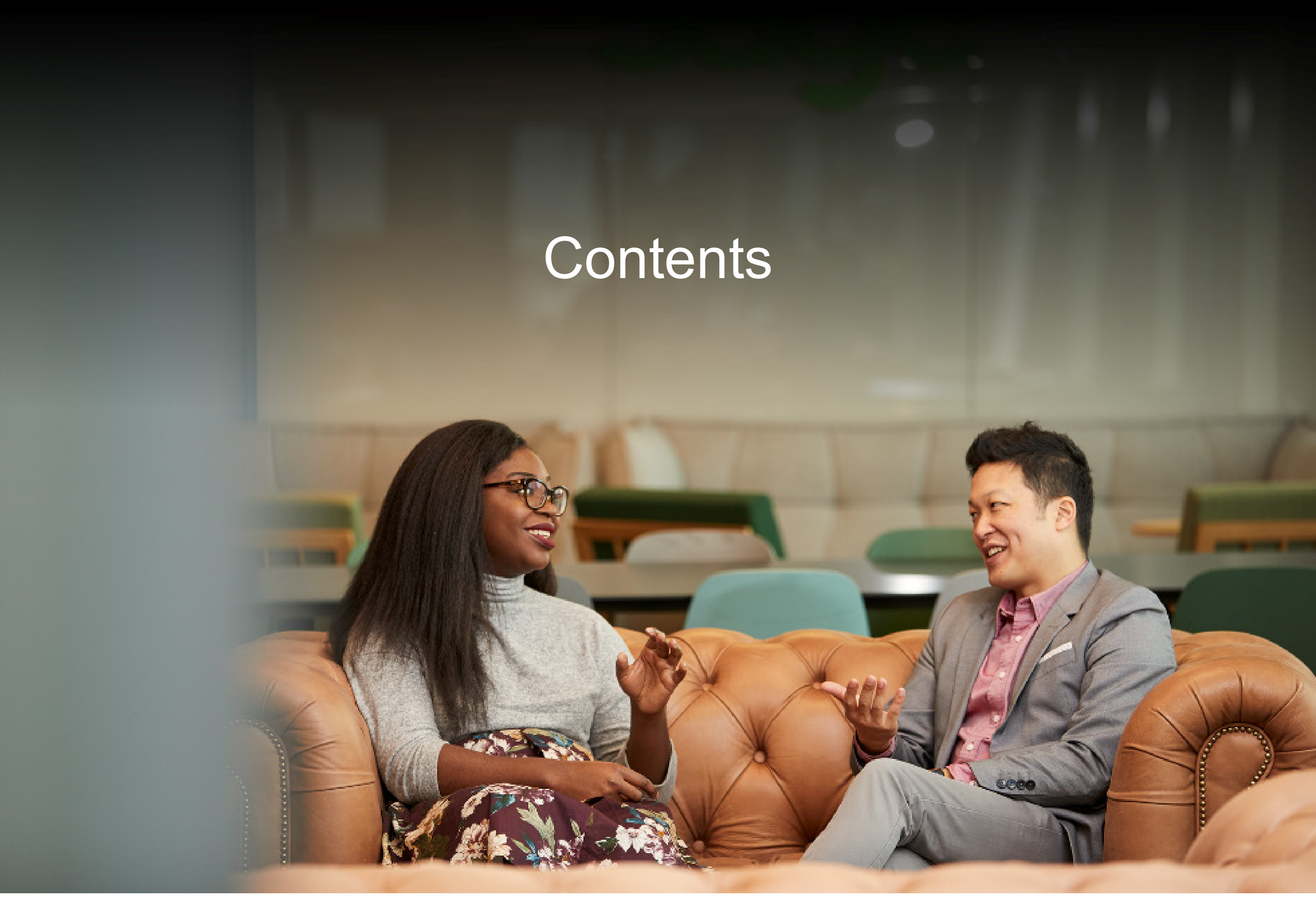
Sage XRT Common Services

Sage.fcs.client

Presentation



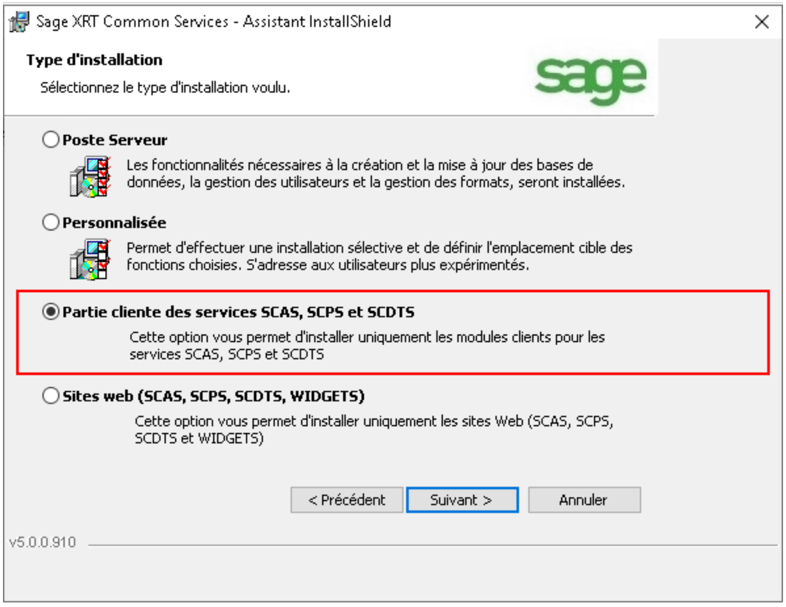
Contents

Introduction

This document presents and describes the DLL: **Sage.fcs.client**.

This *DLL* (*Dynamic Link Library*) is compatible with 32-bit and 64-bit applications. It may be used as a .NET library, or as a COM component.

The option: **SCAS, SCPS and SCDTS client part** enables its installation on the client station during **Sage XRT Common Services** installation process.



Sage XRT Common Services - Assistant InstallShield

Type d'installation

Sélectionnez le type d'installation voulu.

○ **Poste Serveur**

Les fonctionnalités nécessaires à la création et la mise à jour des bases de données, la gestion des utilisateurs et la gestion des formats, seront installées.

○ **Personnalisée**

Permet d'effectuer une installation sélective et de définir l'emplacement cible des fonctions choisies. S'adresse aux utilisateurs plus expérimentés.

● **Partie cliente des services SCAS, SCPS et SCDTS**

Cette option vous permet d'installer uniquement les modules clients pour les services SCAS, SCPS et SCDTS

○ **Sites web (SCAS, SCPS, SCDTS, WIDGETS)**

Cette option vous permet d'installer uniquement les sites Web (SCAS, SCPS, SCDTS et WIDGETS)

For its integration as a COM component in a 32-bit application, it must be saved with *RegAsm.exe* of *32-bit .NET framework*.

Usage

Sage.fcs.client enables the use of an *API Rest* set hosted by *SCASServer*, *SCDTSServer* and *SCPSServer* in **Sage XRT Common Services**. The applications including this library can simply call the methods.

The setup for *Sage.fcs.client.dll.config* enables the location of **Sage XRT Common Services** server.

Example of Configuration File:

```
<?xml version="1.0" encoding="utf-8" ?>

<configuration>

<appSettings>

<add key="securitylevel" value="http"/>

<add key="http_servicehost_SCASServer" value="http://qa-w2k12-
unipay.sagefr.adinternal.com:8760/Auth" />

<add key="https_servicehost_SCASServer" value="https://qa-w2k12-
unipay.sagefr.adinternal.com:8761/Auth" />

<add key="http_servicehost_SCDTSServer" value="http://qa-w2k12-
unipay.sagefr.adinternal.com:8740" />

<add key="https_servicehost_SCDTSServer" value="https://qa-w2k12-
unipay.sagefr.adinternal.com:8741" />

<add key="http_servicehost_SCPSServer" value="http://qa-w2k12-
unipay.sagefr.adinternal.com:8733"/>

<add key="https_servicehost_SCPSServer" value="https://qa-w2k12-
unipay.sagefr.adinternal.com:8734"/>

<add key="SerialNumber" value="5f1936dcfe0a688943322975d785efed"/> <!—Pour la récupération
des chaînes de connexion -->

</appSettings>

</configuration>
```

Integrated Methods Description

DoLogin

Overview:

```
void DoLogin(object p_oReq)
```

p_oReq represents the following object:

```
    public class LoginReq : ILoginReq

    {

        public LoginMode mode { get; set; }

        public string workgroup { get; set; }

        public string user { get; set; }

        public string password { get; set; }

        public string certificatebase64 { get; set; }

        public string challengerawbase64 { get; set; }

        public string challengesignedhex { get; set; }


    }

    public enum LoginMode

    {

        SAMLV2_AUTH,

        STRONG_AUTH,

        STANDARD_AUTH

    }
```

Description:

This method enables authentication and token generation to access the service. This token is transparently used during methods calls.

DoLoginForm

Overview:

```
void DoLoginForm(LoginProductID product, string version, string component, LanguageID
language);
```

Product represents the product ID.

```
public enum LoginProductID

{

    XCS,

    XBE,

    XBESign,

    U2,

    U2Sign,

    U2Com

}
```

Version represents the product version (eg.: 2019).

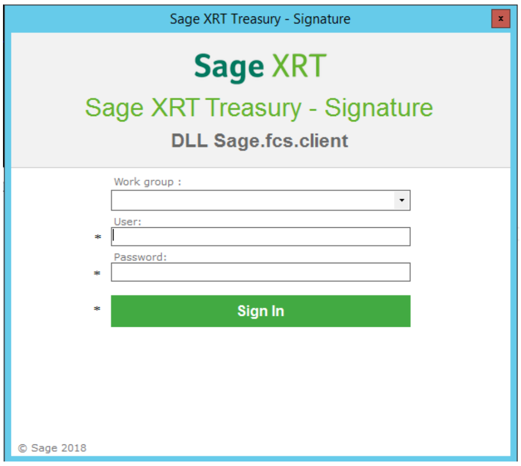
Component: name of the component to connect to (eg.: DLL Sage.fcs.client).

Language: interface language.

```
public enum LanguageID

{

    French,

    English,

    Spanish,

    German,

    Italian,

    Portuguese,

    Polish,

    Catalan

};
```

Description:

This method enables authentication and token generation to access the service through a *Win32* interface. This token is transparently used during methods calls. The method manages license checking.



WorkgroupList

Overview:

```
object WorkgroupList()
```

Description:

This method may be called without authentication, nor access token. It returns a table presenting the available workgroups list.

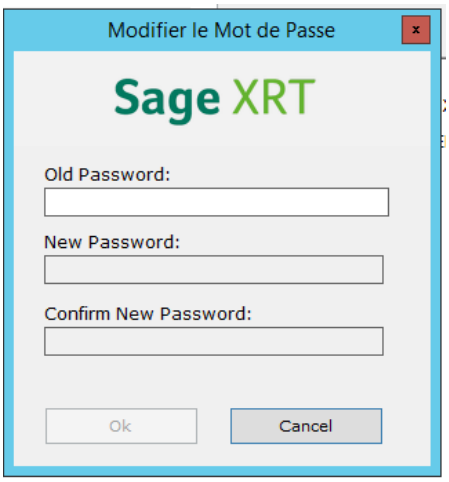
PasswordUpdatingByActionForm

Overview:

```
void PasswordUpdatingByActionForm();
```

Description:

It enables the update of the connected user’s password through the graphical interface.



PasswordUpdatingByAction

Overview:

```
void PasswordUpdatingByAction(object p_oReq);
```

p_oReq represents the following object:

```
public class PasswordReqByAction : IPasswordReqByAction

{

    public string old_password { get; set; }


    public string new_password { get; set; }


}
```

Description:

It enables the update of the connected user’s password.

ProjectList

Overview:

```
object ProjectList(object p_oReq);
```

p_oReq represents the following object:

```
public class ProjectListReq : IProjectListReq

{

    public string product { get; set; }

}
```

The answer is the following object:

```
     public class ProjectListRep : IProjectListRep

    {

     public object projects { get; set; }

    }

      public class ProjectList : IProjectList

    {

        public string code { get; set; }

        public string name { get; set; }

        public object types { get; set; }

        public int licence { get; set; }

        public ApifmtFamily family { get; set; }

        public ApifmtFormat format { get; set; }

        public int nbrwork { get; set; }

        public int productid { get; set; }

    }
```

Description:

This function enables the listing of the format library projects associated with a product.

WorkList

Overview:

```
object WorkList(object p_oReq);
```

p_oReq represents the following object:

```
     public class WorkListReq : IWorkListReq

    {

        public string product { get; set; }

        public string project { get; set; }


    }
```

©Sage 2019

The answer is the following object:

```
public class WorkListRep : IWorkListRep
{
    public object works { get; set; }
}
public class WorkList : IWorkList
{
    public string code { get; set; }
    public string name { get; set; }
    public string version { get; set; }
    public object types { get; set; }
    public object tasks { get; set; }
}
```

Description:

This function enables the listing of the format library processes associated with a product and a project.

UserList

Overview:

```
object UserList(object p_oReq);
```

p_oReq represents the following object:

```
    public class UserListReq : IUserListReq

    {

        public bool bGetProfileList { get; set; } = false; //Permet de récupérer la liste
des profiles auxquels appartient l’utilisateur

        public bool bGetSiteList { get; set; } = false; //Permet de récupérer la liste des
sites auxquels appartient l’utilisateur



    }
```

The answer is the following object:

```
    public class UserListRep : IUserListRep

    {

        public object users { get; set; }

        public object profiles { get; set; }

        public object sites { get; set; }

    }

    public class User : IUser

    {

        public string name { get; set; }

        public string description { get; set; }

        public int securitylevel { get; set; }

        public string lang { get; set; }

        public string email { get; set; }

        public int authenticationtype { get; set; }

        public bool authenticatex509 { get; set; }

        public string validtilldate { get; set; }

        public bool active { get; set; }

        public string lastactionuser { get; set; }

        public bool isvirtual { get; set; }

        public bool issecurityleveldefined { get; set; }

        public bool islanguagedefined { get; set; }
        public bool twofactorauthlevel { get; set; }

        public string tfasecretcodevalid { get; set; }

        public string lockedtilldate { get; set; }

        public string lockedtilltime { get; set; }


        public object profiles { get; set; }

        public object sites { get; set; }

                          }
```

Description:

This function enables the listing of the users associated with a token workgroup.

TrancoTableList

Overview:

```
object TranscoTableList();
```

The answer is the following object:

```
     public class TranscoTableListRep : ITranscoTableListRep

    {

        public object tables { get; set; }

    }

    public class Table : ITable

    {

        public int id { get; set; }

        public string name { get; set; }

        public string description { get; set; }

        public int input { get; set; }

        public int output { get; set; }

        public string lastactionuser { get; set; }

    }
```

Description:

This function enables the listing of the transcoding tables associated with a token workgroup.

ProfileList

Overview:

```
object ProfileList();
```

The answer is the following object:

```
 public class ProfileListRep : IProfileListRep

{

    public object profiles { get; set; }

}

public class Profile : IProfile

{

    public string name { get; set; }

    public string description { get; set; }

    public int type { get; set; }

    public bool active { get; set; }

    public string lastactionuser { get; set; }

}
```

Description:

This function enables the listing of the user profiles associated with a token workgroup.

FormatList

Overview:

```
object FormatList();
```

The answer is the following object:

```
     public class ApifmtFormatListRep : IApifmtFormatListRep

    {

        public object formats { get; set; }

    }

    public class ApifmtFormat : IApifmtFormat

    {

        public int id { get; set; }

        public string name { get; set; }

        public string desc { get; set; }

        public int licence { get; set; }

    }
```

Description:

This function enables the listing of the formats from format library.

FamilyList

Overview:

```
object FamilyList();
```

The answer is the following object:

```
     public class ApifmtFamilyListRep : IApifmtFamilyListRep

    {

        public object families { get; set; }

    }

    public class ApifmtFamily : IApifmtFamily

    {

        public int id { get; set; }

        public string name { get; set; }

        public string desc { get; set; }

    }
```

Description:

This function enables the listing of the format library categories.

ProductList

Overview:

```
object ProductList();
```

The answer is the following object:

```
 public class ApifmtProductListRep : IApifmtProductListRep

{

    public object products { get; set; }

}

public class ProductList : IProductList

{

    public int id { get; set; }

    public string code { get; set; }

    public string name { get; set; }

    public int nbrproject { get; set; }

}
```

Description:

This function enables the listing of the format library products.

FunctionRights

Overview:

```
object FunctionRights(object p_oReq);
```

p_oReq represents the following object:

```
public class FunctionRightsReq : IFunctionRightsReq

{

    public FunctionRightsLevel level { get; set; } = FunctionRightsLevel.CURRENTUSER;

    public string code { get; set; }

    public string[] functions { get; set; }

}
```

The answer is the following object:

```
public class FunctionRightsRep : IFunctionRightsRep

{

    public object functions { get; set; }

}

public class FuncRight: IFuncRight

{

    public string function { get; set; }

    public bool right { get; set; }

}
```

Description:

This function enables the listing of the rights associated with a function list for the connected user, the user specified as a parameter or the profile specified as a parameter.

PwdInfos

Overview:

```
object PwdInfos();
```

The answer is the following object:

```
public class PwdInfosRep : IPwdInfosRep

{

    public object pwd { get; set; }

}

public class DataParamsUpdatePassword : IDataParamsUpdatePassword

{

    public int bTrustedConnection { get; set; } = -1;

    public int iAccountsMinDigitsInPassword { get; set; } = -1;

    public int iAccountsMinCapLettersInPassword { get; set; } = -1;

    public int iAccountsMinPasswordLength { get; set; } = -1;

    public int iAccountsCheckPasswordsHistoryLength { get; set; } = -1;

    public int bAccountsCheckPasswordsDifferFromUserName { get; set; } = -1;

    public string sAccountsDefaultPassword { get; set; }

    public int iAccountsDefaultPasswordValidityPeriodType { get; set; } = -1;

    public int iAccountsDefaultPasswordValidityPeriodCount { get; set; } = -1;

    public int iAccountsPasswordValidityPeriodType { get; set; } = -1;
```

```
        public int iAccountsPasswordValidityPeriodCount { get; set; } = -1;

        public int iAccountsLockoutAfterAttemtsCount { get; set; } = -1;

        public int iAccountsLockoutPeriodType { get; set; } = -1;

        public int iAccountsLockoutPeriodCount { get; set; } = -1;

    }
```

Description:

This function enables the listing of the password rules for **Sage XRT Common Services** standard users.

LicenceDetails

Overview:

```
object LicenseDetails(object p_oReq);
```

p_oReq represents the following object:

```
    public class LicenseDetailsReq : ILicenseDetailsReq

    {

        public string product { get; set; }

    }
```

The answer is the following object:

```
   public class LicenseDetailsRep : ILicenseDetailsRep

    {

        public object features { get; set; }

    }

    public class Feature : IFeature

    {

        public string id { get; set; }

        public string description { get; set; }

        public string value { get; set; }

    }
```

Description:

This function enables the listing of detailed info. on the license for a given product.

LicenceList

Overview:

```
object LicenseList();
```

The answer is the following object:

```
    public class LicenseListRep : ILicenseListRep

    {

        public object licenses { get; set; }

        public object products { get; set; }

    }

    public class License : ILicense

    {

        public string product { get; set; }

        public string customer { get; set; }

        public string enddate { get; set; }

        public string key { get; set; }

    }

    public class Product : IProduct

    {

        public string product { get; set; }

        public string name { get; set; }

    }
```

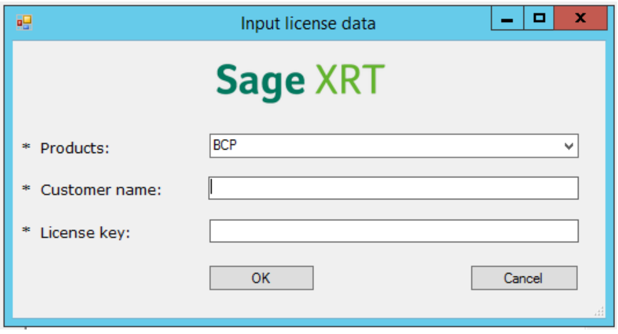
Description:

This function enables the listing of the available licenses.

LicenceCreateForm

Overview:

```
void LicenseCreateForm();
```


Input license data

Sage XRT

* Products: BCP

* Customer name:

* License key:

Description:

This function enables license creation in a *Win32* graphical interface.

ConnectionString

Overview:

```
object ConnectionString(object p_oReq);
```

p_oReq represents the following object:

```
    public class ConnectionStringReqPub : IConnectionStringReqPub

    {

        public DatabaseProductID product { get; set; }

        public DatabaseOwner type { get; set; }

    }

    public class ConnectionStringReqPub : IConnectionStringReqPub

    {

        public DatabaseProductID product { get; set; }

        public DatabaseOwner type { get; set; }

    }

    public enum DatabaseOwner

    {

        DBA,

        DBO,

        USER

    }
```

The answer is the following object:

```
public class ConnectionStringRep : IConnectionStringRep

{

    public string connectionstring { get; set; }

}
```

Description:

This function enables the retrieval of a connection string at the level of **DBA**, **DBO** or **USER**. For this method to work properly, the **SerialNumber** key of the configuration file must be specified. The connection string is actually encrypted (*RSA*) for security reasons.