecords (bool stripEndpoint)

  List all records at the endpoint.

- XmlDocument ListRecords ()

  List all records at the endpoint, by calling endpoint/id/records (
  `LIST_RECORD_PATH`
).

Properties

- string Username `[set]`

  Sets the username associated with the repository.

- string Password `[set]`

  Sets the password associated with the repository.

- string Endpoint `[set]`

  Sets the endpoint associated with the repository.

Private Member Functions

- internal void ResolveInbox ()

  Resolves the inbox based on the currently-supplied endpoint.

- HttpWebRequest GetBasicAuthAtomWebRequest (string targetUri)

  Creates a web request instance to the target URI, adding in basic authorisation from the username and password on the instance, and setting the
  `accept`
  header of the request to be
  `application/atom+xml`

- HttpWebRequest GetBasicAuthAtomWebRequest (Uri targetUri)

Creates a web request instance to the target URI, adding in basic authorisation from the username and password on the instance, and setting the
`accept`
header of the request to be
`application/atom+xml`

- void WriteFileToRequest (string filename, HttpWebRequest wr)

  Writes the file in
  `filename`
  to the web request
  `wr`
  .

- XmlDocument GetXmlResponse (HttpWebRequest wr)

  Gets an XML document as the response from a web request.

Private Attributes

- internal bool _resolveInbox = false

  TODO: Whether the inbox URL should be resolved from the HTML of the supplied endpoint.

- string _username

  The repository username.

- string _password

  The repository password.

- string _endpoint

  The main repository endpoint.

Static Private Attributes

- static int BUFFER_SIZE = 16 ∗ 1024

  Buffer size for uploads in bytes.

- static string EP2_PACKAGING = "http://eprints.org/ep2/data/2.0"

  EP2 packaging header value.

- static string DOCX_CONTENTTYPE = "application/vnd.openxmlformats-officedocument.wordprocessingml.document"

  .docx content type

- static string LIST_RECORD_PATH = "/id/records"

  The path for listing records.

- static string EXTRACT_MEDIA = "true"

  Should media be extracted by default, or ignored? Note that this is going into a header so it should be a string rather than a Boolean.

4.11.1 Detailed Description

SWORD 2 client implementation for .NET. Library functions for communicating with a SWORD 2 compliant server

Definition at line 35 of file SwordClientHandler.cs.

4.11.2 Constructor & Destructor Documentation

4.11.2.1 uk.ac.soton.ses.SwordClientHandler.SwordClientHandler (string *username*, string *password*, string *endpoint*)

Creates a new SWORD 2 handler instance.

Parameters

username The repository username

password The repository password

endpoint The main repository endpoint

Definition at line 124 of file SwordClientHandler.cs.

```
{
    this._username = username;
    this._password = password;
    this._endpoint = endpoint;
}
```

4.11.2.2 uk.ac.soton.ses.SwordClientHandler.SwordClientHandler (string *username*, string *password*, string *endpoint*, bool *resolveInbox*)

Creates a new SWORD 2 handler instance.

Parameters

username The repository username

password The repository password

endpoint The main repository endpoint

resolveInbox Whether to attempt to resolve the 'inbox' from the supplied endpoint

Definition at line 138 of file SwordClientHandler.cs.

```
{
    this._username = username;
    this._password = password;
    this._endpoint = endpoint;
    this._resolveInbox = resolveInbox;
    this.ResolveInbox();
}
```

4.11.3 Member Function Documentation

4.11.3.1 string uk.ac.soton.ses.SwordClientHandler.CreateContainer ( )

Creates an empty container on the initialised repository.

Returns

The address of the new empty container

Definition at line 379 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestCreateContainer().

```
{
    return this.CreateContainer(this._endpoint);
}
```

4.11.3.2 string uk.ac.soton.ses.SwordClientHandler.CreateContainer (string *endpoint*)

Creates an empty container on the repository at the supplied endpoint.

Parameters

endpoint The endpoint at which to create the new empty container

Returns

The address of the new empty container

Definition at line 389 of file SwordClientHandler.cs.

```
{
    HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(endpoint);
    wr.Method = "POST";
    wr.ContentType = "application/atom+xml";
    string emptyAtom = @"<?xml version=""1.0"" encoding=""utf-8"" ?>
     <entry xmlns=""http://www.w3.org/2005/Atom"" />";
    wr.ContentLength = (long)emptyAtom.Length;

    // TODO: not the most elegant way to write this 'empty' Atom request
    // but functional
    using (Stream requestStream = wr.GetRequestStream())
    {
        foreach (byte b in emptyAtom)
        {
            requestStream.WriteByte(b);
        }
    }

    return this.GetResponseLocation(wr);
}
```

4.11.3.3 XmlDocument uk.ac.soton.ses.SwordClientHandler.Delete (string *swordUrl*)

Deletes the document at the target address `swordUrl`

.

Parameters

swordUrl The target address at which the document resides

Returns

Atom response

Definition at line 608 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestDelete().

```
  {
      HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(swordUrl);
      wr.Method = "DELETE";
      return this.GetXmlResponse(wr);
  }
```

4.11.3.4 string uk.ac.soton.ses.SwordClientHandler.DepositDocxToContents (string *docxFilename,* string *contentType,* string *contentUri*)

Deposits the .docx from the supplied filename to the supplied endpoint with the supplied content type.

Parameters

docxFilename The filename of the .docx to deposit

contentType The content type to use for the deposition

contentUri The endpoint against which the deposition should occur

Returns

Atom response

Definition at line 299 of file SwordClientHandler.cs.

```
  {
      FileInfo fi = new FileInfo(docxFilename);
      HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(contentUri);

      wr.Method = "POST";
      wr.Headers.Add("Content-Disposition", "form-data; name=\"" + fi.Name
+ "\"; filename=\"" + fi.Name + "\"");
      wr.ContentType = contentType;
      wr.ContentLength = fi.Length;
      // should the default be to extract media, or ignore it?
      wr.Headers.Add("X-Extract-Media", EXTRACT_MEDIA);
      wr.Headers.Add("X-Override-Metadata", "true");
      this.WriteFileToRequest(docxFilename, wr);

      return this.GetResponseLocation(wr); //this.GetXmlResponse(wr);
  }
```

4.11.3.5 XmlDocument uk.ac.soton.ses.SwordClientHandler.DepositEprintsXml (XmlDocument *xmlDocument*)

Deposits the EP2 XML document to the initialised endpoint with X-Packaging type

```
EP2_PACKAGING
```

Parameters

xmlDocument The EP2 XML document to deposit

Returns

Atom response

Definition at line 540 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestDepositEprintsXml().

```
        {
            return this.DepositXml(xmlDocument, EP2_PACKAGING);
        }
```

4.11.3.6 XmlDocument uk.ac.soton.ses.SwordClientHandler.DepositEprintsXml (string *xmlFileName*)

Deposits the EP2 XML document in the supplied filename to the initialised endpoint with X-Packaging type

```
EP2_PACKAGING
```

Parameters

xmlFileName The filename of the EP2 XML document to deposit

Returns

Atom response

Definition at line 551 of file SwordClientHandler.cs.

```
        {
            return this.DepositXml(xmlFileName, EP2_PACKAGING);
        }
```

4.11.3.7 XmlDocument uk.ac.soton.ses.SwordClientHandler.DepositXml (XmlDocument *xmlDocument,* string *xPackaging*)

Deposits the XML document to the initialised endpoint with the supplied X-Packaging type.

Parameters

xmlDocument The XML document to deposit

xPackaging The X-Packaging type

Returns

Atom response

Definition at line 577 of file SwordClientHandler.cs.

```
    {
        HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(this._endpoint);

        wr.Method = "POST";
        wr.Headers.Add("X-Packaging", xPackaging);
        wr.ContentType = "text/xml";

        // note that we can’t wrap the Stream here with a using as there
        // is a possibility that Dispose will get called more than once
        // and this will generate an ObjectDisposedException. Instead,
        // we’ll not assign the request stream to anything and just
        // pass it through directly

        XmlWriterSettings xws = new XmlWriterSettings();
        xws.CloseOutput = true;

        using (XmlWriter xw = XmlWriter.Create(wr.GetRequestStream(), xws))
        {
            xmlDocument.WriteTo(xw);

            // ensure that the XML writer is flushes as this does *not* happe
  n automatically
            xw.Flush();
        }
        return this.GetXmlResponse(wr);
    }
```

4.11.3.8 XmlDocument uk.ac.soton.ses.SwordClientHandler.DepositXml (string *xmlFileName,* string *xPackaging*)

Deposits the XML document in the supplied filename to the initialised endpoint with the supplied X-Packaging type.

Parameters

xmlFileName The filename of the XML document to deposit

xPackaging The X-Packaging type

Returns

Atom response

Definition at line 563 of file SwordClientHandler.cs.

```
    {
        XmlDocument xmlDocument = new XmlDocument();
        xmlDocument.Load(xmlFileName);
        return this.DepositXml(xmlDocument, xPackaging);
    }
```

4.11.3.9 XmlDocument uk.ac.soton.ses.SwordClientHandler.DepositZipToEprint (string *zipFileName,* string *contentType*)

Deposits the zip file in the supplied filename to the initalised endpoint with the supplied content type.

Parameters

zipFileName The zip filename

contentType The content type

Returns

Atom response

Definition at line 520 of file SwordClientHandler.cs.

```
    {
        FileInfo fi = new FileInfo(zipFileName);
        HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(this._endpoint);
        wr.Method = "POST";
        wr.Headers.Add("Content-Disposition", "form-data; name=\"" + fi.Name
  + "\"; filename=\"" + fi.Name + "\"");
        wr.ContentType = contentType;
        wr.ContentLength = fi.Length;
        wr.Headers.Add("X-Extract-Archive", "true");
        this.WriteFileToRequest(zipFileName, wr);

        return this.GetXmlResponse(wr);
    }
```

4.11.3.10 XmlDocument uk.ac.soton.ses.SwordClientHandler.DepositZipToEprint (string *zipFileName*)

Deposits the zip file in the supplied filename to the initialised endpoint with a gzip content type (to be formally standardised at the server side).

Parameters

zipFileName The zip filename

Returns

Atom response

Definition at line 508 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestDepositZipToEprint().

```
    {
        return this.DepositZipToEprint(zipFileName, "application/g-zip");
    }
```

4.11.3.11 HttpWebRequest uk.ac.soton.ses.SwordClientHandler.GetBasicAuthAtomWebRequest (string *targetUri*) [private]

Creates a web request instance to the target URI, adding in basic authorisation from the username and password on the instance, and setting the

```
accept
```

header of the request to be

```
application/atom+xml
```

Parameters

targetUri The URI to which the request should be made

Returns

Web request

Definition at line 173 of file SwordClientHandler.cs.

```
        {
            return this.GetBasicAuthAtomWebRequest(new Uri(targetUri));
        }
```

4.11.3.12 HttpWebRequest uk.ac.soton.ses.SwordClientHandler.GetBasicAuthAtomWebRequest (Uri *targetUri*) [private]

Creates a web request instance to the target URI, adding in basic authorisation from the username and password on the instance, and setting the

```
accept
```

header of the request to be

```
application/atom+xml
```

Parameters

targetUri The URI to which the request should be made

Returns

Web request

Definition at line 185 of file SwordClientHandler.cs.

```
        {
            HttpWebRequest request = (HttpWebRequest)WebRequest.Create(targetUri)
      ;
            CredentialCache cc = new CredentialCache();
            cc.Add(targetUri, "Basic", new NetworkCredential(this._username, this
      ._password, "Authenticate"));
            //cc.Add(new Uri(targetUri), "Basic", new NetworkCredential(this._use
      rname, this._password));
            request.Credentials = cc;
            request.Headers.Add(HttpRequestHeader.Authorization, "Basic " + Conve
      rt.ToBase64String(new ASCIIEncoding().GetBytes(this._username + ':' + this.
      _password)));
            request.Accept = "application/atom+xml";
            return request;
        }
```

4.11.3.13 XmlDocument uk.ac.soton.ses.SwordClientHandler.GetEprintInfo (string *uri*)

Gets information about the ePrint at the provided URI.

Parameters

uri ePrint URI

Returns

Atom XML document

Definition at line 431 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestGetInfo().

```
  {
      HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(uri);
      wr.Method = "GET";
      return this.GetXmlResponse(wr);
  }
```

4.11.3.14 string uk.ac.soton.ses.SwordClientHandler.GetResponseLocation (HttpWebRequest *wr*)

Gets the location of the response from the provided

`HttpWebRequest`

Parameters

wr The web request to query

Returns

Location string

Definition at line 416 of file SwordClientHandler.cs.

```
  {
      string location = null;
      using (HttpWebResponse resp = (HttpWebResponse)wr.GetResponse())
      {
          location = resp.Headers[HttpResponseHeader.Location].ToString();
      }
      return location;
  }
```

4.11.3.15 XmlDocument uk.ac.soton.ses.SwordClientHandler.GetXmlResponse (HttpWebRequest *wr*) [private]

Gets an XML document as the response from a web request.

The response will be

```
null
```

if it is invalid, either because it is not well-formed XML (e.g. HTML 4.0) or there was no acceptable response available from the server (e.g. 404)

Parameters

wr The web request from which the response should be requested

Returns

The XML document response

Definition at line 232 of file SwordClientHandler.cs.

```
    {
        XmlDocument xmlResponse = null;

        try
        {
            using (WebResponse res = wr.GetResponse())
            {
                using (Stream resStream = res.GetResponseStream())
                {
                    // when this method was implemented, the SWORD 2 server w
as still returning HTML:
                    // if (((HttpWebResponse)res).StatusCode == HttpStatusCod
e.OK)
                    // {
                    //    return null;
                    // }
                    // - so try/catch this in case it changes

                    try
                    {
                        xmlResponse = new XmlDocument();
                        xmlResponse.Load(resStream);
                    }
                    catch (XmlException xex)
                    {
                        Debug.WriteLine(xex);
                        // xmlResponse = null; // depends how we want to hand
le this
                    }
                }
            }
        }
        catch (WebException wex)
        {
            // likely a 404, write the exception if we're in debug mode, and
fall through
            // to return the null response
            Debug.WriteLine(wex);
        }
        return xmlResponse;
    }
```

4.11.3.16 XmlDocument uk.ac.soton.ses.SwordClientHandler.ListRecords ( )

List all records at the endpoint, by calling endpoint/id/records (

```
LIST_RECORD_PATH
```

).

Returns

List of records in XML format, else

```
null
```

Exceptions

WebException Typically thrown at a timeout on the endpoint

Definition at line 690 of file SwordClientHandler.cs.

```
        {
            return this.ListRecords(false);
        }
```

4.11.3.17 XmlDocument uk.ac.soton.ses.SwordClientHandler.ListRecords (bool *stripEndpoint*)

List all records at the endpoint.

Parameters

stripEndpoint `true`

if the endpoint should be stripped down and the default of /id/records (

```
LIST_RECORD_PATH
```

) used, else

```
false
```

if the literal endpoint, suffixed with /id/records (

```
LIST_RECORD_PATH
```

) value should be used

Returns

List of records in XML format, else

```
null
```

Exceptions

WebException Typically thrown at a timeout on the endpoint

Definition at line 624 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DumpListingToLog(), and uk.ac.soton.ses.SwordClientHandlerTests.TestList().

```
    {
        string target = this._endpoint;

        HttpWebRequest initialWr = null;
        HttpWebRequest realWr = null;
        WebResponse wresp = null;
        XmlDocument xmlDocument = null;

        if (stripEndpoint)
        {
            target = new Uri(this._endpoint).GetLeftPart(UriPartial.Authority
);
        }
        string records = target + LIST_RECORD_PATH;

        Debug.WriteLine("Requesting records from " + records);
        initialWr = this.GetBasicAuthAtomWebRequest(records);
        initialWr.Method = "GET";

        try
        {
            wresp = initialWr.GetResponse();
        }
        catch (WebException wex)
        {
            // if we've timed out, rethrow the exception
            if (wex.Status == WebExceptionStatus.Timeout)
            {
                throw wex;
            }

            // being forced to handle exceptions as part of normal program co
ntrol
            // is not pleasant (303 in this case)
            Debug.WriteLine(wex.Response.ResponseUri);
            Uri realUri = wex.Response.ResponseUri;
            realWr = this.GetBasicAuthAtomWebRequest(realUri);
            realWr.Method = "GET";
            wresp = realWr.GetResponse();
        }

        using (Stream resStream = wresp.GetResponseStream())
        {
            try
            {
                xmlDocument = new XmlDocument();
                xmlDocument.Load(resStream);
            }
            catch (XmlException xex)
            {
                Debug.WriteLine(xex);
            }
        }

        if (wresp != null)
        {
            wresp.Close();
        }

        return xmlDocument;
    }
```

4.11.3.18 string uk.ac.soton.ses.SwordClientHandler.PostDocxToContainer (string *docxFilename,* string *contentType,* string *containerUri*)

Posts the supplied docx file to the supplied container.

Parameters

docxFilename The filename of the docx to post

contentType The content type of the docx

containerUri The URI of the container

Returns

Definition at line 360 of file SwordClientHandler.cs.

References uk.ac.soton.ses.SwordAtomReader.ContentsHref.

```
        {
            XmlDocument response = this.GetEprintInfo(containerUri);
            if (response == null) return null;

            SwordAtomReader sar = new SwordAtomReader(response);
            if (sar == null) return null;

            string contentsHref = sar.ContentsHref;
            Debug.WriteLine("Contents href is {0}", contentsHref);
            if (String.IsNullOrEmpty(contentsHref)) return null;

            return this.DepositDocxToContents(docxFilename, contentType, contents
      Href);
        }
```

4.11.3.19 string uk.ac.soton.ses.SwordClientHandler.PostDocxToContainer (string *docxFilename*)

Deposits the .docx from the supplied filename to the initialised endpoint with the default content type.

Parameters

docxFilename The filename of the .docx to deposit

Returns

Atom response

Definition at line 276 of file SwordClientHandler.cs.

```
        {
            return this.PostDocxToContainer(docxFilename, DOCX_CONTENTTYPE);
        }
```

4.11.3.20 string uk.ac.soton.ses.SwordClientHandler.PostDocxToContainer (string *docxFilename*, string *contentType*)

Deposits the .docx from the supplied filename to the initialised endpoint with the supplied content type.

Parameters

docxFilename The filename of the .docx to deposit

contentType The content type to use for the deposition

Returns

Atom response

Definition at line 287 of file SwordClientHandler.cs.

```
        {
            return this.PostDocxToContainer(docxFilename, contentType, this.
      _endpoint);
        }
```

4.11.3.21 string uk.ac.soton.ses.SwordClientHandler.PostDocxToRepository (string *docxFilename*, string *contentType*, string *endpoint*)

Creates a container on the repository and posts the supplied docx file to that container.

Parameters

docxFilename The filename of the docx to post

contentType The content type of the docx

endpoint The endpoint against which the posting should occur

Returns

Definition at line 324 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl._-SubmitDocument(), and uk.ac.soton.ses.SwordClientHandlerTests.TestPutToDocument().

```
        {
            string container = this.CreateContainer(endpoint);
            if (String.IsNullOrEmpty(container))
            {
                return null;
            }

            return this.PostDocxToContainer(docxFilename, contentType);
        }
```

4.11.3.22 string uk.ac.soton.ses.SwordClientHandler.PostDocxToRepository (string *docxFilename*)

Creates a container on the repository and posts the supplied docx file to that container, using the default docx content type and the initialised endpoint on the instance.

Parameters

docxFilename The filename of the docx to post

Returns

Definition at line 342 of file SwordClientHandler.cs.

```
    {
        string container = this.CreateContainer(this._endpoint);
        if (String.IsNullOrEmpty(container))
        {
            return null;
        }

        return this.PostDocxToContainer(docxFilename, DOCX_CONTENTTYPE, conta
iner);
    }
```

4.11.3.23 bool uk.ac.soton.ses.SwordClientHandler.PutToDocument (string *documentFileName*, string *endpoint*)

Performs a PUT to the supplied endpoint with the supplied document filename using

```
DOCX_CONTENTTYPE
```

.

This will update an existing entry on the SWORD target

Parameters

documentFileName The filename of the document to use for the update

endpoint The endpoint against which the update should occur

Returns

Atom response

Definition at line 445 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl._SubmitDocument(), uk.ac.soton.ses.SwordClientHandlerTests.TestPutOnly(), and uk.ac.soton.ses.SwordClientHandlerTests.TestPutToDocument().

```
    {
        return this.PutToDocument(documentFileName, endpoint,
DOCX_CONTENTTYPE);
    }
```

4.11.3.24 bool uk.ac.soton.ses.SwordClientHandler.PutToDocument (string *documentFileName,* string *endpoint,* string *contentType*)

Performs a PUT to the supplied endpoint with the supplied document filename using the supplied content type.

This will update an existing entry on the SWORD target

Parameters

documentFileName The filename of the document to use for the update

endpoint The endpoint against which the update should occur

contentType The content type to use for the update

Returns

Atom response

Definition at line 458 of file SwordClientHandler.cs.

```
        {
            HttpWebRequest wr = this.GetBasicAuthAtomWebRequest(endpoint);
            FileInfo fi = new FileInfo(documentFileName);
            wr.Method = "PUT";
            wr.Headers.Add("Content-Disposition", "form-data; name=\"" + fi.Name
      + "\"; filename=\"" + fi.Name + "\"");
            wr.ContentType = contentType;
            wr.ContentLength = fi.Length;
            wr.Headers.Add("X-Extract-Media", "true");
            wr.Headers.Add("X-Override-Metadata", "true");
            this.WriteFileToRequest(documentFileName, wr);
            HttpWebResponse response = null;
            try

            {
                response = (HttpWebResponse)wr.GetResponse();
            }
            catch (WebException ex)
            {
                Debug.WriteLine(ex);
                if (ex.Response != null && ex.Response.Headers != null)
                {
                    foreach (HttpResponseHeader h in ex.Response.Headers)
                    {
                        Debug.WriteLine("Header: " + h.ToString());
                    }
                }
            }

            if (response == null) { return false; }

            bool responseOk = false;
            if (response.StatusCode == HttpStatusCode.OK)
            {
                responseOk = true;
            }

            if (response != null)
            {
                response.Close();
            }
            return responseOk;
        }
```

4.11.3.25 internal void uk.ac.soton.ses.SwordClientHandler.ResolveInbox ( ) [private]

Resolves the inbox based on the currently-supplied endpoint.

Definition at line 150 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestResolveInbox().

```
        {
            // if we're not supposed to resolve the inbox, skip this step
            if (!this._resolveInbox)
            {
                return;
            }

            string inbox = HtmlScraper.GetAttributeContent(this._endpoint, "link"
      , "rel", "SwordDeposit", "href");

            if (!String.IsNullOrEmpty(inbox))
            {
                this._endpoint = inbox;
            }
        }
```

4.11.3.26 void uk.ac.soton.ses.SwordClientHandler.WriteFileToRequest (string *filename*, HttpWebRequest *wr*) [private]

Writes the file in

```
filename
```

to the web request

```
wr
```

.
This uses a buffer whose size is stored statically in

```
BUFFER_SIZE
```

to speed up the transfer

Parameters

filename The absolute filename to send

wr The web request on which to send the file

Definition at line 204 of file SwordClientHandler.cs.

```
        {
            wr.Timeout = 15000;

            using (Stream requestStream = wr.GetRequestStream())
            {
                using (FileStream fs = File.OpenRead(filename))
                {
                    int readCount;
```

```
              byte[] buffer = new byte[BUFFER_SIZE];
              while ((readCount = fs.Read(buffer, 0, buffer.Length)) != 0)
              {
                  Debug.WriteLine(String.Format("Doing a block, at {0}", fs
.Position));
                  requestStream.Write(buffer, 0, readCount);

              }
              Debug.WriteLine("Finished with input file");
          }
          Debug.WriteLine("Finished with output stream");
      }
  }
```

4.11.4 Member Data Documentation

4.11.4.1 string uk.ac.soton.ses.SwordClientHandler._endpoint [private]

The main repository endpoint.

Definition at line 83 of file SwordClientHandler.cs.

4.11.4.2 string uk.ac.soton.ses.SwordClientHandler._password [private]

The repository password.

Definition at line 78 of file SwordClientHandler.cs.

4.11.4.3 internal bool uk.ac.soton.ses.SwordClientHandler._resolveInbox = false [private]

TODO: Whether the inbox URL should be resolved from the HTML of the supplied endpoint.

TODO

Definition at line 68 of file SwordClientHandler.cs.

Referenced by uk.ac.soton.ses.SwordClientHandlerTests.TestResolveInbox().

4.11.4.4 string uk.ac.soton.ses.SwordClientHandler._username [private]

The repository username.

Definition at line 73 of file SwordClientHandler.cs.

4.11.4.5 int uk.ac.soton.ses.SwordClientHandler.BUFFER_SIZE = 16 ∗ 1024 [static, private]

Buffer size for uploads in bytes.

Definition at line 40 of file SwordClientHandler.cs.

4.11.4.6 string uk.ac.soton.ses.SwordClientHandler.DOCX_CONTENTTYPE = "application/vnd.openxmlformats-officedocument.wordprocessingml.document" `[static, private]`

.docx content type

Definition at line 50 of file SwordClientHandler.cs.

4.11.4.7 string uk.ac.soton.ses.SwordClientHandler.EP2_PACKAGING = "http://eprints.org/ep2/data/2.0" `[static, private]`

EP2 packaging header value.

Definition at line 45 of file SwordClientHandler.cs.

4.11.4.8 string uk.ac.soton.ses.SwordClientHandler.EXTRACT_MEDIA = "true" `[static, private]`

Should media be extracted by default, or ignored? Note that this is going into a header so it should be a string rather than a Boolean.

Definition at line 62 of file SwordClientHandler.cs.

4.11.4.9 string uk.ac.soton.ses.SwordClientHandler.LIST_RECORD_PATH = "/id/records" `[static, private]`

The path for listing records.

Definition at line 55 of file SwordClientHandler.cs.

4.11.5 Property Documentation

4.11.5.1 string uk.ac.soton.ses.SwordClientHandler.Endpoint `[set]`

Sets the endpoint associated with the repository.

Definition at line 111 of file SwordClientHandler.cs.

4.11.5.2 string uk.ac.soton.ses.SwordClientHandler.Password `[set]`

Sets the password associated with the repository.

Definition at line 100 of file SwordClientHandler.cs.

4.11.5.3 string uk.ac.soton.ses.SwordClientHandler.Username `[set]`

Sets the username associated with the repository.

Definition at line 89 of file SwordClientHandler.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/SwordClientHandler.cs

4.12 uk.ac.soton.ses.SwordClientHandlerTests Class Reference

A series of basic tests for the .NET API.

Public Member Functions

- SwordClientHandlerTests (string sword2User, string sword2Password, string sword2Endpoint)
- void Run ()

Private Member Functions

- void TestDelete (SwordClientHandler sword2Handler, string targetToDelete)
- void TestPutOnly (SwordClientHandler sword2Handler)
- void TestPutToDocument (SwordClientHandler sword2Handler)
- void TestResolveInbox (SwordClientHandler sword2Handler)
- void TestDepositZipToEprint (SwordClientHandler sword2Handler)
- void TestDepositEprintsXml (SwordClientHandler sword2Handler)
- void TestGetInfo (SwordClientHandler sword2Handler)
- void TestList (SwordClientHandler sword2Handler)
- void TestAtomReader ()
- void TestCreateContainer (SwordClientHandler sword2Handler)

Private Attributes

- string sword2User = null
- string sword2Password = null
- string sword2Endpoint = null
- internal string EPRINT_TO_DELETE = "http://depositmo.eprints.org/id/eprint/58"
- XmlWriterSettings xws = new XmlWriterSettings()
- XmlWriter consoleWriter = null
- XmlDocument response = null

4.12.1 Detailed Description

A series of basic tests for the .NET API.

Definition at line 30 of file SwordClientHandlerTests.cs.

4.12.2 Constructor & Destructor Documentation

4.12.2.1 uk.ac.soton.ses.SwordClientHandlerTests.SwordClientHandlerTests (string *sword2User*, string *sword2Password*, string *sword2Endpoint*)

Definition at line 42 of file SwordClientHandlerTests.cs.

```
        {
            this.sword2User = sword2User;
            this.sword2Password = sword2Password;
            this.sword2Endpoint = sword2Endpoint;
            this.xws.Indent = true;
            this.consoleWriter = XmlWriter.Create(Console.Out, this.xws);
        }
```

4.12.3 Member Function Documentation

4.12.3.1 void uk.ac.soton.ses.SwordClientHandlerTests.Run ( )

Definition at line 51 of file SwordClientHandlerTests.cs.

```
        {
            bool runAllTests = false;
            bool testPutToDocument = true;
            bool testDepositZipToEprint = false;
            bool testDepositEprintsXml = false;
            bool testAtomReader = false;
            bool testDelete = false;
            bool testList = false;
            bool testResolveInbox = false;
            bool testGetInfo = false;
            bool testCreateContainer = false;
            bool testPutOnly = false;

            SwordClientHandler sword2Handler = new SwordClientHandler(this.
      sword2User, this.sword2Password, this.sword2Endpoint);

            if (testResolveInbox || runAllTests)
            {
                this.TestResolveInbox(sword2Handler);
            }

            if (testAtomReader || runAllTests)
            {
                this.TestAtomReader();
            }

            if (testDelete || runAllTests)
            {
                this.TestDelete(sword2Handler, EPRINT_TO_DELETE);
            }

            if (testDepositZipToEprint || runAllTests)
            {
                this.TestDepositZipToEprint(sword2Handler);
            }

            if (testPutToDocument || runAllTests)
            {
                TestPutToDocument(sword2Handler);
            }

            if (testDepositEprintsXml || runAllTests)
            {
                this.TestDepositEprintsXml(sword2Handler);
            }

            if (testList || runAllTests)
            {
                this.TestList(sword2Handler);
            }
            if (testGetInfo || runAllTests)
            {
                this.TestGetInfo(sword2Handler);
            }

            if (testCreateContainer || runAllTests)
            {
                this.TestCreateContainer(sword2Handler);
            }

            if (testPutOnly)
            {
                this.TestPutOnly(sword2Handler);
            }

            if (this.response != null)
            {
                response.WriteTo(consoleWriter);
                consoleWriter.Flush();
            }
        }
```

4.12.3.2 void uk.ac.soton.ses.SwordClientHandlerTests.TestAtomReader ( ) [private]

Definition at line 244 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordAtomReader.AtomId, uk.ac.soton.ses.SwordAtomReader.EditHref, and uk.ac.soton.ses.SwordAtomReader.EditMediaHref.

```
        {
            XmlDocument atomResponse = new XmlDocument();
            //atomResponse.LoadXml(@"<?xml version=""1.0"" encoding=""UTF-8""?><a
      tom:entry xmlns:atom=""http://www.w3.org/2005/Atom"" xmlns:sword=""http://purl.or
      g/net/sword/"">  <atom:title>On Testing the Atom Protocol...</atom:title>  <atom:
      id>http://depositmo.eprints.org/id/eprint/40</atom:id>  <atom:updated>2010-11-04T
      05:35:21Z</atom:updated>  <atom:published>2006-10-25T00:45:02Z</atom:published>
      <atom:author>    <atom:name>admin</atom:name>    <atom:email>davetaz@ecs.soton.ac
      .uk</atom:email>  </atom:author>  <atom:summary type=""text"" />  <atom:content t
      ype=""text/xml"" src=""http://depositmo.eprints.org/id/document/206"" />  <atom:l
      ink rel=""edit-media"" href=""http://depositmo.eprints.org/id/document/206"" />
      <atom:link rel=""edit"" href=""http://depositmo.eprints.org/sword-app/atom/40.ato
      m"" />  <atom:generator uri=""http://depositmo.eprints.org"" version=""1.3"">Depo
      sitMO / SWORD 2 Endpoint [eprints-build-2010-11-26-r5991]</atom:generator>  <swor
      d:treatment>Deposited items will remain in your user inbox until you manually sen
      d them for reviewing.</sword:treatment>  <sword:packaging>http://eprints.org/ep2/
      data/2.0</sword:packaging>  <sword:noOp>false</sword:noOp></atom:entry>");
            atomResponse.LoadXml(@"<entry>
  <title>A magical and modified document</title>
  <link rel=""self"" href=""http://depositmo.eprints.org/cgi/export/eprint/311/At
      om/depositmo-eprint-311.xml""/>
  <link rel=""edit"" href=""http://depositmo.eprints.org/id/eprint/311""/>
  <link rel=""edit-media"" href=""http://depositmo.eprints.org/id/eprint/311/cont
      ents""/>
  <link rel=""alternate"" href=""http://depositmo.eprints.org/id/eprint/311""/>
  <summary>Q., Hercules (2011) A magical and modified document.</summary>
  <updated>2011-03-03T08:01:37Z</updated>
  <id>http://depositmo.eprints.org/id/eprint/311</id>
  <category term=""article"" scheme=""http://depositmo.eprints.org/data/eprint#ty
      pe""/>
  <author>
    <name>Hercules Q.</name>
  </author>
</entry>
");

            XmlWriterSettings xws = new XmlWriterSettings();
            xws.Indent = true;
            XmlWriter xw = XmlWriter.Create(Console.Out);
            atomResponse.WriteTo(xw);
            xw.Flush();

            SwordAtomReader sar = new SwordAtomReader(atomResponse);
            Console.WriteLine("EditMediaHref: " + sar.EditMediaHref);
            Console.WriteLine("EditHref: " + sar.EditHref);
            Console.WriteLine("AtomId: " + sar.AtomId);
        }
```

4.12.3.3 void uk.ac.soton.ses.SwordClientHandlerTests.TestCreateContainer (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 276 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordClientHandler.CreateContainer().

```
        {
            string response = sword2Handler.CreateContainer();
            return;
        }
```

4.12.3.4 void uk.ac.soton.ses.SwordClientHandlerTests.TestDelete (SwordClientHandler *sword2Handler,* string *targetToDelete*) `[private]`

Definition at line 124 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordClientHandler.Delete().

```
        {
            Debug.WriteLine(String.Format("Deleting document at {0}", targetToDel
      ete));
            XmlDocument atomResponse = sword2Handler.Delete(targetToDelete);
            Debug.WriteLine("Finished deletion test");
        }
```

4.12.3.5 void uk.ac.soton.ses.SwordClientHandlerTests.TestDepositEprintsXml (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 190 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordAtomReader.AtomId, and uk.ac.soton.ses.SwordClientHandler.DepositEprintsXm

```
        {
            Debug.WriteLine("Depositing XML file");
            XmlDocument xmlDocument = new XmlDocument();
            xmlDocument.Load(@"P:\tmp\depositmo_sword2_test_scripts\test-import.x
      ml");
            this.response = sword2Handler.DepositEprintsXml(xmlDocument);
            SwordAtomReader sar = new SwordAtomReader(this.response);
            Debug.WriteLine("ID was " + sar.AtomId);
        }
```

4.12.3.6 void uk.ac.soton.ses.SwordClientHandlerTests.TestDepositZipToEprint (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 182 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordAtomReader.AtomId, and uk.ac.soton.ses.SwordClientHandler.DepositZipToEpri

```
        {
            Debug.WriteLine("Depositing zip file");
            this.response = sword2Handler.DepositZipToEprint(@"P:\tmp\depositmo_s
      word2_test_scripts\pics.tar.gz");
            SwordAtomReader sar = new SwordAtomReader(this.response);
            Debug.WriteLine("ID was " + sar.AtomId);
        }
```

4.12.3.7 void uk.ac.soton.ses.SwordClientHandlerTests.TestGetInfo (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 200 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordAtomReader.AtomTitle, uk.ac.soton.ses.SwordAtomReader.ContentsHref, and uk.ac.soton.ses.SwordClientHandler.GetEprintInfo().

```
        {
            this.response = sword2Handler.GetEprintInfo("http://bazaar.eprints.or
      g/id/eprint/23");
            SwordAtomReader sar = new SwordAtomReader(this.response);
            Console.WriteLine("Atom title is {0}", sar.AtomTitle);
            Console.WriteLine("Contents href is {0}", sar.ContentsHref);

            Debug.WriteLine("Got info!");
        }
```

4.12.3.8 void uk.ac.soton.ses.SwordClientHandlerTests.TestList (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 210 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordClientHandler.ListRecords(), and uk.ac.soton.ses.SwordListEntry.Title.

```
        {
            Debug.WriteLine("Retrieving listing");
            //XmlDocument xmlDocument = new XmlDocument();
            //this.response = sword2Handler.ListRecords();
            //Uri uri = new Uri("http://yomiko.ecs.soton.ac.uk:8027/more/parts/to
      /this/");
            //string leftPart = uri.GetLeftPart(UriPartial.Authority);

            //XmlReader xr = XmlTextReader.Create(@"C:\listresults.xml");
            //this.response = new XmlDocument();
            //this.response.Load(@"C:\listresults.xml");
            //SwordListReader slr2 = new SwordListReader(this.response);

            //return;

            this.response = sword2Handler.ListRecords(true);
            //this.response.WriteTo(this.consoleWriter);
            SwordListReader slr = new SwordListReader(this.response);

            //XmlWriter xw=XmlTextWriter.Create(@"C:\listresults.xml");
            //this.response.WriteTo(xw);
            //xw.Close();

            foreach (SwordListEntry sle in slr.Entries)
            {
                Console.WriteLine("Title: " + sle.Title);
            }

            this.consoleWriter.Flush();
            Console.Out.Flush();
            Console.ReadKey();
        }
```

4.12.3.9 void uk.ac.soton.ses.SwordClientHandlerTests.TestPutOnly (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 131 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordClientHandler.PutToDocument().

```
        {
            bool result = sword2Handler.PutToDocument(@"P:\tmp\hyb.docx", "http:/
      /depositmo.eprints.org/id/document/1736664");//"http://depositmo.eprints.org/id/d
      ocument/1736641");
            return;
        }
```

4.12.3.10 void uk.ac.soton.ses.SwordClientHandlerTests.TestPutToDocument (SwordClientHandler *sword2Handler*) `[private]`

Definition at line 137 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordClientHandler.PostDocxToRepository(), and uk.ac.soton.ses.SwordClientHandler.PutToDocument().

```
        {
            Debug.WriteLine("Depositing new .docx");
            //XmlDocument atomResponse = sword2Handler.DepositDocxToEprint(@"P:\t
      mp\hy.docx");
            string locationHeader = sword2Handler.PostDocxToRepository(@"P:\tmp\h
      y.docx");

            Debug.WriteLine("Initialising Atom response reader");
           /*
            SwordAtomReader sar = new SwordAtomReader(atomResponse);

            Debug.WriteLine("Title was " + sar.AtomTitle);
            Debug.WriteLine("Edit media href is " + sar.EditMediaHref);
            Debug.WriteLine("Contents href is " + sar.ContentsHref);*/
            /*
            XmlDocument dereference = sword2Handler.GetAtomXml(sar.EditMediaHref)
      ;
            XmlDocument subdocument = new XmlDocument();
            XmlNamespaceManager xmlns = new XmlNamespaceManager(dereference.NameT
      able);
            //xmlns.AddNamespace("default", "");
            //xmlns.AddNamespace("atom", "http://www.w3.org/2005/Atom");
            xmlns.AddNamespace(String.Empty, "http://www.w3.org/2005/Atom");
            subdocument.AppendChild(dereference.SelectSingleNode("//feed", xmlns)
);
            SwordAtomReader sar2 = new SwordAtomReader(subdocument);
            */


            Debug.WriteLine("Attempting to update document after a few seconds ..
.");
            Thread.Sleep(30 * 1000);
            //bool result = sword2Handler.PutToDocument(@"P:\tmp\hyb.docx", sar.E
ditMediaHref);
            bool result = sword2Handler.PutToDocument(@"P:\tmp\hyb.docx", locatio
nHeader); //sar.ContentsHref);

            // note that until at least 15th December 2010, this response is not
Atom XML so
            // the following is/was not valid. However, the update to the title w
as occurring

            Debug.WriteLine("Update result: " + result.ToString());
        }
```

4.12.3.11 void uk.ac.soton.ses.SwordClientHandlerTests.TestResolveInbox (SwordClientHandler *sword2Handler*) [private]

Definition at line 173 of file SwordClientHandlerTests.cs.

References uk.ac.soton.ses.SwordClientHandler._resolveInbox, and uk.ac.soton.ses.SwordClientHandler.ResolveInbox().

```
        {
            bool origvalue = sword2Handler._resolveInbox;
            sword2Handler._resolveInbox = true;
            sword2Handler.ResolveInbox();
            // leave things as we found them
            sword2Handler._resolveInbox = origvalue;
        }
```

4.12.4 Member Data Documentation

4.12.4.1 XmlWriter uk.ac.soton.ses.SwordClientHandlerTests.consoleWriter = null [private]

Definition at line 39 of file SwordClientHandlerTests.cs.

4.12.4.2 internal string uk.ac.soton.ses.SwordClientHandlerTests.EPRINT_TO_DELETE = "http://depositmo.eprints.org/id/eprint/58" [private]

Definition at line 36 of file SwordClientHandlerTests.cs.

4.12.4.3 XmlDocument uk.ac.soton.ses.SwordClientHandlerTests.response = null [private]

Definition at line 40 of file SwordClientHandlerTests.cs.

4.12.4.4 string uk.ac.soton.ses.SwordClientHandlerTests.sword2Endpoint = null [private]

Definition at line 34 of file SwordClientHandlerTests.cs.

4.12.4.5 string uk.ac.soton.ses.SwordClientHandlerTests.sword2Password = null [private]

Definition at line 33 of file SwordClientHandlerTests.cs.

4.12.4.6 string uk.ac.soton.ses.SwordClientHandlerTests.sword2User = null [private]

Definition at line 32 of file SwordClientHandlerTests.cs.

4.12.4.7 XmlWriterSettings uk.ac.soton.ses.SwordClientHandlerTests.xws = new XmlWriterSettings() [private]

Definition at line 38 of file SwordClientHandlerTests.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/SwordClientHandlerTests.cs

4.13 uk.ac.soton.ses.SwordListAuthor Class Reference

SWORD author.

Public Member Functions

- SwordListAuthor (string name)
  Author.

Properties

- string Name [get, set]
  Gets or sets the author's name.

Private Attributes

- string name

4.13.1 Detailed Description

SWORD author.

Definition at line 28 of file SwordListReader.cs.

4.13.2 Constructor & Destructor Documentation

4.13.2.1 uk.ac.soton.ses.SwordListAuthor.SwordListAuthor (string *name*)

Author.

Parameters

name Author's name

Definition at line 41 of file SwordListReader.cs.

```
        {
            this.Name = name;
        }
```

4.13.3 Member Data Documentation

4.13.3.1 string uk.ac.soton.ses.SwordListAuthor.name `[private]`

Definition at line 30 of file SwordListReader.cs.

4.13.4 Property Documentation

4.13.4.1 string uk.ac.soton.ses.SwordListAuthor.Name `[get, set]`

Gets or sets the author's name.

Definition at line 35 of file SwordListReader.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/SwordListReader.cs

4.14 uk.ac.soton.ses.SwordListEntry Class Reference

SWORD entry for the collection list.

Public Member Functions

- void AddAuthor (string name)

  Adds the author
  `name`
  to the entry.

- SwordListEntry (string title)

  Creates a new entry for the collection with the supplied title.

Properties

- string Title `[get, set]`
  Title of the entry.
- string Href `[get, set]`
  Href to the entry.
- string Summary `[get, set]`
  Summary of the entry.
- DateTime Updated `[get, set]`
  Update time of the entry.
- string Id `[get, set]`
  ID of the entry.
- List< SwordListAuthor > Authors `[get]`
  List of authors for the entry.

Private Attributes

- string title
- string href
- string summary
- DateTime updated
- string id
- List< SwordListAuthor > authors

4.14.1 Detailed Description

SWORD entry for the collection list.

Definition at line 50 of file SwordListReader.cs.

4.14.2 Constructor & Destructor Documentation

4.14.2.1 uk.ac.soton.ses.SwordListEntry.SwordListEntry (string *title*)

Creates a new entry for the collection with the supplied title.

Parameters

title Entry title

Definition at line 103 of file SwordListReader.cs.

```
{
    this.authors = new List<SwordListAuthor>();
    this.Title = title;
}
```

4.14.3 Member Function Documentation

4.14.3.1 void uk.ac.soton.ses.SwordListEntry.AddAuthor (string *name*)

Adds the author

```
name
```

to the entry.

Parameters

name Author name

Definition at line 93 of file SwordListReader.cs.

Referenced by uk.ac.soton.ses.SwordListReader.ParseEntries().

```
        {
            SwordListAuthor author = new SwordListAuthor(name);
            this.authors.Add(author);
        }
```

4.14.4 Member Data Documentation

4.14.4.1 List<SwordListAuthor> uk.ac.soton.ses.SwordListEntry.authors `[private]`

Definition at line 57 of file SwordListReader.cs.

4.14.4.2 string uk.ac.soton.ses.SwordListEntry.href `[private]`

Definition at line 53 of file SwordListReader.cs.

4.14.4.3 string uk.ac.soton.ses.SwordListEntry.id `[private]`

Definition at line 56 of file SwordListReader.cs.

4.14.4.4 string uk.ac.soton.ses.SwordListEntry.summary `[private]`

Definition at line 54 of file SwordListReader.cs.

4.14.4.5 string uk.ac.soton.ses.SwordListEntry.title `[private]`

Definition at line 52 of file SwordListReader.cs.

4.14.4.6 DateTime uk.ac.soton.ses.SwordListEntry.updated `[private]`

Definition at line 55 of file SwordListReader.cs.

4.14.5 Property Documentation

4.14.5.1 List<SwordListAuthor> uk.ac.soton.ses.SwordListEntry.Authors `[get]`

List of authors for the entry.

Definition at line 87 of file SwordListReader.cs.

4.14.5.2 string uk.ac.soton.ses.SwordListEntry.Href `[get, set]`

Href to the entry.

Definition at line 67 of file SwordListReader.cs.

4.14.5.3 string uk.ac.soton.ses.SwordListEntry.Id `[get, set]`

ID of the entry.

Definition at line 82 of file SwordListReader.cs.

Referenced by uk.ac.soton.ses.SwordListReader.GetEntryById(), and uk.ac.soton.ses.SwordListReader.ParseEntries().

4.14.5.4 string uk.ac.soton.ses.SwordListEntry.Summary `[get, set]`

Summary of the entry.

Definition at line 72 of file SwordListReader.cs.

Referenced by uk.ac.soton.ses.SwordListReader.ParseEntries().

4.14.5.5 string uk.ac.soton.ses.SwordListEntry.Title `[get, set]`

Title of the entry.

Definition at line 62 of file SwordListReader.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DumpListingToLog(), and uk.ac.soton.ses.SwordClientHandlerTests.TestList().

4.14.5.6 DateTime uk.ac.soton.ses.SwordListEntry.Updated `[get, set]`

Update time of the entry.

Definition at line 77 of file SwordListReader.cs.

Referenced by uk.ac.soton.ses.Word2010DepositMOAddIn.DepositMORepositoryControl.DumpListingToLog(), and uk.ac.soton.ses.SwordListReader.ParseEntries().

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/SwordListReader.cs

4.15 uk.ac.soton.ses.SwordListReader Class Reference

SWORD list reader.

Public Member Functions

- SwordListReader (XmlDocument swordListXml)

  Creates a new SwordListReader from the supplied listing document.

- SwordListEntry GetEntryById (string id)

  Returns the entry that matches the supplied ID.

Properties

- List< SwordListEntry > Entries `[get]`

  List of entries in the collection.

Private Member Functions

- internal void ParseEntries ()

  Parses the entries in the supplied document and populates the list.

Private Attributes

- XmlDocument swordListXml = null
- XmlNamespaceManager xnm = null
- string title
- DateTime updated
- List< SwordListEntry > entries = new List<SwordListEntry>()

4.15.1 Detailed Description

SWORD list reader.

Definition at line 113 of file SwordListReader.cs.

4.15.2 Constructor & Destructor Documentation

4.15.2.1 uk.ac.soton.ses.SwordListReader.SwordListReader (XmlDocument *swordListXml*)

Creates a new SwordListReader from the supplied listing document.

Parameters

swordListXml XML document containing list of entries

Definition at line 132 of file SwordListReader.cs.

```
{
    this.swordListXml = swordListXml;
    if (this.swordListXml != null)
    {
        this.xnm = new XmlNamespaceManager(this.swordListXml.NameTable);

        this.xnm.AddNamespace("atom", "http://www.w3.org/2005/Atom");

    }
    this.ParseEntries();
}
```

4.15.3 Member Function Documentation

4.15.3.1 SwordListEntry uk.ac.soton.ses.SwordListReader.GetEntryById (string *id*)

Returns the entry that matches the supplied ID.

Parameters

id ID to match

Returns

Matching entry, else null

Definition at line 149 of file SwordListReader.cs.

References uk.ac.soton.ses.SwordListEntry.Id.

```
{
    foreach (SwordListEntry sle in this.Entries)
    {
        if (sle.Id.Equals(id))
        {
            return sle;
        }
    }
    return null;
}
```

4.15.3.2 internal void uk.ac.soton.ses.SwordListReader.ParseEntries ( ) `[private]`

Parses the entries in the supplied document and populates the list.

Definition at line 164 of file SwordListReader.cs.

References uk.ac.soton.ses.SwordListEntry.AddAuthor(), uk.ac.soton.ses.SwordListEntry.Id, uk.ac.soton.ses.SwordListEntry.Summary, and uk.ac.soton.ses.SwordListEntry.Updated.

```
{
    if (this.swordListXml == null)
    {
        // don’t parse a null document
        return;
    }
```

```
      this.title = this.swordListXml.SelectSingleNode("/atom:feed/atom:titl
e", this.xnm).InnerText;
      this.updated = DateTime.Parse(this.swordListXml.SelectSingleNode("/at
om:feed/atom:updated", this.xnm).InnerText);

      // get entries
      XmlNodeList nodeList = this.swordListXml.SelectNodes("/atom:feed/atom
:entry", this.xnm);
      foreach (XmlNode node in nodeList)
      {
          SwordListEntry sle = new SwordListEntry(node.SelectSingleNode("at
om:title", this.xnm).InnerText);
          sle.Updated = DateTime.Parse(node.SelectSingleNode("atom:updated"
, this.xnm).InnerText);
          sle.Id = node.SelectSingleNode("atom:id", this.xnm).InnerText;
          sle.Summary = node.SelectSingleNode("atom:summary", this.xnm).Inn
erText;
          XmlNodeList authorList = node.SelectNodes("atom:author", this.
xnm);
          foreach (XmlNode author in authorList)
          {
              sle.AddAuthor(author.SelectSingleNode("atom:name", this.xnm).
InnerText);
          }
          this.Entries.Add(sle);
      }
      return;
  }
```

4.15.4 Member Data Documentation

4.15.4.1 List<SwordListEntry> uk.ac.soton.ses.SwordListReader.entries = new List<SwordListEntry>() `[private]`

Definition at line 121 of file SwordListReader.cs.

4.15.4.2 XmlDocument uk.ac.soton.ses.SwordListReader.swordListXml = null `[private]`

Definition at line 115 of file SwordListReader.cs.

4.15.4.3 string uk.ac.soton.ses.SwordListReader.title `[private]`

Definition at line 118 of file SwordListReader.cs.

4.15.4.4 DateTime uk.ac.soton.ses.SwordListReader.updated `[private]`

Definition at line 119 of file SwordListReader.cs.

4.15.4.5 XmlNamespaceManager uk.ac.soton.ses.SwordListReader.xnm = null `[private]`

Definition at line 116 of file SwordListReader.cs.

4.15.5 Property Documentation

4.15.5.1 List<SwordListEntry> uk.ac.soton.ses.SwordListReader.Entries `[get]`

List of entries in the collection.

Definition at line 126 of file SwordListReader.cs.

The documentation for this class was generated from the following file:

- Q:/Projects/SwordHandler/SwordHandler/SwordListReader.cs

4.16 uk.ac.soton.ses.Word2010DepositMOAddIn.ThisRibbonCollection Class Reference

Properties

- internal Word2010DepositMORibbon Word2010DepositMORibbon `[get]`

Private Member Functions

- internal ThisRibbonCollection (global::Microsoft.Office.Tools.Ribbon.RibbonFactory factory)

4.16.1 Detailed Description

Definition at line 231 of file Word2010DepositMOAddIn.Designer.cs.

4.16.2 Constructor & Destructor Documentation

4.16.2.1 internal uk.ac.soton.ses.Word2010DepositMOAddIn.ThisRibbonCollection.ThisRibbonCollection (global::Microsoft.Office.Tools.Ribbon.RibbonFactory *factory*) `[private]`

Definition at line 234 of file Word2010DepositMOAddIn.Designer.cs.

```
              :
        base(factory) {
}
```

4.16.3 Property Documentation

4.16.3.1 internal Word2010DepositMORibbon uk.ac.soton.ses.Word2010DepositMOAddIn.ThisRibbonCollection.Word2010DepositMORibbon `[get, private]`

Definition at line 196 of file Word2010DepositMORibbon.Designer.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMOAddIn.Designe
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMORibbon.Design

4.17 uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddI Class Reference

Entry point for the Microsoft Word 2010 DepositMO add-in.

Public Member Functions

- Word2010DepositMOAddIn (global::Microsoft.Office.Tools.Word.ApplicationFactory factory, global::System.IServiceProvider serviceProvider)

Protected Member Functions

- override void Initialize ()
- override void FinishInitialization ()
- override void InitializeDataBindings ()
- override void OnShutdown ()

Properties

- internal string DefaultEndpoint `[get, set]`

  Gets or sets the default endpoint.

- internal string DefaultUsername `[get, set]`

  Gets or sets the default username.

- internal string DefaultPassword `[get, set]`

  Gets or sets the default password.

- internal uint SubmissionThreadCount `[get]`

  Gets the number of active submission threads.

Private Member Functions

- internal Core.DocumentProperties GetActiveDocumentProperties ()

  Gets the properties of the active document.

- internal void IncrementSubmissionThreadCount ()

  Increments the thread count.

- internal void DecrementSubmissionThreadCount ()

  Decrements the thread count.

- string GetActiveSubmissionString ()

  Returns a formatted string containing the current pending submission summary.

- internal void LogMessage (string message)

  Logs the message
  `message`
  to Word's status bar, any debug tracers and the log string list on this instance.

- void AddMessageToLogStack (string message)

  Adds the message to the log stack, and purges the log stack if the size exceeds the maximum.

- internal string[ ] GetAllLogMessages ()

  Returns the log stack as a string array.

- internal void SetSingleLineTextBox (string text, TextBox textBox)

  Sets the value of
  `textBox`
  to
  `text`
  , in a thread-safe manner.

- delegate void SetTextCallback (string message, TextBox textBox)

  Text set callback delegate for
  `textBox`
  .

- internal void LogMessage (string message, TextBox textBox)

  Logs the message to the internal stack, debug output (if appropriate) and the supplied text box.

- internal void MakeAllInvisible ()

  For every task pane that can be made invisible, make it invisible.

- internal void MakeAllVisible ()

  Makes each available task pane and any child controls visible, and shows the associated control.

- void ThisAddIn_Startup (object sender, System.EventArgs e)

  Called when the plug-in is started.

- void AddRepositoryControl ()

  Adds a new repository control panel to the list and adds it to the custom task panes group.

- void Application_DocumentOpen (Word.Document Doc)

  Event method to be called when a document is opened (or created, or focus switched, the latter of which is handled by the repository control add method.

- void ThisAddIn_Shutdown (object sender, System.EventArgs e)

  Event triggered when the add-in is shut down.

- void InternalStartup ()

  Required method for Designer support - do not modify the contents of this method with the code editor.

- void InitializeCachedData ()

- void InitializeData ()
- void BindToData ()
- void StartCaching (string MemberName)
- void StopCaching (string MemberName)
- bool IsCached (string MemberName)
- void BeginInitialization ()
- void EndInitialization ()
- void InitializeControls ()
- void InitializeComponents ()
- bool NeedsFill (string MemberName)

Private Attributes

- string defaultEndpoint = "http://depositmo.eprints.org/id/contents"

  The default endpoint for new controls (could be "").

- string defaultUsername = ""

  The default username for new controls (could be "").

- string defaultPassword = ""

  The default password for new controls (could be "").

- internal List< DepositMORepositoryControl > repositoryControls = new List<DepositMORepositoryControl>()

  A list of all the repository controls the addin deals with.

- uint submissionThreadCounter = 0

  The number of active submission threads.

- List< String > log = new List<string>()

  Internal log buffer.

- internal Microsoft.Office.Tools.CustomTaskPaneCollection CustomTaskPanes
- internal Microsoft.Office.Tools.SmartTagCollection VstoSmartTags
- global::System.Object missing = global::System.Type.Missing
- internal Microsoft.Office.Interop.Word.Application Application

Static Private Attributes

- static readonly object lockobj = new object()

  Lock object for threading.

- static int MAX_LOG_LINES = 32 ∗ 1024

  Maximum number of lines to store in the internal log buffer.

- static int PURGE_NUMBER_OF_LOG_LINES = 1 ∗ 1024

  Number of lines by which to reduce the log buffer if `MAX_LOG_LINES` *is reached.*

4.17.1 Detailed Description

Entry point for the Microsoft Word 2010 DepositMO add-in.

Definition at line 30 of file Word2010DepositMOAddIn.cs.

4.17.2 Constructor & Destructor Documentation

4.17.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.Word2010DepositMOAddIn (global::Microsoft.Office.Tools.Word.ApplicationFactory *factory*, global::System.IServiceProvider *serviceProvider*)

Definition at line 33 of file Word2010DepositMOAddIn.Designer.cs.

```
                                                                :
        base(factory, serviceProvider, "AddIn", "ThisAddIn") {
    Globals.Factory = factory;
}
```

4.17.3 Member Function Documentation

4.17.3.1 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.AddMessageToLogStack (string *message*) `[private]`

Adds the message to the log stack, and purges the log stack if the size exceeds the maximum.

Parameters

message Message to log

Definition at line 163 of file Word2010DepositMOAddIn.cs.

```
{
    this.log.Add(message);
    if (this.log.Count > MAX_LOG_LINES)
    {
        this.log.RemoveRange(0, PURGE_NUMBER_OF_LOG_LINES);
    }
}
```

4.17.3.2 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.AddRepositoryControl ( ) `[private]`

Adds a new repository control panel to the list and adds it to the custom task panes group.

Definition at line 306 of file Word2010DepositMOAddIn.cs.

```
{
    if (this.Application.Documents.Count == 0)
    {
        // COM will throw an exception if we try to do anything with Acti
veWindow (even a test for null)
          // if there are no open documents
          return;
      }

      foreach (CustomTaskPane ctp in this.CustomTaskPanes)
      {
          if (ctp != null && ctp.Window != null && ctp.Control != null && !
ctp.Control.Disposing && ctp.Window == this.Application.ActiveWindow)
          {
              // we already have a control for this task pane so don't cont
inue to add a new one
              return;
          }
      }

      // create a new repository control, add it to the internal list and a
dd it to
      // a custom task pane associated with this document window
      DepositMORepositoryControl dmrc = new DepositMORepositoryControl();
      this.repositoryControls.Add(dmrc);
      this.CustomTaskPanes.Add(dmrc, "Repository control", this.
Application.ActiveWindow);
  }
```

4.17.3.3 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.Application_DocumentOpen (Word.Document *Doc*) [private]

Event method to be called when a document is opened (or created, or focus switched, the latter of which is handled by the repository control add method.

Parameters

Doc The associated Word document

Definition at line 336 of file Word2010DepositMOAddIn.cs.

```
  {
      // add a new repository control when a new document has been created
      // (not ideal, but Word will otherwise end up with multiple panels pe
r document)
      this.AddRepositoryControl();
  }
```

4.17.3.4 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.BeginInitialization ( ) [private]

Definition at line 124 of file Word2010DepositMOAddIn.Designer.cs.

```
                                    {
      this.BeginInit();
      this.CustomTaskPanes.BeginInit();
      this.VstoSmartTags.BeginInit();
  }
```

4.17.3.5 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.BindToData ( ) [private]

Definition at line 96 of file Word2010DepositMOAddIn.Designer.cs.

```
                           {
}
```

4.17.3.6 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.DecrementSubmissionThrea ( ) [private]

Decrements the thread count.

Definition at line 109 of file Word2010DepositMOAddIn.cs.

```
{
    lock (lockobj)
    {
        this.submissionThreadCounter--;
        this.LogMessage(this.GetActiveSubmissionString());
    }
}
```

4.17.3.7 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.EndInitialization ( ) [private]

Definition at line 134 of file Word2010DepositMOAddIn.Designer.cs.

```
                                  {
    this.VstoSmartTags.EndInit();
    this.CustomTaskPanes.EndInit();
    this.EndInit();
}
```

4.17.3.8 override void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.FinishInitialization ( ) [protected]

Definition at line 57 of file Word2010DepositMOAddIn.Designer.cs.

```
                                                {
    this.InternalStartup();
    this.OnStartup();
}
```

4.17.3.9 internal Core.DocumentProperties uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.GetActiveDocumentProperti ( ) [private]

Gets the properties of the active document.

Returns

Definition at line 73 of file Word2010DepositMOAddIn.cs.

```
        {
            return (Core.DocumentProperties)Globals.Word2010DepositMOAddIn.Applic
      ation.ActiveDocument.BuiltInDocumentProperties;
        }
```

4.17.3.10 string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.GetActiveSubmissionStrin ( ) [private]

Returns a formatted string containing the current pending submission summary.

Returns

Definition at line 122 of file Word2010DepositMOAddIn.cs.

```
        {
            return String.Format("{0} submission{1} currently active", this.
      SubmissionThreadCount, (this.SubmissionThreadCount == 1) ? "" : "s");
        }
```

4.17.3.11 internal string [] uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.GetAllLogMessages ( ) [private]

Returns the log stack as a string array.

Returns

String array of log messages

TODO: add some size management here

Definition at line 176 of file Word2010DepositMOAddIn.cs.

```
        {
            return this.log.ToArray();
        }
```

4.17.3.12 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.IncrementSubmissionThrea ( ) [private]

Increments the thread count.

Definition at line 97 of file Word2010DepositMOAddIn.cs.

```
        {
            lock (lockobj)
            {
                this.submissionThreadCounter++;
                this.LogMessage(this.GetActiveSubmissionString());
            }
        }
```

4.17.3.13 override void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.Initialize ( ) [protected]

Definition at line 42 of file Word2010DepositMOAddIn.Designer.cs.

```
                                         {
            base.Initialize();
            this.Application = this.GetHostItem<Microsoft.Office.Interop.Word.App
      lication>(typeof(Microsoft.Office.Interop.Word.Application), "Application");
            Globals.Word2010DepositMOAddIn = this;
            global::System.Windows.Forms.Application.EnableVisualStyles();
            this.InitializeCachedData();
            this.InitializeControls();
            this.InitializeComponents();
            this.InitializeData();
        }
```

4.17.3.14 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.InitializeCachedData ( ) [private]

Definition at line 76 of file Word2010DepositMOAddIn.Designer.cs.

```
                                        {
            if ((this.DataHost == null)) {
                return;
            }
            if (this.DataHost.IsCacheInitialized) {
                this.DataHost.FillCachedData(this);
            }
        }
```

4.17.3.15 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.InitializeComponents ( ) [private]

Definition at line 153 of file Word2010DepositMOAddIn.Designer.cs.

```
                                    {
        }
```

4.17.3.16 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.InitializeControls ( ) [private]

Definition at line 144 of file Word2010DepositMOAddIn.Designer.cs.

```
                                  {
            this.CustomTaskPanes = Globals.Factory.CreateCustomTaskPaneCollection
      (null, null, "CustomTaskPanes", "CustomTaskPanes", this);
            this.VstoSmartTags = Globals.Factory.CreateSmartTagCollection(null, n
      ull, "VstoSmartTags", "VstoSmartTags", this);
        }
```

4.17.3.17 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.InitializeData ( ) [private]

Definition at line 89 of file Word2010DepositMOAddIn.Designer.cs.

```
                              {
        }
```

4.17.3.18 override void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.InitializeDataBindings ( ) [protected]

Definition at line 66 of file Word2010DepositMOAddIn.Designer.cs.

```
                                              {
            this.BeginInitialization();
            this.BindToData();
            this.EndInitialization();
        }
```

4.17.3.19 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.InternalStartup ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 368 of file Word2010DepositMOAddIn.cs.

```
        {
            this.Startup += new System.EventHandler(ThisAddIn_Startup);
            this.Shutdown += new System.EventHandler(ThisAddIn_Shutdown);
        }
```

4.17.3.20 bool uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.IsCached (string *MemberName*) `[private]`

Definition at line 116 of file Word2010DepositMOAddIn.Designer.cs.

```
                                            {
            return this.DataHost.IsCached(this, MemberName);
        }
```

4.17.3.21 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.LogMessage (string *message,* TextBox *textBox*) `[private]`

Logs the message to the internal stack, debug output (if appropriate) and the supplied text box.

Thread-safe

Parameters

message The message to log

textBox The text box to which to log

Definition at line 215 of file Word2010DepositMOAddIn.cs.

```
        {
            if (textBox.InvokeRequired)
            {
                SetTextCallback tc = new SetTextCallback(LogMessage);
                textBox.Invoke(tc, new Object[] { message, textBox });
            }
            else
            {
                if (textBox != null && !textBox.Disposing && !textBox.IsDisposed)

                {
                    textBox.AppendText(message + Environment.NewLine);
                }

                // call the regular logger only if the invoke isn't required to
                // avoid multiple messages appearing in the other logs
                this.LogMessage(message);
            }
        }
```

4.17.3.22 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.LogMessage (string *message*) `[private]`

Logs the message

```
message
```

to Word's status bar, any debug tracers and the log string list on this instance.

Parameters

message The log message

Definition at line 151 of file Word2010DepositMOAddIn.cs.

```
{
    Globals.Word2010DepositMOAddIn.Application.StatusBar = message;
    Debug.WriteLine(message);
    this.AddMessageToLogStack(message);
}
```

4.17.3.23 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.MakeAllInvisible ( ) [private]

For every task pane that can be made invisible, make it invisible.

Definition at line 239 of file Word2010DepositMOAddIn.cs.

```
{
    foreach (CustomTaskPane c in this.CustomTaskPanes)
    {
        if (c != null)
        {
            c.Visible = false;
        }
    }
}
```

4.17.3.24 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.MakeAllVisible ( ) [private]

Makes each available task pane and any child controls visible, and shows the associated control.

Also cleans up task panes where the child control is null.

Definition at line 255 of file Word2010DepositMOAddIn.cs.

```
  {
      // make all of the custom task panes visible, along with their contro
ls
      foreach (CustomTaskPane c in this.CustomTaskPanes)
      {
          if (c != null)
          {
              c.Visible = true;
              if (c.Control != null)
              {
                  c.Control.Show();
                  c.Control.Visible = true;
              }
              else
              {
                  // the control itself is null, so dispose of the pane
                  c.Dispose();
              }
          }
```

```
            }
            // ensure the repository controls are also visible
            foreach (DepositMORepositoryControl dmrc in this.repositoryControls)
            {
                dmrc.Visible = true;
                dmrc.Show();
            }
        }
```

4.17.3.25 bool uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.NeedsFill (string *MemberName*) `[private]`

Definition at line 159 of file Word2010DepositMOAddIn.Designer.cs.

```
                                              {
            return this.DataHost.NeedsFill(this, MemberName);
        }
```

4.17.3.26 override void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.OnShutdown ( ) `[protected]`

Definition at line 167 of file Word2010DepositMOAddIn.Designer.cs.

```
                                         {
            this.VstoSmartTags.Dispose();
            this.CustomTaskPanes.Dispose();
            base.OnShutdown();
        }
```

4.17.3.27 internal void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.SetSingleLineTextBox (string *text,* TextBox *textBox*) `[private]`

Sets the value of

`textBox`

to

`text`

, in a thread-safe manner.

Parameters

text The target value for

`textBox`

textBox The text box form control to set

Definition at line 188 of file Word2010DepositMOAddIn.cs.

```
        {
            if (textBox.InvokeRequired)
            {
                SetTextCallback tc = new SetTextCallback(SetSingleLineTextBox);
                textBox.Invoke(tc, new Object[] { text, textBox });
            }
            else
            {
                textBox.Text = text;
            }
        }
```

4.17.3.28 delegate void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.SetTextCallback (string *message,* TextBox *textBox*) `[private]`

Text set callback delegate for

```
textBox
```

.

Needed for invocation by thread-safe text box value updater

Parameters

message

textBox

4.17.3.29 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.StartCaching (string *MemberName*) `[private]`

Definition at line 102 of file Word2010DepositMOAddIn.Designer.cs.

```
                                               {
            this.DataHost.StartCaching(this, MemberName);
        }
```

4.17.3.30 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.StopCaching (string *MemberName*) `[private]`

Definition at line 109 of file Word2010DepositMOAddIn.Designer.cs.

```
                                              {
            this.DataHost.StopCaching(this, MemberName);
        }
```

4.17.3.31 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.ThisAddIn_-Shutdown (object *sender,* System.EventArgs *e*) `[private]`

Event triggered when the add-in is shut down.

Note that we will wait briefly for threads to clean up here, but this will 'lock' Word when the user tries to close if a submission is in limbo (and causing Task Manager-level 'force quits'). As at this point the user won't have received a 'successfully submitted' message, and there's no guarantee that the document has been submitted. Word itself will terminate the threads after a short while

Parameters

sender Sender object

e Event arguments

Definition at line 352 of file Word2010DepositMOAddIn.cs.

```
        {
            string statusText = "Shutting down DepositMO plugin";
            if (this.SubmissionThreadCount > 0)
            {
                statusText += String.Format("; {0} - you will have to check these
       manually", this.GetActiveSubmissionString());
            }
            this.Application.StatusBar = statusText;
        }
```

4.17.3.32 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.ThisAddIn_-Startup (object *sender,* System.EventArgs *e*) `[private]`

Called when the plug-in is started.

Parameters

sender Sender

e Event arguments

Definition at line 288 of file Word2010DepositMOAddIn.cs.

```
        {
            // when the addin is started, create a new repository control
            // and place an event to repeat this when a document is opened
            this.AddRepositoryControl();

            // no independent event handler for a new document, only [new|open|fo
      cus] or [open]
            //this.Application.DocumentOpen += new Word.ApplicationEvents4_Docume
      ntOpenEventHandler(Application_DocumentOpen);
            this.Application.DocumentChange += new Word.ApplicationEvents4_Docume
      ntChangeEventHandler(AddRepositoryControl);

            // log version and timestamp for diagnostics
            this.LogMessage("Started " + System.Reflection.Assembly.GetExecutingA
      ssembly().FullName + " at " + DateTime.Now);
        }
```

4.17.4 Member Data Documentation

4.17.4.1 internal Microsoft.Office.Interop.Word.Application uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.Application `[private]`

Definition at line 28 of file Word2010DepositMOAddIn.Designer.cs.

4.17.4.2 internal Microsoft.Office.Tools.CustomTaskPaneCollection uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.CustomTaskPanes `[private]`

Definition at line 20 of file Word2010DepositMOAddIn.Designer.cs.

4.17.4.3 string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.defaultEndpoint = "http://depositmo.eprints.org/id/contents" `[private]`

The default endpoint for new controls (could be "").

Definition at line 36 of file Word2010DepositMOAddIn.cs.

4.17.4.4 string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.defaultPassword = "" `[private]`

The default password for new controls (could be "").

Definition at line 46 of file Word2010DepositMOAddIn.cs.

4.17.4.5 string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.defaultUsername = "" `[private]`

The default username for new controls (could be "").

Definition at line 41 of file Word2010DepositMOAddIn.cs.

4.17.4.6 readonly object uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.lockobj = new object() `[static, private]`

Lock object for threading.

Definition at line 82 of file Word2010DepositMOAddIn.cs.

4.17.4.7 List<String> uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.log = new List<string>() `[private]`

Internal log buffer.

Definition at line 144 of file Word2010DepositMOAddIn.cs.

4.17.4.8 int uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.MAX_-LOG_LINES = 32 ∗ 1024 `[static, private]`

Maximum number of lines to store in the internal log buffer.

Definition at line 133 of file Word2010DepositMOAddIn.cs.

4.17.4.9 global::System.Object uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.missing = global::System.Type.Missing `[private]`

Definition at line 25 of file Word2010DepositMOAddIn.Designer.cs.

4.17.4.10 int uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.PURGE_-NUMBER_OF_LOG_LINES = 1 ∗ 1024 `[static, private]`

Number of lines by which to reduce the log buffer if

```
MAX_LOG_LINES
```

is reached.
Note this should be less than

```
MAX_LOG_LINES
```

.

Definition at line 139 of file Word2010DepositMOAddIn.cs.

4.17.4.11 internal List<DepositMORepositoryControl> uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.repositoryControls = new List<DepositMORepositoryControl>() `[private]`

A list of all the repository controls the addin deals with.

Definition at line 67 of file Word2010DepositMOAddIn.cs.

4.17.4.12 uint uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.submissionThreadCounter = 0 `[private]`

The number of active submission threads.

Definition at line 87 of file Word2010DepositMOAddIn.cs.

4.17.4.13 internal Microsoft.Office.Tools.SmartTagCollection uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.VstoSmartTags `[private]`

Definition at line 22 of file Word2010DepositMOAddIn.Designer.cs.

4.17.5 Property Documentation

4.17.5.1 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.DefaultEndpoint `[get, set, private]`

Gets or sets the default endpoint.

Definition at line 51 of file Word2010DepositMOAddIn.cs.

4.17.5.2 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.DefaultPassword `[get, set, private]`

Gets or sets the default password.

Definition at line 61 of file Word2010DepositMOAddIn.cs.

4.17.5.3 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.DefaultUsername `[get, set, private]`

Gets or sets the default username.

Definition at line 56 of file Word2010DepositMOAddIn.cs.

4.17.5.4 internal uint uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMOAddIn.SubmissionThreadCount `[get, private]`

Gets the number of active submission threads.

Definition at line 92 of file Word2010DepositMOAddIn.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMOAddIn.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMOAddIn.Designe

4.18 uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibb Class Reference

Ribbon interface for Microsoft Word 2010 DepositMO add-in.

Public Member Functions

- Word2010DepositMORibbon ()

  The constructor for the DepositMO Word ribbon.

Protected Member Functions

- override void Dispose (bool disposing)

  Clean up any resources being used.

Private Member Functions

- void Word2010DepositMORibbon_Load (object sender, RibbonUIEventArgs e)

  Event triggered when the ribbon is loaded.

- void Word2010DepositMOMainGroup_DialogLauncherClick (object sender, RibbonControlEventArgs e)

  Event triggered when the first group's dialogue launcher is selected (bottom-right miniature icon).

- void quickSubmissionRibbonGroup_DialogLauncherClick (object sender, RibbonControlEventArgs e)

  Event triggered when the quick submission dialogue launcher is selected (bottom-right miniature icon).

- void Word2010DepositMORepositoryButton_Click (object sender, RibbonControlEventArgs e)

  Event triggered when the show panel button is clicked.

- void Word2010DepositMORepositoryHideButton_Click (object sender, RibbonControlEventArgs e)

  Event triggered when the hide panel button is clicked.

- void quickSubmitButton_Click (object sender, RibbonControlEventArgs e)

  Event triggered when the quick submit button is clicked.

- void aboutRibbonButton_Click (object sender, RibbonControlEventArgs e)

  Event triggered when the about button is clicked.

- void InitializeComponent ()

  Required method for Designer support - do not modify the contents of this method with the code editor.

Private Attributes

- internal string quickUsername = null
- internal string quickPassword = null
- System.ComponentModel.IContainer components = null

  Required designer variable.

- internal Microsoft.Office.Tools.Ribbon.RibbonTab tab1
- Microsoft.Office.Tools.Ribbon.RibbonTab Word2010DepositMORibbonTab
- internal Microsoft.Office.Tools.Ribbon.RibbonGroup Word2010DepositMOMainGroup
- internal Microsoft.Office.Tools.Ribbon.RibbonButton Word2010DepositMORepositoryButton
- internal Microsoft.Office.Tools.Ribbon.RibbonButton Word2010DepositMORepositoryHideButton
- internal Microsoft.Office.Tools.Ribbon.RibbonGroup quickSubmissionRibbonGroup
- internal Microsoft.Office.Tools.Ribbon.RibbonEditBox endpointEditBox
- internal Microsoft.Office.Tools.Ribbon.RibbonButton quickSubmitButton
- internal Microsoft.Office.Tools.Ribbon.RibbonGroup aboutRibbonGroup
- internal Microsoft.Office.Tools.Ribbon.RibbonButton aboutRibbonButton

4.18.1 Detailed Description

Ribbon interface for Microsoft Word 2010 DepositMO add-in.

Definition at line 28 of file Word2010DepositMORibbon.cs.

4.18.2 Constructor & Destructor Documentation

4.18.2.1 uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORibbo ( )

The constructor for the DepositMO Word ribbon.

Definition at line 29 of file Word2010DepositMORibbon.Designer.cs.

```
            : base(Globals.Factory.GetRibbonFactory())
        {
            InitializeComponent();
        }
```

4.18.3 Member Function Documentation

4.18.3.1 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.aboutRibbonButton_Click (object *sender*, RibbonControlEventArgs *e*) [private]

Event triggered when the about button is clicked.

Launches the about box

Parameters

sender Event sender

e Event arguments

Definition at line 105 of file Word2010DepositMORibbon.cs.

```
        {
            AboutBox aboutBox = new AboutBox();
            aboutBox.ShowDialog((System.Windows.Forms.IWin32Window)Globals.Ribbon
      s.GetRibbon<Word2010DepositMORibbon>().Container);
        }
```

4.18.3.2 override void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Dispose (bool *disposing*) [protected]

Clean up any resources being used.

Parameters

disposing true if managed resources should be disposed; otherwise, false.

Definition at line 39 of file Word2010DepositMORibbon.Designer.cs.

```
        {
            if (disposing && (components != null))
            {
                components.Dispose();
            }
            base.Dispose(disposing);
        }
```

4.18.3.3 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.InitializeComponent ( ) [private]

Required method for Designer support - do not modify the contents of this method with the code editor.

Definition at line 54 of file Word2010DepositMORibbon.Designer.cs.

```
        {
            Microsoft.Office.Tools.Ribbon.RibbonDialogLauncher ribbonDialogLaunch
erImpl1 = this.Factory.CreateRibbonDialogLauncher();
            Microsoft.Office.Tools.Ribbon.RibbonDialogLauncher ribbonDialogLaunch
erImpl2 = this.Factory.CreateRibbonDialogLauncher();
            System.ComponentModel.ComponentResourceManager resources = new System
.ComponentModel.ComponentResourceManager(typeof(Word2010DepositMORibbon));
            this.tab1 = this.Factory.CreateRibbonTab();
            this.Word2010DepositMORibbonTab = this.Factory.CreateRibbonTab();
            this.Word2010DepositMOMainGroup = this.Factory.CreateRibbonGroup();
            this.Word2010DepositMORepositoryButton = this.Factory.CreateRibbonBut
ton();
            this.Word2010DepositMORepositoryHideButton = this.Factory.CreateRibbo
nButton();
            this.quickSubmissionRibbonGroup = this.Factory.CreateRibbonGroup();
            this.endpointEditBox = this.Factory.CreateRibbonEditBox();
            this.quickSubmitButton = this.Factory.CreateRibbonButton();
            this.aboutRibbonGroup = this.Factory.CreateRibbonGroup();
            this.aboutRibbonButton = this.Factory.CreateRibbonButton();
            this.tab1.SuspendLayout();
            this.Word2010DepositMORibbonTab.SuspendLayout();
            this.Word2010DepositMOMainGroup.SuspendLayout();
            this.quickSubmissionRibbonGroup.SuspendLayout();
            this.aboutRibbonGroup.SuspendLayout();
            //
            // tab1
            //
            this.tab1.ControlId.ControlIdType = Microsoft.Office.Tools.Ribbon.Rib
bonControlIdType.Office;
            this.tab1.Label = "TabAddIns";
            this.tab1.Name = "tab1";
```

```
      //
      // Word2010DepositMORibbonTab
      //
      this.Word2010DepositMORibbonTab.Groups.Add(this.
Word2010DepositMOMainGroup);
      this.Word2010DepositMORibbonTab.Groups.Add(this.
quickSubmissionRibbonGroup);
      this.Word2010DepositMORibbonTab.Groups.Add(this.aboutRibbonGroup);
      this.Word2010DepositMORibbonTab.KeyTip = "DEP";
      this.Word2010DepositMORibbonTab.Label = "DepositMO 2010";
      this.Word2010DepositMORibbonTab.Name = "Word2010DepositMORibbonTab";
      //
      // Word2010DepositMOMainGroup
      //
      this.Word2010DepositMOMainGroup.DialogLauncher = ribbonDialogLauncher
Impl1;
      this.Word2010DepositMOMainGroup.Items.Add(this.
Word2010DepositMORepositoryButton);
      this.Word2010DepositMOMainGroup.Items.Add(this.
Word2010DepositMORepositoryHideButton);
      this.Word2010DepositMOMainGroup.Label = "Repository control";
      this.Word2010DepositMOMainGroup.Name = "Word2010DepositMOMainGroup";
      //
      // Word2010DepositMORepositoryButton
      //
      this.Word2010DepositMORepositoryButton.ControlSize = Microsoft.Office
.Core.RibbonControlSize.RibbonControlSizeLarge;
      this.Word2010DepositMORepositoryButton.Image = global::uk.ac.soton.se
s.Word2010DepositMOAddIn.Properties.Resources.DocumentImage;
      this.Word2010DepositMORepositoryButton.Label = "Show";
      this.Word2010DepositMORepositoryButton.Name = "Word2010DepositMORepos
itoryButton";
      this.Word2010DepositMORepositoryButton.ScreenTip = "Displays the repo
sitory control panels if they aren\’t already shown";
      this.Word2010DepositMORepositoryButton.ShowImage = true;
      this.Word2010DepositMORepositoryButton.Click += new Microsoft.Office.
Tools.Ribbon.RibbonControlEventHandler(this.
Word2010DepositMORepositoryButton_Click);
      //
      // Word2010DepositMORepositoryHideButton
      //
      this.Word2010DepositMORepositoryHideButton.ControlSize = Microsoft.Of
fice.Core.RibbonControlSize.RibbonControlSizeLarge;
      this.Word2010DepositMORepositoryHideButton.Image = global::uk.ac.soto
n.ses.Word2010DepositMOAddIn.Properties.Resources.DocumentImage;
      this.Word2010DepositMORepositoryHideButton.Label = "Hide";
      this.Word2010DepositMORepositoryHideButton.Name = "Word2010DepositMOR
epositoryHideButton";
      this.Word2010DepositMORepositoryHideButton.ScreenTip = "Hides the rep
ository controls if they\’re visible";
      this.Word2010DepositMORepositoryHideButton.ShowImage = true;
      this.Word2010DepositMORepositoryHideButton.Click += new Microsoft.Off
ice.Tools.Ribbon.RibbonControlEventHandler(this.
Word2010DepositMORepositoryHideButton_Click);
      //
      // quickSubmissionRibbonGroup
      //
      this.quickSubmissionRibbonGroup.DialogLauncher = ribbonDialogLauncher
Impl2;
      this.quickSubmissionRibbonGroup.Items.Add(this.endpointEditBox);
      this.quickSubmissionRibbonGroup.Items.Add(this.quickSubmitButton);
      this.quickSubmissionRibbonGroup.Label = "Quick submission";
      this.quickSubmissionRibbonGroup.Name = "quickSubmissionRibbonGroup";
      //
      // endpointEditBox
      //
      this.endpointEditBox.Label = "Endpoint";
      this.endpointEditBox.Name = "endpointEditBox";
      this.endpointEditBox.SizeString = "http://depositmo.eprints.org/sword
-app/deposit/inbox";
      this.endpointEditBox.SuperTip = "The endpoint against which submissio
n will occur";
      this.endpointEditBox.Text = null;
      //
      // quickSubmitButton
      //
      this.quickSubmitButton.Image = global::uk.ac.soton.ses.Word2010Deposi
tMOAddIn.Properties.Resources.DocumentImage;
      this.quickSubmitButton.Label = "Submit";
      this.quickSubmitButton.Name = "quickSubmitButton";
      this.quickSubmitButton.ShowImage = true;
      this.quickSubmitButton.SuperTip = resources.GetString("quickSubmitBut
ton.SuperTip");
      this.quickSubmitButton.Click += new Microsoft.Office.Tools.Ribbon.Rib
bonControlEventHandler(this.quickSubmitButton_Click);
      //
      // aboutRibbonGroup
      //
      this.aboutRibbonGroup.Items.Add(this.aboutRibbonButton);
      this.aboutRibbonGroup.Label = "About";
      this.aboutRibbonGroup.Name = "aboutRibbonGroup";
      //
      // aboutRibbonButton
      //
      this.aboutRibbonButton.ControlSize = Microsoft.Office.Core.RibbonCont
rolSize.RibbonControlSizeLarge;
      this.aboutRibbonButton.Image = global::uk.ac.soton.ses.Word2010Deposi
tMOAddIn.Properties.Resources.DocumentImage;
      this.aboutRibbonButton.Label = "About DepositMO 2010";
      this.aboutRibbonButton.Name = "aboutRibbonButton";
      this.aboutRibbonButton.ShowImage = true;
      this.aboutRibbonButton.SuperTip = "Displays some information about th
e DepositMO Word author add-in";
      this.aboutRibbonButton.Click += new Microsoft.Office.Tools.Ribbon.Rib
bonControlEventHandler(this.aboutRibbonButton_Click);
      //
      // Word2010DepositMORibbon
      //
      this.Name = "Word2010DepositMORibbon";
      this.RibbonType = "Microsoft.Word.Document";
      this.Tabs.Add(this.tab1);
      this.Tabs.Add(this.Word2010DepositMORibbonTab);
      this.Load += new Microsoft.Office.Tools.Ribbon.RibbonUIEventHandler(t
his.Word2010DepositMORibbon_Load);
      this.tab1.ResumeLayout(false);
      this.tab1.PerformLayout();
      this.Word2010DepositMORibbonTab.ResumeLayout(false);
      this.Word2010DepositMORibbonTab.PerformLayout();
      this.Word2010DepositMOMainGroup.ResumeLayout(false);
      this.Word2010DepositMOMainGroup.PerformLayout();
      this.quickSubmissionRibbonGroup.ResumeLayout(false);
      this.quickSubmissionRibbonGroup.PerformLayout();
      this.aboutRibbonGroup.ResumeLayout(false);
      this.aboutRibbonGroup.PerformLayout();

  }
```

4.18.3.4 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.quickSubmissionRibbonGr DialogLauncherClick (object *sender,* RibbonControlEventArgs *e*) `[private]`

Event triggered when the quick submission dialogue launcher is selected (bottom-right miniature icon).

Brings up the quick submission form

Parameters

sender Event sender

e Event arguments

Definition at line 63 of file Word2010DepositMORibbon.cs.

```
        {
            QuickSubmitForm qsf = new QuickSubmitForm();
            // this really can't find the parent window!
            qsf.ShowDialog((System.Windows.Forms.IWin32Window)Globals.Ribbons.Get
      Ribbon<Word2010DepositMORibbon>().Container);
        }
```

4.18.3.5 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.quickSubmitButton_Click (object *sender,* RibbonControlEventArgs *e*) `[private]`

Event triggered when the quick submit button is clicked.

Parameters

sender Event sender

e Event arguments

Definition at line 95 of file Word2010DepositMORibbon.cs.

```
        {
            this.quickSubmissionRibbonGroup_DialogLauncherClick(sender, e);
        }
```

4.18.3.6 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMOMain DialogLauncherClick (object *sender,* RibbonControlEventArgs *e*) `[private]`

Event triggered when the first group's dialogue launcher is selected (bottom-right miniature icon).

Brings up the log viewer

Parameters

sender Event sender

e Event arguments

Definition at line 51 of file Word2010DepositMORibbon.cs.

```
        {
            LogViewerForm lvf = new LogViewerForm();
            lvf.Show();
        }
```

4.18.3.7 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORepos Click (object *sender,* RibbonControlEventArgs *e*) `[private]`

Event triggered when the show panel button is clicked.

Parameters

sender Event sender

e Event arguments

Definition at line 75 of file Word2010DepositMORibbon.cs.

```
        {
            Globals.Word2010DepositMOAddIn.MakeAllVisible();
        }
```

4.18.3.8 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORepos Click (object *sender,* RibbonControlEventArgs *e*) `[private]`

Event triggered when the hide panel button is clicked.

Parameters

sender Event sender

e Event arguments

Definition at line 85 of file Word2010DepositMORibbon.cs.

```
        {
            Globals.Word2010DepositMOAddIn.MakeAllInvisible();
        }
```

4.18.3.9 void uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORibbo Load (object *sender,* RibbonUIEventArgs *e*) `[private]`

Event triggered when the ribbon is loaded.

Parameters

- *sender* Event sender
- *e* Event arguments

Definition at line 38 of file Word2010DepositMORibbon.cs.

```
    {
        this.endpointEditBox.Text = Globals.Word2010DepositMOAddIn.DefaultEnd
  point;
        this.quickSubmissionRibbonGroup.DialogLauncherClick += new RibbonCont
  rolEventHandler(quickSubmissionRibbonGroup_DialogLauncherClick);
        this.Word2010DepositMOMainGroup.DialogLauncherClick += new RibbonCont
  rolEventHandler(Word2010DepositMOMainGroup_DialogLauncherClick);
    }
```

4.18.4 Member Data Documentation

4.18.4.1 internal Microsoft.Office.Tools.Ribbon.RibbonButton uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.aboutRibbonButton [private]

Definition at line 190 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.2 internal Microsoft.Office.Tools.Ribbon.RibbonGroup uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.aboutRibbonGroup [private]

Definition at line 189 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.3 System.ComponentModel.IContainer uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.components = null [private]

Required designer variable.

Definition at line 24 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.4 internal Microsoft.Office.Tools.Ribbon.RibbonEditBox uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.endpointEditBox [private]

Definition at line 187 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.5 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.quickPassword = null [private]

Definition at line 31 of file Word2010DepositMORibbon.cs.

4.18.4.6 internal Microsoft.Office.Tools.Ribbon.RibbonGroup uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.quickSubmissionRibbonGr `[private]`

Definition at line 186 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.7 internal Microsoft.Office.Tools.Ribbon.RibbonButton uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.quickSubmitButton `[private]`

Definition at line 188 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.8 internal string uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.quickUsername = null `[private]`

Definition at line 30 of file Word2010DepositMORibbon.cs.

4.18.4.9 internal Microsoft.Office.Tools.Ribbon.RibbonTab uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.tab1 `[private]`

Definition at line 181 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.10 internal Microsoft.Office.Tools.Ribbon.RibbonGroup uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMOMair `[private]`

Definition at line 183 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.11 internal Microsoft.Office.Tools.Ribbon.RibbonButton uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORepo `[private]`

Definition at line 184 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.12 internal Microsoft.Office.Tools.Ribbon.RibbonButton uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORepo `[private]`

Definition at line 185 of file Word2010DepositMORibbon.Designer.cs.

4.18.4.13 Microsoft.Office.Tools.Ribbon.RibbonTab uk.ac.soton.ses.Word2010DepositMOAddIn.Word2010DepositMORibbon.Word2010DepositMORibb `[private]`

Definition at line 182 of file Word2010DepositMORibbon.Designer.cs.

The documentation for this class was generated from the following files:

- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMORibbon.cs
- Q:/Projects/Word2010DepositMOAddIn/Word2010DepositMOAddIn/Word2010DepositMORibbon.Design

Index