

Документация разработчика по библиотеке ITProject RFID DocumentIntegration DLL

Библиотека `mobinteg.dll` используемые для обмена данными между внешней системой и программным продуктом `ITProject RFID DocumentService`.

Экспортируемые функции Библиотека описываются ниже.

1. Экспортируемые функции

1. 1. Функция Init

Инициализирует библиотеку по параметрам.

Входные параметры:

- DataSource (LPWSTR) - адрес сервера;
- Login (LPWSTR) - логин;
- Password (LPWSTR) - пароль;
- InitialCatalog (LPWSTR) - имя базы данных;
- InstanceName (LPWSTR) - имя экземпляра или NULL.

Возвращаемое функцией целое типа (целое), принимает значения:

- - если инициализация была успешной;
- -1 - если инициализация завершилась с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 2. Функция InitFromFile

Инициализирует библиотеку из файла конфигурации формата XML.

Входные параметры:

- ConfigFile (LPWSTR) - путь к файлу конфигурации.

Возвращаемое функцией целое типа (целое), принимает значения:

- - если инициализация была успешной;
- -1 - если инициализация завершилась с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 3. Функция GetTerminals

Записывает во входной буфер данные по терминалам.

Входные параметры:

- InStr (LPWSTR) - указатель на входной буфер
- InSize (INT) - размер буфера

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если функция завершилась успешно;
- > 0 - размер данных, если размер входного буфера недостаточен;
- -1 если выполнение завершилось с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 4. Функция **GetDocumentHeaders**

Записывает во входной буфер заголовки документов.

Входные параметры:

- TermCode (LPWSTR) - код терминала;
- AllFlag (INT) - флаг выгрузки (0 – только не выгруженные ранее документы, 1 – все документы);
- InStr (LPWSTR) - указатель на входной буфер;
- InSize (INT) - размер буфера.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если функция завершилась успешно;
- > 0 - размер данных, если размер входного буфера недостаточен;
- -1 если выполнение завершилось с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 5. Функция **GetDocumentData**

Записывает во входной буфер данные документа.

Входные параметры:

- DocID (INT) - номер документа на терминале;
- InStr (LPWSTR) - указатель на входной буфер;
- InSize (INT) - размер буфера.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если функция завершилась успешно;
- > 0 - размер данных, если размер входного буфера недостаточен;
- -1 если выполнение завершилось с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 6. Функция **GetDictionary**

Записывает во входной буфер записи из справочника.

Входные параметры:

- DictName (LPWSTR) - имя справочника;
- InStr (LPWSTR) - указатель на входной буфер;
- InSize (INT) - размер буфера.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если функция завершилась успешно;
- > 0 - размер данных, если размер входного буфера недостаточен;
- -1 если выполнение завершилось с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 7. Функция DeleteDictionary

Удаляет запись из справочника.

- Входные параметры:
- DictName (LPWSTR) - имя справочника;
- ID (INT) - идентификатор записи.

Возвращаемое функцией целое типа (целое), принимает значения:

- - если запись была удалена;
- -1 - если при удалении возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 8. Функция InsertDictionary

Добавляет одну или несколько записей в справочник.

- DictName (LPWSTR) - имя справочника;
- DataString (LPWSTR) - строка, содержащая данные записи.

Возвращаемое функцией целое типа (целое), принимает значения:

- > 0 - число загруженных записей в справочник;
- -1 - если при загрузке записей возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 9. Функция GetLastError

Записывает во входной буфер описание последней ошибки.

Входные параметры:

- InStr (LPWSTR) - указатель на входной буфер;
- InSize (INT) - размер буфера.

Возвращаемое функцией целое типа (целое), принимает значения:

- > 0 - размер данных, если размер входного буфера недостаточен.
- 0 – если функция завершила работу без ошибки;
- -1 – если функция завершила работу с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

1. 10. Функция ImportContract

Импортирует запись в справочник контрагентов.

Входные параметры:

- ID (INT) - код контрагента;
- ParentID (INT) - код родительской категории (или NULL);
- IsCategory (INT) - флаг категории (1 – «категория», 0 – «контрагент»);
- Name (LPWSTR) - наименование контрагента;
- TIN (LPWSTR) - ИНН (или NULL);
- RCR (LPWSTR) - КПП (или NULL).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 11. Функция ImportGoods

Импортирует запись в справочник товаров.

Входные параметры:

- ID (INT) - код товара;
- ParentID (INT) - код родительской категории (или NULL);
- IsCategory (INT) - флаг категории (1 – «категория», 0 – «товар»);
- Name (LPWSTR) - наименование товара;
- IsWeight (INT) - тип товара (1 – «весовой», 0 – «штучный»);
- IsCharacter (INT) - флаг характеристики (1 – «характеристика», 0 – «товар»);
- Article (LPWSTR) - артикул (или NULL).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 12. Функция ImportGoodsUnits

Импортирует запись в справочник единиц товаров.

Входные параметры:

- ID (INT) - код единицы товара;

- GoodID (INT) - код товара;
- Name (LPWSTR) - наименование единицы товара;
- Ratio (FLOAT) – пропорция;
- PriceBuy (FLOAT) - закупочная цена;
- PriceSell (FLOAT) - цена продажи;
- CurrID (INT) - код валюты;
- TagID (LPWSTR) - TagID метки;
- UnitID (INT) - код единицы измерения;
- BarCode (LPWSTR) - штрих - код (или NULL);
- CharSet (LPWSTR) - набор характеристик (или NULL).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 13. Функция ImportCurr

Импортирует запись в справочник валют.

Входные параметры:

- ID (INT) - код валюты;
- Name (LPWSTR) - наименование валюты;
- Ratio (FLOAT) - отношение к текущей базовой валюте;
- Code (INT) - код валюты, международный;
- IsDefault (INT) - флаг базовой валюты (1 – валюта «по умолчанию»).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 14. Функция ImportLocationt

Импортирует запись в справочник местоположений.

Входные параметры:

- ID (INT) - код местоположения;
- ParentID (INT) - код родительской категории (или NULL);
- IsCategory (INT) - флаг категории (1 – «категория», 0 – «местоположение»);

- Name (LPWSTR) - наименование местоположения;
- FuncID (INT) - код ответственного лица (или NULL).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 15. Функция ImportUnit

Импортирует запись в справочник единиц измерений.

Входные параметры:

- ID (INT) - код единицы измерения;
- Name (LPWSTR) - наименование единицы измерения.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 16. Функция ImportFunc

Импортирует запись в справочник ответственных лиц.

Входные параметры:

- ID (INT) - код ответственного лица
- Name (LPWSTR) - Ф. И. О. ответственного лица;
- Job (LPWSTR) - должность (или NULL).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 17. Функция ImportCapital

Импортирует запись в справочник основных средств.

Входные параметры:

- ID (INT) - код основного средства;
- ParentID (INT) - код родительской категории (или NULL);
- IsCategory (INT) - флаг категории (1 – «категория», 0 – «основное средство»);

- Name (LPWSTR) - наименование основного средства;
- FuncID (INT) - код ответственного лица (NULL);
- TagID (LPWSTR) - TagID метки (или NULL);
- OrgDivID (INT) - код основного средства (или NULL);
- InvNum (LPWSTR) - инвентарный номер (или NULL).

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 18. Функция ImportOrgdiv

Импортирует запись в справочник подразделений.

Входные параметры:

- ID (INT) - код подразделения;
- ParentID (INT) - код родительской категории (или NULL);
- IsCategory (INT) - флаг категории (1 – «категория», 0 – «подразделение»);
- Name (LPWSTR) - наименование подразделения.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 19. Функция ImportInvnum

Импортирует запись в справочник инвентарных номеров.

Входные параметры:

- ID (INT) - код инвентарного номера;
- Number (LPWSTR) - инвентарный номер.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 20. Функция SetCharset

Устанавливает наборы характеристик для единицы товара.

Входные параметры:

- BarCode (LPWSTR) - штрих – код;
- CharSet (LPWSTR) - набор характеристик.

Возвращаемое функцией целое типа (целое), принимает значения:

- 0 - если импорт завершен успешно;
- -1 - если при импорте записи возникла ошибка.

Описание ошибки можно получить через функцию GetLastError.

1. 21. Функция GetCharset

Получить набор характеристик для единицы товара.

Входные параметры:

- BarCode (LPWSTR) - штрих – код;
- InStr (LPWSTR) - указатель 4 - байта на буфер;
- InSize (INT) - размер буфера.

Возвращаемое функцией целое типа (целое), принимает значения:

- > 0 - размер данных, если размер входного буфера недостаточен.
- 0 – если функция завершила работу без ошибки;
- -1 – если функция завершила работу с ошибкой.

Описание ошибки можно получить через функцию GetLastError.

2. Формат файла конфигурации

Пример файла XML конфигурации который может быть использован для инициализации библиотеки приведен ниже.

```
<Settings>
  <Database>
    <DataSource>192.168.0.113</DataSource>
    <Login>sa</Login>
    <Password>123host+</Password>
    <InitialCatalog>ITProject_RFID_Journal</InitialCatalog>
    <InstanceName>MSSQLSERVER</InstanceName>
  </Database>
</Settings>
```

Описание тэгов:

- DataSource - адрес сервера;
- Login - логин;
- Password - пароль;
- InitialCatalog - имя базы данных;
- InstanceName - имя экземпляра (по умолчанию MSSQLSERVER).

3. Формат экспортируемых и импортируемых данных

Экспортируемые и импортируемые данные передаются в формате XML, представляют собой содержимое строк таблицы данных.

Общий формат структуры данных:

```
<Rows>
  <Row>
    ...
  </Row>
</Rows>
```

Между каждой парой тэгов Row заключены данные строки, где имена тэгов совпадают с именами параметров, а текст внутри тэгов хранит их значение. Тэги Rows не являются необходимыми, если строка только одна.

Так как справочники ссылаются друг на друга, их следует импортировать в строгом порядке, как они перечислены ниже.

Далее приводятся только описания тэгов строк.

3. 1. Данные по терминалам

- ID (целое) - ключ;
- TermCode (целое) - код терминала;
- Description (целое) - описание;
- Login (целое) - имя пользователя;
- App (целое) - приложение;
- IsActive (целое) - флаг состояния (0 - отключен, 10 - подключен);
- Licence (целое) - тип лицензии.

3. 2. Заголовки документов

- ID (целое) - ключ;
- TermCode (строка) - код терминала;
- BackRef (строка) - признак выгрузки на терминал (1 - «выгружен», 0 - «НЕвыгружен»);
- TypeID (целое) - тип документа (из справочника DocTypes);
- ContractorID (целое) - ID контрагента (из справочника Contractors);
- SourceLocID (целое) - ID исходного места (из справочника Locations);

- DestLocID (целое) - ID места конечного перемещения (из справочника Locations);
- Summ (число с точкой) - общая сумма документа;
- CurrID (целое) - ID валюты документа (из справочника Currencies);
- Comment (строка) - комментарий;
- OrdRef (строка) - номера заказа 1С;
- OrgDivID (целое) - ID подразделения (из справочника OrgDivisions);
- UserID (целое) - ID пользователя (из справочника Users);
- Timestamp (строка) - дата / время создания документа;
- Udt_Ts (дата и время) - время обновления.

3.3. Данные документов

- DocID (целое) - ID документа
- Num (целое) - номер записи;
- GID (целое) - ID товара (из справочника Goods);
- UnitID (целое) - ID единицы измерения (из справочника Units);
- Cnt (целое) - количество товара;
- Price (число с точкой) - цена товара;
- Summ (число с точкой) - сумма;
- Acct_Cnt (целое) - количество товара (учетное);
- Acct_Price (число с точкой) - цена товара (учетная);
- Acct_Summ (число с точкой) - сумма (учетная);
- CapID (целое) - ID основного средства (из справочника Capital);
- FactPrice (число с точкой) - цена товара (учетная);
- FactExist (целое) - наличие товара (1 - «есть», 0 - «нет»).

3.4. Записи справочника контрагентов (Contractors)

- ID (целое) - ключ;
- Desc (строка) - наименование;
- ParentID (целое) - ID родительской категории;
- IsCategory (целое) - тип категории (1 - «категория», 0 - «значение»);
- TIN (строка) – ИНН;
- RCR (строка) – КПП;
- Udt_Ts (дата и время) - время обновления.

3. 5. Записи справочника характеристик (Characters)

- ID (целое) – ключ;
- Desc (строка) – наименование.

3. 6. Записи справочника валют (Currencies)

- ID (целое) - ключ;
- Desc (строка) - наименование;
- Ratio real - отношение к текущей базовой валюте;
- Code (целое) - код валюты, международный (трехзначный);
- IsDefault (целое) - флаг базовой валюты (1 – валюта "по умолчанию");
- Udt_Ts (дата и время) - время обновления.

3. 7. Записи справочника ответственных лиц (Functionaries)

- ID (целое) - ключ;
- FIO (строка) - ФИО ответственного лица;
- JobTitle (строка) - должность;
- Udt_Ts (дата и время) - время обновления.

3. 8. Записи справочника инвентарных номеров (InventNumbers)

- ID (целое) - ключ;
- Number (строка) - номер;
- Udt_Ts (дата и время) - время обновления.

3. 9. Записи справочника подразделений (OrgDivisions)

- ID (целое) - ключ;
- Desc (строка) - наименование;
- ParentID (целое) - ID родительской категории;
- IsCategory (целое) - тип категории (1 - «категория», 0 - «значение»);
- Udt_Ts (дата и время) - время обновления.

3. 10. Записи справочника единицы измерения (Units)

- ID (целое) - ключ;
- Desc (строка) - наименование;
- Udt_Ts (дата и время) - время обновления.

3. 11. Записи справочника основные средств (Capital)

- ID (целое) - ключ;
- Desc (строка) - наименование;
- ParentID (целое) - ID родительской категории;
- IsCategory (целое) - тип (1 - «категория», 0 - «значение»);
- FuncID (целое) - ID ответственного лица (из справочника Functionaries);
- TagID (строка) - TagID метки;
- OrgDivID (целое) - ID подразделения (из справочника OrgDivisions);
- InvNum (целое) - ID инвентарного номера (из справочника InventNumbers);
- Udt_Ts (дата и время) - время обновления.

3. 12. Записи справочника набора характеристик (CharacterSets)

- Code (целое) - код набора;
- CharID (целое) - ID характеристики (из таблицы Characters);
- Value (строка) - значение характеристики.

3. 13. Записи справочника товаров (Goods)

- ID (целое) - ключ;
- Desc (строка) - наименование товара;
- ParentID (целое) - ID родительской категории;
- IsCategory (целое) - тип категории (1 - «категория», 0 - «значение»);
- IsWeight (целое) - тип товара (1 - «весовой», 0 - «штучный»);
- RestTotal (число с точкой) - общий остаток;
- Article (строка)(255) - артикул;
- IsCharacter (целое) - флаг характеристики (1 – «характеристика»);
- Udt_Ts (дата и время) - время обновления.

3. 14. Записи справочника единицы товаров (GoodsUnits)

- ID (целое) - ключ;
- GID (целое) - ID товара (из справочника Goods);
- Desc (строка) - наименование;
- Ratio real - пропорция;
- PriceBuy real - закупочная цена;

- PriceSell real - цена продажи;
- CurrID (целое) - ID валюты (из справочника Currencies);
- TagID (строка) - TagID метки;
- BC(строка) - штрих – код;
- CharSet (строка) - набора характеристик вида <название>=<значение> разделенных точкой с запятой.

Пример: «Разрешение (dpi)=2400;Цветность (bit)=16;Формат бумаги=A5;»

- UnitID (целое) - ID единицы измерения (из справочника Units);
- Udt_Ts (дата и время) - время обновления.

3. 15. Записи справочника местоположений (Locations)

- ID (целое) - ключ;
- Desc (строка) - наименование;
- FuncID (целое) - ID ответственного лица (из справочника Functionaries);
- ParentID (целое) - ID родительской категории;
- IsCategory (целое) - тип категории (1 - «категория», 0 - «значение»);
- Udt_Ts (дата и время) - время обновления.

3. 16. Записи справочника по остаткам товаров (GoodsAtLocations)

- GID (целое) - ID товара (из справочника Goods);
- UnitID (целое) - ID единицы товара (из справочника GoodsUnits);
- LocID (целое) - ID местоположения (из справочника Locations);
- Rest (целое) - остаток товара.

4. Пример работы с библиотекой на языке программирования с++

```
#include <iostream>
#include <conio.h>
#include <windows.h>
#include <windowsx.h>

// Указатели на функции библиотеки
int (__stdcall *pInit)(wchar_t* DataSource, wchar_t* Login, wchar_t* Password, wchar_t* InitialCatalog,
    wchar_t* InstanceName);
int (__stdcall *pInitFromFile)(wchar_t* ConfigFile);
int (__stdcall *pGetTerminals)(const intptr_t InStr, const int InSize);
int (__stdcall *pGetDocumentHeaders)(wchar_t* TermCode, int AllFlag, const intptr_t InStr, const int InSize);
int (__stdcall *pGetDocumentData)(int DocID, const intptr_t InStr, const int InSize);
int (__stdcall *pGetDictionary)(wchar_t* DictName, const intptr_t InStr, const int InSize);
int (__stdcall *pDeleteDictionary)(wchar_t* DictName, int ID);
int (__stdcall *pInsertDictionary)(wchar_t* DictName, wchar_t* DataString);
int (__stdcall *pGetLastError)(const intptr_t InStr, const int InSize);
int (__stdcall *pImportContract)(int ID, int ParentID, int IsCategory, wchar_t* Name, wchar_t* TIN, wchar_t* RCR);
int (__stdcall *pImportGoods)(int ID, int ParentID, int IsCategory, wchar_t* Name, int IsWeight, int IsCharacter,
    wchar_t* Article);
int (__stdcall *pImportGoodsUnits)(int ID, int GoodID, wchar_t* Name, float Ratio, float PriceBuy, float PriceSell, int CurrID,
    wchar_t* TagID, int UnitID, wchar_t* BarCode, wchar_t* CharSet);
int (__stdcall *pImportCurr)(int ID, wchar_t* Name, float Ratio, int Code, int IsDefault);
int (__stdcall *pImportLocation)(int ID, int ParentID, int IsCategory, wchar_t* Name, int FuncID);
int (__stdcall *pImportUnit)(int ID, wchar_t* Name);
int (__stdcall *pImportFunc)(int ID, wchar_t* Name, wchar_t* Job);
int (__stdcall *pImportCapital)(int ID, int ParentID, int IsCategory, wchar_t* Name, int FuncID, wchar_t* TagID,
    int OrgDivID, wchar_t* InvNum);
int (__stdcall *pImportOrgdiv)(int ID, int ParentID, int IsCategory, wchar_t* Name);
int (__stdcall *pImportInvnum)(int ID, wchar_t* Number);
int (__stdcall *pSetCharSet)(wchar_t* BarCode, wchar_t* CharSet);
int (__stdcall *pGetCharSet)(wchar_t* BarCode, const intptr_t InStr, const int InSize);

void main(void)
{
    setlocale(LC_ALL, "");

    // Загружаем указатели на функции из DLL
    HMODULE hLib = LoadLibrary("../..\\ITP RFID DocumentIntegration DLL\\bin\\Debug\\mobinteg.dll");
    (FARPROC &) pInit = GetProcAddress(hLib, "Init");
    (FARPROC &) pInitFromFile = GetProcAddress(hLib, "InitFromFile");
    (FARPROC &) pGetTerminals = GetProcAddress(hLib, "GetTerminals");
    (FARPROC &) pGetDocumentHeaders = GetProcAddress(hLib, "GetDocumentHeaders");
    (FARPROC &) pGetDocumentData = GetProcAddress(hLib, "GetDocumentData");
    (FARPROC &) pGetDictionary = GetProcAddress(hLib, "GetDictionary");
    (FARPROC &) pDeleteDictionary = GetProcAddress(hLib, "DeleteDictionary");
    (FARPROC &) pinsertDictionary = GetProcAddress(hLib, "InsertDictionary");
    (FARPROC &) pGetLastError = GetProcAddress(hLib, "GetLastError");
    (FARPROC &) pimportContract = GetProcAddress(hLib, "ImportContract");
    (FARPROC &) pimportGoods = GetProcAddress(hLib, "ImportGoods");
    (FARPROC &) pimportGoodsUnits = GetProcAddress(hLib, "ImportGoodsUnits");
    (FARPROC &) pimportCurr = GetProcAddress(hLib, "ImportCurr");
```

```

(FARPROC &) pImportLocation = GetProcAddress(hLib, "ImportLocation");
(FARPROC &) pImportUnit = GetProcAddress(hLib, "ImportUnit");
(FARPROC &) pImportFunc = GetProcAddress(hLib, "ImportFunc");
(FARPROC &) pImportCapital = GetProcAddress(hLib, "ImportCapital");
(FARPROC &) pImportOrgdiv = GetProcAddress(hLib, "ImportOrgdiv");
(FARPROC &) pImportInvnum = GetProcAddress(hLib, "ImportInvnum");
(FARPROC &) pSetCharset = GetProcAddress(hLib, "SetCharset");
(FARPROC &) pGetCharset = GetProcAddress(hLib, "GetCharset");

// Инициализация
int result = pInit(L"192.168.0.113", L"sa", L"123host+", L"ITProject_RFID_Journal", NULL);
if(result != 0)
{
    wchar_t* error = new wchar_t[65535];
    pGetLastError((intptr_t) (void*) error, 65535);
    wprintf(error);

    _getch();
    return;
}

// Получение данных
wchar_t* terminals = new wchar_t[65535];
pGetTerminals((intptr_t) (void*) terminals, 65535);
wprintf(terminals);

wchar_t* headers = new wchar_t[65535];
pGetDocumentHeaders(L"4010B109-0060-0000-0000-0040056A0E9D", 0, (intptr_t) (void*) headers, 65535);
wprintf(headers);

wchar_t* datas = new wchar_t[65535];
pGetDocumentData(1, (intptr_t) (void*) datas, 65535);
wprintf(datas);

wchar_t* dics = new wchar_t[65535];
pGetDictionary(L"Contractors", (intptr_t) (void*) dics, 65535);
wprintf(dics);

// Удаление данных
pDeleteDictionary(L"Contractors", 1);
pDeleteDictionary(L"Contractors", 2);
pDeleteDictionary(L"Contractors", 3);

// Добавление данных
wchar_t* dics1 = L"<Row><ID>1</ID><Desc>Поставщики</Desc><ParentID /><IsCategory>1</IsCategory>" +
    L"<TIN /><RCR /><Udt_Ts>05.10.2012 21:34:06</Udt_Ts></Row>";
wchar_t* dics2 = L"<Row><ID>2</ID><Desc>ПК \"Компас-С\"</Desc><ParentID>1</ParentID>" +
    L"<IsCategory>0</IsCategory><TIN>098765432109</TIN><RCR>456456456</RCR>" +
    L"<Udt_Ts>05.10.2012 21:35:22</Udt_Ts></Row>";
wchar_t* dics3 = L"<Row><ID>3</ID><Desc>Альбатрос</Desc><ParentID>1</ParentID>" +
    L"<IsCategory>0</IsCategory><TIN>132435465768</TIN><RCR>789789789</RCR>" +
    L"<Udt_Ts>24.09.2012 13:38:39</Udt_Ts></Row>";
pInsertDictionary(L"Contractors", dics1);
pInsertDictionary(L"Contractors", dics2);
pInsertDictionary(L"Contractors", dics3);

_getch();
}

```