Documentation about the Sharepoint Manager DLL

Presentation of the DLL functionalities with usage samples

Index

1. Introduction

This document intends to present the DLL (*Dynamic Link Library*) created by Visione for an integration with the Sharepoint: ***Visione.Sharepoint.Manager.dll***

This DLL can be used, for example in the Enterprise management system Dynamics NAV, allowing a simplified connection to the enterprise Sharepoint portal.

The DLL functionalities use the WebServices available in the Sharepoint, turning the connection of any application with Sharepoint more easily.

To be possible to access to the DLL functionalities you need, first of all, to have a Sharepoint portal and to have an administrator login to it.

In the annex, you will find some lists with necessary information to certain nodes of the XML needed for the DLL functionalities.

1.1. Sharepoint

Sharepoint is a platform, developed by Microsoft, which can be used like a web portal for intranet or internet applications.

Sharepoint creates sites and subsites, forming a hierarchical structure, allowing to have collaborative resources like forums, blogs, wikis, applications, shared documents, document versioning, action flowcharts, access permissions, shared calendar, search, resources management, discussions, lists, etc.

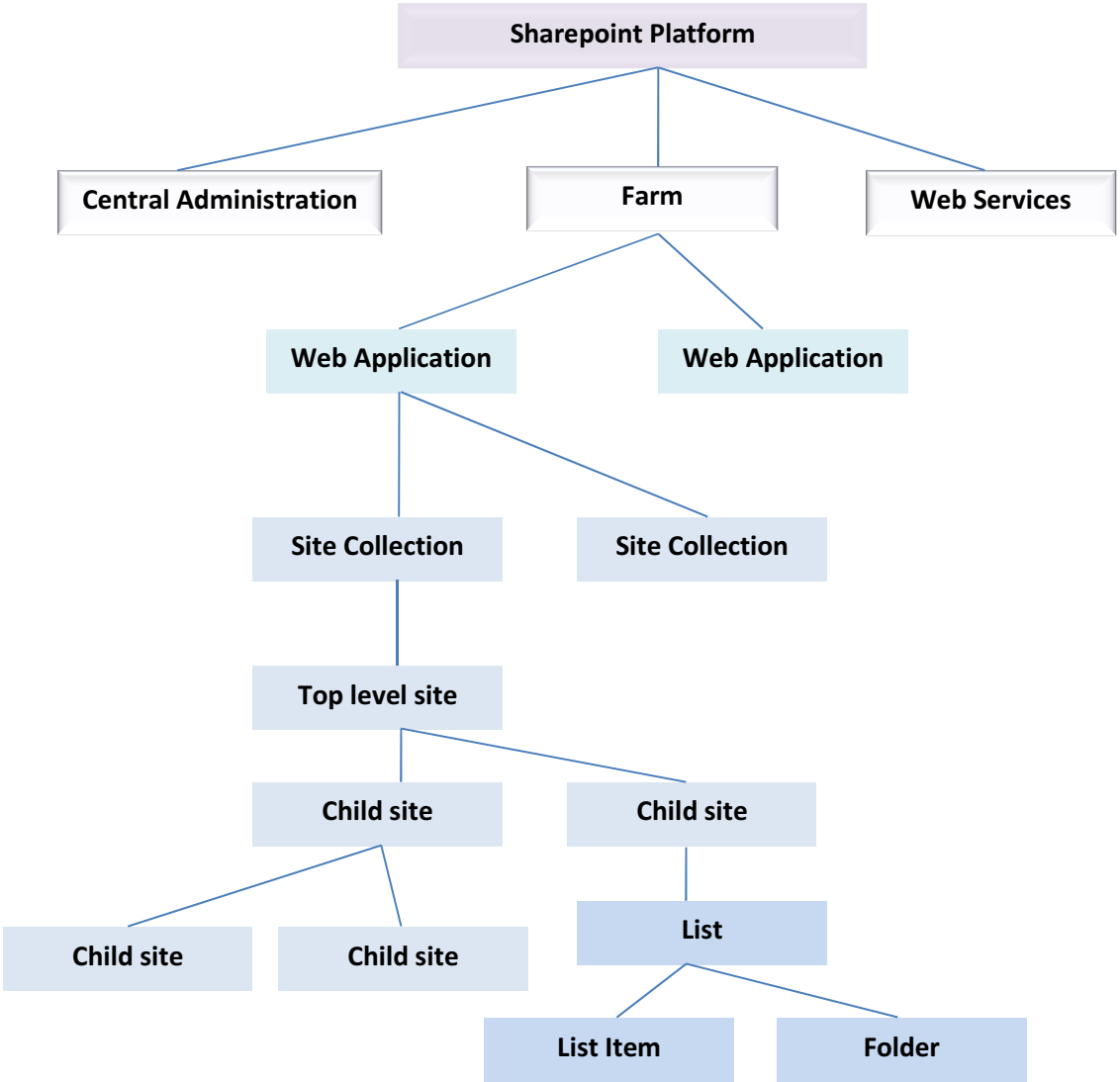
The Sharepoint platform includes 3 applications:

- Sharepoint Central Administration - management of the Sharepoint platform, available only to sharepoint administrators, available on a specific port different then the port for the sharepoint site collection.
- Sharepoint Web Services - collection of Sharepoint Web Services.
- Sharepoint Farm - collection of sharepoint sites available to the users, by default at the port 80.

The accesses to the Sharepoint platform are made through the browser, using the URL like in the following examples:

- http://111.111.111.111:11111 - Sharepoint central administration;
- http://111.111.111.111 - root Sharepoint site Collection;
- http://111.111.111.111/_vti_bin/Webs.asmx - Sharepoint webService Webs;
- http:// 111.111.111.111/Site2 - subsite in the root Sharepoint Site Collection;
- http:// 111.111.111.111/sites/TestSite1 - Sharepoint site Collection;
- http:// 111.111.111.111/sites/TestSite1/Site2 - subsite in a Sharepoint Site Collection.

The following image demonstrates a simple architecture of the Sharepoint platform.



The most used Sharepoint products, at the moment, are:

- SharePoint 2007
- SharePoint 2010

- Sharepoint Online

Following are presented the existing versions of the 3 Sharepoint products.

Sharepoint 2007

In Sharepoint 2007 there are 2 versions:

- WSS 3.0 - Windows SharePoint Services

  The free version of Sharepoint for who has a Windows Server 2003 or Windows Server 2008 license.

- MOSS 2007 - Microsoft Office SharePoint Server

  The version that needs a license. It's based on WSS 3, but with much more functionalities like, for example the enterprise portal and personalized templates.

  This version is divided in two:

  - MOSS 2007 Standard;
  - MOSS 2007 Enterprise – the complete version that includes, for example an improved information search, personalized forms development and Business Intelligence from MS Excel 2007.

Sharepoint 2010

In Sharepoint 2010 there are 3 versions:

- Microsoft SharePoint Foundation 2010

  It is equivalent to the WSS 3 version from Sharepoint 2007.

  It's the free version of Sharepoint for who have a Windows Server 2008 license.

  Has for requirements:

  - Windows Server 2008,
  - SQL Server 2008,
  - IIS,
  - .NET 3.5.

- Microsoft SharePoint Server 2010 (Standard)

  Equivalent to the MOSS 2007 Standard version.

  It's the version that requires a license and that, besides the functionalities of the Sharepoint Foundation version, also includes more management functionalities

and allows a bigger number of users. It allows also the publication of internet portals.

- Microsoft SharePoint Enterprise 2010

  Equivalent to the MOSS 2007 Enterprise version.

  It's the complete version that includes, for example an improved information search and integration with Business Intelligence from MS Excel 2010.

Sharepoint Online

Sharepoint Online is the Online service for the Cloud, integrated in the Office 365 product from Microsoft.

The Sharepoint Online is based on the SharePoint 2010 (more likely to the Standard version).

The requirement to have access to Sharepoint Online is to have a subscription plan of Microsoft Office 365.

2. DLL for the Sharepoint Manager

In this section it's presented the DLL for the Sharepoint Manager.

It will be done a presentation of the functionalities available in the DLL and some samples will be presented.

2.1. Register the DLL

To start using the DLL it's necessary to register the DLL in the Windows Operating System where it will be used.

The registration can be made through the installation that is available in the folder that contains the DLL.

The installation can be done through one of the following three ways:

- Executing the Installation file "setup.exe";
- Executing the Command file "setup.cmd";
- Initializing the Command Prompt "cmd.exe" and executing the following commands:
    - Go to the folder that holds the DLL, through the command:
      > cd <<folder location>>
    - Generate the TLB (COM Type Library) for the dll, using the command:
      > regasm Visione.Sharepoint.Manager.dll /tlb:Visione.Sharepoint.Manager.tlb
    - Register the DLL in the GAC (Global Assembly Chache), with the command:
      > gacutil /i Visione.Sharepoint.Manager.dll

All the options need administrator permission.

2.2. Trial Version

There is a Trial version available for 30 days, so that you can try and test the DLL. After the end of that time you will receive the following error message whenever you call the DLL functionalities:

> *"Your trial version of Visione Sharepoint Manager DLL has expired. If you wish to continue using the Visione Sharepoint Manager DLL, please buy a full version. Consult conditions on http://www.visione.pt/en/index.html"*

To continue using the DLL you should buy the full version of the DLL. You can consult the conditions on:

http://www.visione.pt/en/index.html

2.3. DLL functionalities

The DLL is divided in 2 classes, both with the same functionalities:

- SharepointManagerFromXML - holds the functionalities, having with parameter the XML object;
- SharepointManagerFromFile - holds the functionalities, having with parameter the XML file location.

The necessary information for each functionality is based in XML (*Extensible Markup Language*), having the chance of the functionality to receive as parameter the XML object itself or the location for the file that holds the XML to be read by the functionality.

Each functionality returns an object of the *string* type. The returned value can be:

- In case of success: “Success”;
- In case of error: description of the error.

In the case of the functionalities *GetLastInsertedTaskItemID*, *GetDocLibraryItems* and *GetTaskListItems* the inputs and outputs are different:

- The functionalities *GetDocLibraryItems* and *GetTaskListItems* receive has input the XML object or the XML file location (depending of the class where the functionality is) and has output a XML object;
- The functionality *GetLastInsertedTaskItemID* receives has input an object of the *integer* type and has output an object of the *string* type.

The tags of the XML for the functionalities are case sensitive. For example, if a functionality is expecting that the XML contains a tag “<ip>”, but instead of it, the XML contains a tag “<IP>”, the functionality will return error.

For the authentication in Sharepoint platform, it depends on the Sharepoint version:

- Sharepoint 2007 and 2010 - set the tag “<sharepointOnline>” to “False” and enter your credentials, filling the authentication tags "<login>", "<pass>" and "<domain>";
- Sharepoint Online - set the tag “<sharepointOnline>” to “True”. The DLL will launch the Sharepoint authentication window, so you can enter your credentials. It is not necessary to fill the authentication tags: "<login>", "<pass>", "<domain>".

The functionalities available in each class are:

SharepointManagerFromXML Class

string CreateSiteCollection(DOMDocument60 xmlObj)
string DeleteSiteCollection(DOMDocument60 xmlObj)
string CreateSite(DOMDocument60 xmlObj)
string DeleteSite(DOMDocument60 xmlObj)
string CreateSitePermissions(DOMDocument60 xmlObj)
string CreateLibrary(DOMDocument60 xmlObj)
string CreateDocLibFolder(DOMDocument60 xmlObj)
string CopyFileToSharepoint(DOMDocument60 xmlObj)
string CopyFileFromSharepoint(DOMDocument60 xmlObj)
string DocumentCheckInOut(DOMDocument60 xmlObj)
string OpenFile(DOMDocument60 xmlObj)
string DeleteFile(DOMDocument60 xmlObj)
string InsertTaskItem(DOMDocument60 xmlObj)
string UpdateLibraryItem(DOMDocument60 xmlObj)
string GetLastInsertedTaskItemID(int itemIndex)
DOMDocument60 GetDocLibraryItems(DOMDocument60 xmlObj)
DOMDocument60 GetTaskListItems(DOMDocument60 xmlObj)
string ApproveDocLibraryItem(DOMDocument60 xmlObj)
string InsertTaskItemAttachment(DOMDocument60 xmlObj)
string DeleteTaskItemAttachment(DOMDocument60 xmlObj)

SharepointManagerFromFile Class

string CreateSiteCollection(string filePath)
string DeleteSiteCollection(string filePath)
string CreateSite(string filePath)
string DeleteSite(string filePath)
string CreateSitePermissions(string filePath)
string CreateLibrary(string filePath)
string CreateDocLibFolder(string filePath)
string CopyFileToSharepoint(string filePath)
string CopyFileFromSharepoint(string filePath)
string DocumentCheckInOut(string filePath)
string OpenFile(string filePath)
string DeleteFile(string filePath)
string InsertTaskItem(string filePath)
string UpdateLibraryItem(string filePath)
string GetLastInsertedTaskItemID(int itemIndex)
DOMDocument60 GetDocLibraryItems(string filePath)
DOMDocument60 GetTaskListItems(string filePath)
string ApproveDocLibraryItem(string filePath)
string InsertTaskItemAttachment(string filePath)
string DeleteTaskItemAttachment(string filePath)

Next the functionalities will be presented.

CreateSiteCollection

Functionality to create a site (i.e., site collection).

This functionality is not available for the Sharepoint Online version (Office 365).

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site will be located;
- *port* – port of the *central administration* of the sharepoint application;
- *ipPort* – port of the ip address of the sharepoint application root where the site will be located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the new site inside the root of the sharepoint application;
- *title* – title for the new site;
- *description* – description of the new site;
- *template* – template of the new site. See in the annex the list of codes available for the template;
- *language* – language of the new site. See in the annex the list of codes available for the language;
- *login* – login of the site administrator, that has permissions to access the sharepoint *central administration*;
- *pass* – password of the administrator login;
- *domain* – domain of the administrator login;
- *email* – e-mail of the site administrator.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <port>11111</port>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <title>Test Site 1</title>
  <description>Site for testing</description>
  <template>STS#0</template>
  <language>1033</language>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <email>email@domain.pt</email>
</sharepoint>
```

DeleteSiteCollection

Functionality to delete a site (i.e., site collection).

This functionality is not available for the Sharepoint Online version (Office 365).

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *port* – port of the *central administration* of the sharepoint application;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site to delete inside the root of the sharepoint application;
- *login* – login of the site administrator, that has permissions to access the sharepoint *central administration*;
- *pass* – password of the administrator login;
- *domain* – domain of the administrator login.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <port>11111</port>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
</sharepoint>
```

CreateSite

Functionality to create a subsite (i.e., site).

When you create a subsite, the permissions of the site collection, where the subsite will be located, will be copied for the subsite.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the subsite will be located;
- *ipPort* – port of the ip address of the sharepoint application root where the subsite will be located. If it's not defined, it assumes the default port (80);
- *collection* – name or address of the site collection where the subsite will be located;
- *sitename* – name or address of the new subsite inside the site collection;
- *title* – title for the new subsite;
- *description* – description of the new subsite;
- *template* – template of the new subsite. See in the annex the list of codes available for the template. If the template is a custom template, insert the template name;

- *customTemplate* – template is a custom template. Set "True" in affirmative case and "False" otherwise;
- *language* – language of the new subsite. See in the annex the list of codes available for the language;
- *login* – login of the user that has permissions to create subsites;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <collection>sites/TestSite1</collection>
  <sitename>Site2</sitename>
  <title>Test Site 2</title>
  <description>Site for testing2</description>
  <template>STS#0</template>
  <customTemplate>False</customTemplate>
  <language>1033</language>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
</sharepoint>
```

DeleteSite

Functionality to delete a subsite (i.e., site).

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the subsite is located;
- *ipPort* – port of the ip address of the sharepoint application root where the subsite is located. If it's not defined, it assumes the default port (80);
- *collection* – name or address of the site collection where the subsite is located;
- *sitename* – name or address of the subsite to delete inside the site collection;
- *login* – login of the user that has permissions to delete subsites;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <collection>sites/TestSite1</collection>
```

```
  <sitename>Site2</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
</sharepoint>
```

CreateSitePermissions

Functionality to create permissions for the users of the site.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to create permissions;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *Permissions* – is a "father" node that contains the "child" node with the permissions for the user:
  - *Users* – is a "child" node that contains the node *User* with the permissions for the user. The node *User* has the following attributes:
    - *LoginName* – login of the user, with the domain that he belongs;
    - *Email* – e-mail of the user;
    - *Name* – name of the user;
    - *Notes* – notes about the permission that will be added to the user;
    - *PermissionMask* – permission mask for the user. See in the annex the list of codes available for the permission mask.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login></login>
  <pass></pass>
  <domain></domain>
  <sharepointOnline>True</sharepointOnline>
  <Permissions>
   <Users>
    <User LoginName=
     "domain\user1"
     Email="user1@domain.pt"
     Name="User1"
     Notes=""
```

```
      PermissionMask="-1" />
    </Users>
  </Permissions>
  <Permissions>
    <Users>
      <User LoginName=
        "domain\user2"
        Email="user2@domain.pt"
        Name="User2"
        Notes=""
        PermissionMask="138608641" />
    </Users>
  </Permissions>
</sharepoint>
```

CreateLibrary

Functionality to create sharepoint libraries.

When you create a sharepoint library it will be created a link for the library in the *quick launch menu*. The sharepoint library created has the file versioning activated, for the versions' control of the documents.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to create libraries;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *library* – is the "father" node that contains the "child" nodes with the characteristics of the library to create:
  - *libname* – name of the new library;
  - *libtitle* – title or description of the library;
  - *libtemplate* – template of the library. See in the annex the list of codes available for the library template;
  - *newfield* – is the "father" node that contains the "child" nodes with the characteristics of the new field to create in the library:
    - *type* – type of the field in the library. See in the annex the list of field types available for libraries;
    - *displayname* – name of the field to be displayed in the library. It is not the internal name of the field. The internal name is created automatically based on the display name;
    - *description* – description of the field.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <library>
    <libname>Docs1</libname>
    <libtitle>My Document Library 1</libtitle>
    <libtemplate>101</libtemplate>
    <newfield>
      <type>Text</type>
      <displayname>New Field 1</displayname>
      <description>Description 1</description>
    </newfield>
  </library>
  <library>
    <libname>Docs2</libname>
    <libtitle>My Document Library 2</libtitle>
    <libtemplate>101</libtemplate>
  </library>
</sharepoint>
```

CreateDocLibFolder

Functionality to create a folder inside a library.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to create folders;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *doclibraryfolder* – is the "father" node that contains the "child" nodes with the characteristics of the folder to create:
  - *doclibname* – name of the sharepoint library that will receive the new folder;
  - *doclibfolder* – name of the folder to create inside the sharepoint library.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
```

```
<pass>password</pass>
<domain>domain</domain>
<sharepointOnline>False</sharepointOnline>
<doclibraryfolder>
  <doclibname>Docs1</doclibname>
  <doclibfolder>Folder2</doclibfolder>
</doclibraryfolder>
<doclibraryfolder>
  <doclibname>Docs2</doclibname>
  <doclibfolder>Folder3</doclibfolder>
</doclibraryfolder>
<doclibraryfolder>
  <doclibname>Docs2</doclibname>
  <doclibfolder>Folder3/Folder4</doclibfolder>
</doclibraryfolder>
<doclibraryfolder>
  <doclibname>Docs2</doclibname>
  <doclibfolder>Folder3/Folder4/Folder5</doclibfolder>
</doclibraryfolder>
</sharepoint>
```

CopyFileToSharepoint

Functionality to copy files for a sharepoint library or folder.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to copy files;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *newfile* – is the "father" node that contains the "child" nodes with the characteristics of the file to copy:
  - *source* – location and name of the file to be copied;
  - *filename* – name of the file at the destination (Sharepoint), including the file extension;
  - *libraryDisplayName* – display name of the sharepoint library or folder where the file will be located;
  - *libraryInternalName* – internal name of the sharepoint library or folder where the file will be located;
  - *newfield* – is the "father" node that contains the "child" nodes with the fields of the file to be copied:
    - *fieldType* – type of the field in the library. See in the annex the list of field types available for libraries;
    - *fieldDisplayName* – display name of the field in the library;
    - *fieldInternalName* – internal name of the field in the library;

- *fieldValue* – value of the field.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <newfile>
    <source>C: \test.pdf</source>
    <filename>test.pdf</filename>
    <libraryDisplayName>Docs1/Folder2</libraryDisplayName>
    <libraryInternalName>Docs1/Folder2</libraryInternalName>
    <newfield>
      <fieldType>Text</fieldType>
      <fieldDisplayName>Title</fieldDisplayName>
      <fieldInternalName>Title</fieldInternalName>
      <fieldValue>My file pdf</fieldValue>
    </newfield>
  </newfile>
  <newfile>
    <source>C: \test.xlsx</source>
    <filename>test.xlsx</filename>
    <libraryDisplayName>Docs 2</libraryDisplayName>
    <libraryInternalName>Docs2</libraryInternalName>
    <newfield>
      <fieldType>Text</fieldType>
      <fieldDisplayName>Title</fieldDisplayName>
      <fieldInternalName>Title</fieldInternalName>
      <fieldValue>My file xlsx</fieldValue>
    </newfield>
    <newfield>
      <fieldType>Text</fieldType>
      <fieldDisplayName>My Field</fieldDisplayName>
      <fieldInternalName>MyField</fieldInternalName>
      <fieldValue>My field Value</fieldValue>
    </newfield>
  </newfile>
</sharepoint>
```

CopyFileFromSharepoint

Functionality to copy files from the sharepoint to the local machine.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to copy files;
- *pass* – password of the user login;
- *domain* – domain of the user login;

- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *newfile* – is the "father" node that contains the "child" nodes with the characteristics of the file to copy:
  - *destination* – location and name of the file at the destiny (local machine);
  - *filename* – name of the file in the source (Sharepoint), including the file extension;
  - *filelocation* – name of the sharepoint library or folder where the file is located.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <newfile>
    <destination>C:\teste.pdf</destination>
    <filename>test.pdf</filename>
    <filelocation>Docs1/Folder2</filelocation>
  </newfile>
  <newfile>
    <destination>C:\teste.xlsx</destination>
    <filename>test.xlsx</filename>
    <filelocation>Docs2</filelocation>
  </newfile>
</sharepoint>
```

DocumentCheckInOut

Functionality to make the CheckIn or CheckOut of files in the sharepoint.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to make the CheckIn or CheckOut;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *fileCheckInOut* – is the "father" node that contains the "child" nodes with the characteristics of the checkIn / CheckOut:
  - *doclibname* – name of the sharepoint library or folder where the file is located;

- *docfilename* – name of the file, including the extension;
- *checkin* – distinction of the operation: CheckIn (set "True") or CheckOut (set "False");
- *checkincomment* – comment to the CheckIn. To fill only in case of CheckIn;
- *checkintype* – type of the file CheckIn. To fill only in case of CheckIn. See in the annex the list of codes available for the file CheckIn;
- *checkouttolocal* – checkOut of the file to the local machine (offline edition): set "True" in affirmative case and "False" otherwise. To fill only in case of CheckOut.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <fileCheckInOut>
    <doclibname>Docs1/Folder2</doclibname>
    <docfilename>test.pdf</docfilename>
    <checkin>False</checkin>
    <checkincomment></checkincomment>
    <checkintype></checkintype>
    <checkouttolocal>True</checkouttolocal>
  </fileCheckInOut>
  <fileCheckInOut>
    <doclibname>Docs1/Folder2</doclibname>
    <docfilename>test.pdf</docfilename>
    <checkin>True</checkin>
    <checkincomment>Comment to check in</checkincomment>
    <checkintype>1</checkintype>
    <checkouttolocal></checkouttolocal>
  </fileCheckInOut>
</sharepoint>
```

OpenFile

Functionality to open a Sharepoint file.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *filelocation* – name of the sharepoint library or folder where the file is located;
- *filename* – name of the file, including the extension.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sharepointOnline>True</sharepointOnline>
  <sitename>sites/TestSite1</sitename>
  <filelocation>Docs1/Folder2</filelocation>
  <filename>test.pdf</filename>
</sharepoint>
```

DeleteFile

Functionality to delete a Sharepoint file.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *filelocation* – name of the sharepoint library or folder where the file is located;
- *filename* – name of the file, including the extension;
- *login* – login of the user that has permissions to delete files;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <filelocation>Docs2</filelocation>
  <filename>test.pdf</filename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
</sharepoint>
```

InsertTaskItem

Functionality to insert items in a sharepoint task list.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;

- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to insert items in a list;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *taskitem* – is the "father" node that contains the "child" nodes with the characteristics of the item to insert:
    - *libname* – name of the library where the item will be inserted;
    - *taskfield* – is the "father" node that contains the "child" nodes with the fields of the item to insert:
        - *fieldname* – internal name of the field. This is not the display name of the field. You have to be sure of this name, so that the value is inserted in the right field;
        - *fieldvalue* – value of the field.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <taskitem>
    <libname>Tasks1</libname>
    <taskfield>
      <fieldname>Title</fieldname>
      <fieldvalue>My Title1</fieldvalue>
    </taskfield>
    <taskfield>
      <fieldname>Body</fieldname>
      <fieldvalue>My Description</fieldvalue>
    </taskfield>
  </taskitem>
  <taskitem>
    <libname>Tasks1</libname>
    <taskfield>
      <fieldname>Title</fieldname>
      <fieldvalue>My Title2</fieldvalue>
    </taskfield>
  </taskitem>
</sharepoint>
```

UpdateLibraryItem

Functionality to update items in a sharepoint list.

It can be used, for example, to update the *Status* of a task, or the name of a document.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to update items in a list;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *libraryitem* – is the "father" node that contains the "child" nodes with the characteristics of the item to update:
  - *libname* – name of the library where the item will be updated;
  - *itemID* – ID of the item to be updated;
  - *libraryfield* – is the "father" node that contains the "child" nodes with the fields of the item to update:
    - *fieldname* – internal name of the field. This is not the display name of the field. You have to be sure of this name, so that the value is updated in the right field;
    - *fieldvalue* – value of the field.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login></login>
  <pass></pass>
  <domain></domain>
  <sharepointOnline>True</sharepointOnline>
  <libraryitem>
    <libname>Tasks1</libname>
    <itemID>1</itemID>
    <libraryfield>
      <fieldname>Title</fieldname>
      <fieldvalue>My Title2</fieldvalue>
    </libraryfield>
  </libraryitem>
</sharepoint>
```

GetLastInsertedTaskItemID

Functionality to get the Tasks Item ID of a task inserted previously, if the reference to the DLL was not closed.

This functionality searches the ID of an Item in an object stored locally by the functionality to insert Task items. This object only stores the last inserted task items. When you close the

reference to the DLL, you will lose this information stored. It's recommended to use this functionality right after inserting the task items.

This functionality receives as input the index of the desired item in the list of items inserted in the functionality to insert Task items.

The output of this functionality is the ID of the item in the Tasks List.

For a better understanding of the functionality here is an example:

Suppose that you call the functionality to insert Task items to insert 3 tasks:

- Task1, Task2 and Task3.

The functionality inserts the tasks with Success and stores their IDs:

- Task1 has ID = 3;
- Task2 has ID = 4;
- Task3 has ID = 5.

Calling the functionality

- GetLastInsertedTaskItemID(2)

will retrieve the ID of the second task inserted: ID = 4.

GetDocLibraryItems

Functionality to get a list of items (folders and documents) inside a Document Library.

The output for this functionality is a XML with the list of items inside the Document Library, or document library folder.

The XML for the input of this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to get the items;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *libraryName* – name of the document library to get the items.

The XML for the input of this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
```

```
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <libraryName>Docs1/Folder2</libraryName>
</sharepoint>
```

The XML for the output of this functionality has the following nodes:

- *SuccessStatus* – Success status of the functionality operation. Can be "True" or "False";
- *ErrorMsg* – Error message in case an error occurs during the functionality operation;
- *Item* – is the "father" node that contains the "child" nodes with the characteristics of the item:
  - *ID* – ID of the Item;
  - *Type* – Type of Item. Can be "Document" or "Folder";
  - *LibraryDir* – Item Directory inside the Sharepoint application;
  - *URL* – Complete URL of the Item;
  - *Name* – Name of the Item;
  - *IsCheckedOut* – CheckedOut status of the Item. Can be "True" or "False".

The XML for the output of this functionality will be, for example:

```
<?xml version="1.0"?>
<sharepoint>
  <SuccessStatus>True</SuccessStatus>
  <ErrorMsg></ErrorMsg>
  <Item>
    <ID>1</ID>
    <Type>Document</Type>
    <LibraryDir>sites/TestSite1/Docs1/Folder2</LibraryDir>
    <URL>http://111.111.111.111/sites/TestSite1/Docs1/Folder2/test.pdf</URL>
    <Name>test.pdf</Name>
    <ISCheckedOut>True</ISCheckedOut>
  </Item>
  <Item>
    <ID>4</ID>
    <Type>Folder</Type>
    <LibraryDir>sites/TestSite1/Docs1/Folder2</LibraryDir>
    <URL>http://111.111.111.111/sites/TestSite1/Docs1/Folder2/Folder3</URL>
    <Name>Folder3</Name>
    <ISCheckedOut>False</ISCheckedOut>
  </Item>
</sharepoint>
```

GetTaskListItems

Functionality to get a list of items (tasks and summary tasks) inside a Tasks List.

The output for this functionality is a XML with the list of items inside the Task List, or Summary Task List folder.

The XML for the input of this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to get the items;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *libraryName* – name of the document library to get the items.

The XML for the input of this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <libraryName>Tasks1</libraryName>
</sharepoint>
```

The XML for the output of this functionality has the following nodes:

- *SuccessStatus* – Success status of the functionality operation. Can be "True" or "False";
- *ErrorMsg* – Error message in case an error occurs during the functionality operation;
- *Item* – is the "father" node that contains the "child" nodes with the characteristics of the item:
  - *ID* – ID of the Item;
  - *Type* – Type of Item. Can be "Task" or "Summary Task";
  - *Title* – Title of the Item;
  - *LibraryDir* – Item Directory inside the Sharepoint application;
  - *Attachments* – Number of attachments of the Item;
  - *Status* – Status of the Item;
  - *Body* – Description of the Item.

The XML for the output of this functionality will be, for example:

```
<?xml version="1.0"?>
<sharepoint>
  <SuccessStatus>True</SuccessStatus>
  <ErrorMsg></ErrorMsg>
```

```
  <Item>
    <ID>2</ID>
    <Type>Task</Type>
    <Title>My Title</Title>
    <LibraryDir>sites/TestSite1/Tasks1</LibraryDir>
    <Attachments>0</Attachments>
    <Status>Not Started</Status>
    <Body>My Description</Body>
  </Item>
  <Item>
    <ID>3</ID>
    <Type>Summary Task</Type>
    <Title>Sum Title</Title>
    <LibraryDir>sites/TestSite1/Tasks1</LibraryDir>
    <Attachments>0</Attachments>
    <Status>In Progress</Status>
    <Body>Sum Description</Body>
  </Item>
</sharepoint>
```

ApproveDocLibraryItem

Functionality to approve or reject documents, for an Approval Workflow, in a Sharepoint document library.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to approve files;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *doclibitem* – is the "father" node that contains the "child" nodes with the characteristics of the file to approve:
  - *doclibname* – name of the sharepoint library or folder where the file is located;
  - *itemname* – name of the file, including the extension;
  - *workflowname* – name of the Approval Workflow that will be updated;
  - *status* – new status for the Approval workflow. Can be "Approved" or "Rejected".

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login></login>
```

```
  <pass></pass>
  <domain></domain>
  <sharepointOnline>True</sharepointOnline>
  <doclibitem>
    <doclibname>Docs1</doclibname>
    <itemname>test.xlsx</itemname>
    <workflowname>ApprovalWFName</workflowname>
    <status>Approved</status>
  </doclibitem>
</sharepoint>
```

InsertTaskItemAttachment

Functionality to insert an attachment (file) in a Task List Item.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to add the attachment;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *taskItemAttachment* – is the "father" node that contains the "child" nodes with the characteristics of the file to attach to the task list item:
  - *libname* – name of the sharepoint task list where the item is located;
  - *source* – location and name of the file to be attached to the task list item;
  - *filename* – name of the file at the destination (Task List Item), including the file extension;
  - *itemID* – ID of the task list Item.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <taskItemAttachment>
    <libname>Tasks1</libname>
    <source>c:\test.pdf</source>
    <filename>test.pdf</filename>
    <itemID>1</itemID>
  </taskItemAttachment>
  <taskItemAttachment>
    <libname>Tasks2</libname>
    <source>c:\test.txt</source>
```

```
    <filename>test.txt</filename>
    <itemID>5</itemID>
  </taskItemAttachment>
</sharepoint>
```

DeleteTaskItemAttachment

Functionality to delete an attachment (file) from a Task List Item.

The XML for this functionality has the following nodes:

- *ip* – ip address of the sharepoint application root where the site is located;
- *ipPort* – port of the ip address of the sharepoint application root where the site is located. If it's not defined, it assumes the default port (80);
- *sitename* – name or address of the site inside the root of the sharepoint application;
- *login* – login of the user that has permissions to delete the attachment;
- *pass* – password of the user login;
- *domain* – domain of the user login;
- *sharepointOnline* – Sharepoint Online version (Office 365). Set "True" in affirmative case and "False" otherwise;
- *taskItemAttachment* – is the "father" node that contains the "child" nodes with the characteristics of the file to delete from the task list item:
  - *libname* – name of the sharepoint task list where the item is located;
  - *filename* – name of the file in the Task List Item, including the file extension;
  - *itemID* – ID of the task list Item.

The XML for this functionality will be, for example:

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
  <taskItemAttachment>
    <libname>Tasks1</libname>
    <filename>test.pdf</filename>
    <itemID>1</itemID>
  </taskItemAttachment>
  <taskItemAttachment>
    <libname>Tasks2</libname>
    <filename>test.txt</filename>
    <itemID>5</itemID>
  </taskItemAttachment>
</sharepoint>
```

2.4. Examples

Apart from the examples that were being presented as the functionalities were explained, in this section will be presented more samples of the usage of the DLL.

Access to the DLL in a .NET project

In first place the DLL should be added to the project.

The following code shows how to use the DLL to copy a Sharepoint file, calling the file "c:\File.xml" that have the required xml, created in advance.

```
SharepointManager.SharepointManagerFromFile mySharepointTest = new
        SharepointManager.SharepointManagerFromFile();

string result = mySharepointTest.CopyFileFromSharepoint("c:\File.xml");
```

Access to the DLL in a Dynamics NAV application

Create a variable

- with the name "MySharepointTest"
- of the type "Automation"
- with the subtype "'Visione_Sharepoint_Manager'.SharepointManagerFromXML"

The following code shows how to use the DLL to create a Sharepoint site, calling the variable "MyXML", of the type "Automation", with the subtype "'Microsoft XML, v6.0'.DOMDocument60" loaded in advance with the required xml, and placing the result in the variable "result" which is of the type "Text".

```
CLEAR(MySharepointTest);

IF ISCLEAR(MySharepointTest) THEN

    CREATE(MySharepointTest);

result:=MySharepointTest.CreateSiteCollection(MyXML);

CLEAR(MySharepointTest);
```

Example of the XML to make the CheckIn of a file

The following XML is an example of the XML to make the CheckIn of a file, required to the CheckIn functionality.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
```

```
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <fileCheckInOut>
    <doclibname>Docs1/Folder2</doclibname>
    <docfilename>test.pdf</docfilename>
    <checkin>True</checkin>
    <checkincomment>Comment to check in</checkincomment>
    <checkintype>1</checkintype>
    <checkouttolocal></checkouttolocal>
  </fileCheckInOut>
</sharepoint>
```

Example of the XML to make the CheckOut of a file

The following XML is an example of the XML to make the CheckOut of a file, required to the CheckOut functionality.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <fileCheckInOut>
    <doclibname>Docs1/Folder2</doclibname>
    <docfilename>test.pdf</docfilename>
    <checkin>False</checkin>
    <checkincomment></checkincomment>
    <checkintype></checkintype>
    <checkouttolocal>True</checkouttolocal>
  </fileCheckInOut>
</sharepoint>
```

Example of the XML to create a subsite based on a Custom Template

The following XML is an example of the XML to create a subsite based on a Custom Template named “ABC”.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <collection>sites/TestSite1</collection>
  <sitename>Site2</sitename>
  <title>Test Site 2</title>
  <description>Site for testing2</description>
  <template>ABC</template>
  <customTemplate>True</customTemplate>
  <language>1033</language>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
```

```
</sharepoint>
```

Example of the XML to delete a file from the Sharepoint site located in a port different then the port 80

The following XML is an example of the XML to delete a file located in a Sharepoint site (version not Online) that is located in a port different then the port 80, required to the DeleteFile functionality.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort>22222</ipPort>
  <sitename>sites/TestSite1</sitename>
  <filelocation>Docs2</filelocation>
  <filename>test.pdf</filename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <sharepointOnline>False</sharepointOnline>
</sharepoint>
```

Example of the XML to delete a file from the Sharepoint site version Online

The following XML is an example of the XML to delete a file located in a Sharepoint site (Online version), required to the DeleteFile functionality.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1</sitename>
  <filelocation>Docs2</filelocation>
  <filename>test.pdf</filename>
  <login></login>
  <pass></pass>
  <domain></domain>
  <sharepointOnline>True</sharepointOnline>
</sharepoint>
```

Example of the XML to create a document library in the subsite

The following XML is an example of the XML to create a document library in the subsite, required to the functionality for creating libraries.

```
<?xml version="1.0" encoding="ISO8859-1" ?>
<sharepoint>
  <ip>111.111.111.111</ip>
  <ipPort></ipPort>
  <sitename>sites/TestSite1/Site2</sitename>
  <login>login</login>
  <pass>password</pass>
  <domain>domain</domain>
  <library>
    <libname>Docs1</libname>
```

```
    <libtitle>My Document Library 1</libtitle>
    <libtemplate>101</libtemplate>
  </library>
</sharepoint>
```

3. Contact and Support

You can contact Visione at:

Web: http://www.visione.pt/en/index.html

E-mail: visione@visione.pt

Phone: +351 22 616 3941

Mobile Phone: +351 938 131 000

Fax: +351 22 610 9932

Annex

Annex Index

Codes for the Permission Mask

List of codes for the Permission Mask when creating permissions:

Permission	Permission Mask
Administrator	"-1"
Reader	"138608641"
Web Designer	"1029638927"
Contributor	"1027801615"
Guest	"134283264"

Codes for the site template

List of codes for the Sharepoint site template:

Template	Code
Team Site	STS#0
Blank Site	STS#1
Document Workspace	STS#2
Basic Meeting Workspace	MPS#0
Blank Meeting Workspace	MPS#1
Decision Meeting Workspace	MPS#2
Social Meeting Workspace	MPS#3
Multipage Meeting Workspace	MPS#4
Business Activity Services Team Site	BAS#0
SharePoint Portal Server Site	SPS#0
SharePoint Portal Server Personal Space	SPSPERS#0
SharePoint Portal Server My Site	SPSMSITE#0
Contents area Template	SPSTOC#0
Topic area template	SPSTOPIC#0
News area template	SPSNEWS#0
News Home area template	SPSNHOME#0
Site Directory area template	SPSSITES#0
SharePoint Portal Server BucketWeb Template	SPSBWEB#0
Community area template	SPSCOMMU#0

Codes for the site language

List of codes for the Sharepoint site language:

Language	Code
Afrikaans	1078
Albanian	1052
Arabic	1025
Armenian	1067
Assamese	1101
Azeri (Cyrillic)	2092
Azeri (Latin)	1068
Basque	1069
Belarusian	1059
Bengali	1093
Bulgarian	1026
Catalan	1027
Chinese (Simplified)	2052
Chinese (Traditional)	1028
Croatian	1050
Czech	1029
Danish	1030
Dutch	1043
English (Australian)	3081
English (Canadian)	4105
English (U.K.)	2057
English (U.S.)	1033
Estonian	1061
Faeroese	1080
Farsi	1065
Finnish	1035
French	1036
French (Canadian)	3084
Frisian	1122
Georgian	1079
German	1031
German (Austrian)	3079
German (Swiss)	2055
Greek	1032
Gujarati	1095
Hebrew	1037
Hindi	1081
Hungarian	1038
Icelandic	1039
Indonesian	1057
Italian	1040
Japanese	1041
Kannada	1099

Kashmiri	1120
Kazakh	1087
Konkani	1111
Korean	1042
Latvian	1062
Lithuanian	1063
Lithuanian (Classic)	2087
Macedonian	1071
Malay	1086
Malayalam	1100
Manipuri	1112
Marathi	1102
Nepali	1121
Norwegian Bokmal	1044
Norwegian Nynorsk	2068
Oriya	1096
Polish	1045
Portuguese (Brazil)	1046
Portuguese (Portugal)	2070
Punjabi	1094
Romanian	1048
Russian	1049
Sanskrit	1103
Serbian (Cyrillic)	3098
Serbian (Latin)	2074
Sindhi	1113
Slovak	1051
Slovenian	1060
Spanish	3082
Swahili	1089
Swedish	1053
Tamil	1097
Tatar	1092
Telugu	1098
Thai	1054
Turkish	1055
Ukrainian	1058
Urdu	1056
Uzbek (Cyrillic)	2115
Uzbek (Latin)	1091
Vietnamese	1066

Codes for the Library type

List of codes for the Sharepoint Library type:

Library type	Code
Generic list	100
Document library	101
Survey	102
Links list	103
Announcements list	104
Contacts list	105
Events list	106
Tasks list	107
Discussion board	108
Picture library	109
Data sources	110
Site template gallery	111
User Information list	112
Web Part gallery	113
List template gallery	114
XML Form library	115
Master pages gallery	116
No Code Workflows	117
Custom Workflow Process	118
Wiki Page library	119
Custom grid for a list	120
Data Connection library	130
Workflow History	140
Gantt Tasks list	150
Meeting Series list	200
Meeting Agenda list	201
Meeting Attendees list	202
Meeting Decisions list	204
Meeting Objectives list	207
Meeting text box	210
Meeting Things To Bring list	211
Meeting Workspace Pages list	212
Blog Posts list	301
Blog Comments list	302
Blog Categories list	303
Issue tracking	1100
Administrator tasks list	1200

Codes for the CheckIn of files

List of codes for the CheckIn type of the files in Sharepoint:

CheckIn type	Description	Value
MinorCheckIn	Incremented as a minor version	0
MajorCheckIn	Incremented as a major version	1
OverwriteCheckIn	Overwrite the file	2

Field Types for creating new fields in a Library

List of field types for creation of new fields when creating a Sharepoint Library:

- AllDayEvent
- Attachments
- Boolean
- Calculated
- Choice
- Computed
- ContentTypeId
- Counter
- CrossProjectLink
- Currency
- DateTime
- File
- GridChoice
- Guid
- Integer
- Lookup
- LookupMulti
- ModStat
- MultiChoice
- MultiColumn
- Note
- Number
- PageSeparator
- Recurrence
- Text
- ThreadIndex
- Threading
- URL
- User
- UserMulti
- WorkflowEventType
- WorkflowStatus