|  |  |
| --- | --- |
| **УТВЕРЖДАЮ** | **УТВЕРЖДАЮ** |
| Главный конструктор  ФГУП «Комбинат Электрохимприбор» | Заместитель директора – директор Департамента ТИМ  ООО «РЦ «АСКОН-Урал» |
| **\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_** А.А. Кощеев | \_\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_ А.П. Птицин |
| « » \_\_\_\_\_\_\_\_\_\_\_\_2022 г. | « » \_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. |

ЧАСТНОЕ ТЕХНИЧЕСКОЕ ЗАДАНИЕ

на адаптацию программного продукта Pilot-ICE в части разработки библиотеки dll

|  |  |  |  |
| --- | --- | --- | --- |
|  | | СОГЛАСОВАНО | |
|  | |  | |
|  | |  | |
|  | | «\_\_\_»\_\_\_\_\_\_\_\_\_\_\_\_\_\_ 2022 г. | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |
|  |  | |

Челябинск

2022

УПРАВЛЕНИЕ ДОКУМЕНТОМ

|  |  |
| --- | --- |
| **Автор** |  |
| **Дата последнего редактирования** | 09.12.2022 |

ВЕРСИИ

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Версия** | **Дата изменения** | **Причина изменения** | **Автор изменения** | **Подпись** |
| 1.0 | 09.12.2022 | Создание первой версии документа |  |  |
|  |  |  |  |  |
|  |  |  |  |  |
|  |  |  |  |  |

СОГЛАСОВАНО

**от ФГУП «Комбинат Электрохимприбор»**

|  |  |  |  |
| --- | --- | --- | --- |
| **Должность** | **ФИО** | **Дата** | **Подпись** |
| Зам. главного конструктора | Дьячков Д.В. |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

**от ООО РЦ «АСКОН-Урал»**

|  |  |  |  |
| --- | --- | --- | --- |
| **Должность** | **ФИО** | **Дата** | **Подпись** |
| Технический директор | Керн Н.А. |  |  |
| Ведущий менеджер | Якупов А.Р. |  |  |
| Руководитель проектов | Малявкин С.А. |  |  |

Оглавление

[Введение 4](#_Toc121469311)

[1. Основание для адаптации 4](#_Toc121469312)

[2. Требования к API модуля 5](#_Toc121469313)

[3. Требования функциям модуля 6](#_Toc121469314)

[4. Требования к предоставлению результатов работы модуля 7](#_Toc121469315)

[5. Стадии и этапы адаптации 7](#_Toc121469316)

# Введение

Для автоматизации загрузки электронной конструкторской документации в систему управления проектными данными Pilot ICE (далее СУПД) должно быть разработано комплексное решение, состоящее из двух программных частей: библиотеки dll и модуля для интеграции корпоративного программного обеспечения (далее ПО) с СУПД.

Комплексное решение предназначено для импорта документов с сохранением связей между документами в системе управления проектными данными Pilot ICE. Данные для загрузки представляют собой электронные конструкторские документы (ДЭ): документы в различных форматах (pdf, xps, doc, cdw и т.п.), представленные отдельными файлами, а также набор свойств (атрибутов) документа – наименование, обозначение, тип и др.

Цель адаптации — внесение документов и их свойств в Pilot-ICE на основании информации из корпоративного ПО.

Комплексное решение должно содержать следующие функции:

1. создание/удаление документа в СУПД и заполнение его атрибутов;
2. создание/удаление горизонтальных связей между документами в СУПД;
3. прикрепление/удаление исходного файла;
4. заморозка/разморозка документа;
5. прикрепление файла XPS(XPS, PDF, tiff, png, jpeg);
6. создание задания на ознакомление;
7. простановка электронной цифровой подписи (ЭЦП).

Краткое описание работы библиотеки dll:

Преобразование исходящих запросов из корпоративного ПО в рамках среды программирования Embarcadero Delphi 10+ в машинный язык C#.

Краткое описание работы модуля:

На основании преобразованных запросов выполнять функции, перечисленные , возвращать результат обработки или код ошибки пользователю.

# Основание для адаптации

Основанием для адаптации является Договор № 083-399 на оказание услуг по адаптации программного продукта.

# Требования к API модуля

Для подключения к модулю сторонним ПО должна быть разработана библиотека в виде нативного интерфейса (файл dll), которая экспортирует следующие методы:

1. static extern void Search([MarshalAs(UnmanagedType.LPWStr)] out string result, [MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)]string query);
2. static extern void Export([MarshalAs(UnmanagedType.LPWStr)] out string result, [MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPArray, SizeParamIndex = 2)] Guid[] ids, int idsCount, int reserv);
3. static extern int CreateDocument([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString, [MarshalAs(UnmanagedType.LPWStr)] string guidParentString, int typeId);
4. static extern int DeleteDocument([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString);
5. static extern int SetAttribute([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString, [MarshalAs(UnmanagedType.LPWStr)] string attributeName, [MarshalAs(UnmanagedType.IUnknown)] object attributeValue);
6. static extern int CreateLink([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidFirstObjectString, [MarshalAs(UnmanagedType.LPWStr)] string guidSecondObjectString);
7. static extern int DeleteLink([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidFirstObjectString, [MarshalAs(UnmanagedType.LPWStr)] string guidSecondObjectString);
8. static extern int AttachSourceFile([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString, [MarshalAs(UnmanagedType.LPWStr)] string filePath);
9. static extern int DeleteSourceFile([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString);
10. static extern int AttachXPS([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString, [MarshalAs(UnmanagedType.LPWStr)] string filePath);
11. static extern int FreezeDocument([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString);
12. static extern int UnFreezeDocument([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString);
13. static extern int SignDocument([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidObjectString, [MarshalAs(UnmanagedType.LPWStr)] string role);
14. static extern int CreateTaskToReadDocument([MarshalAs(UnmanagedType.LPWStr)] string pilotUri, [MarshalAs(UnmanagedType.LPWStr)] string guidDocumentString, [MarshalAs(UnmanagedType.SafeArray, SafeArraySubType = VarEnum.VT\_INT)] int[] userIDs, [MarshalAs(UnmanagedType.LPWStr)] string taskHeader, [MarshalAs(UnmanagedType.LPWStr)] string taskText);

# Требования функциям модуля

Выполнение функций модулем осуществляется через клиентское API.

3.1 Создание документа в СУПД

Требования к работоспособности создания объектов модулем: при получении GUID объекта, GUID родительской папки и ID создаваемого объекта - модуль создает новый объект СУПД без заполненных атрибутов. Наименование объекта при этом отображается как «Нет атрибутов для отображения».

3.2 Удаление объекта в СУПД

Требования к работоспособности создания объектов модулем: при получении GUID удаляемого объекта - модуль помещает удаляемый объект в корзину вместе с исходными файлами, связями и потомками.

3.3 Заполнение атрибутов объекта в СУПД

Требования к работоспособности заполнения атрибутов объекта модулем: при получении GUID объекта, имени атрибута и значения атрибута - происходит заполнение карточки объекта в области указанного атрибута.

3.4 Создание горизонтальных связей между объектами в СУПД

Требования к работоспособности создания горизонтальной связи объектов модулем: при получении модулем GUID двух связываемых друг с другом объектов в СУПД - создается горизонтальная связь.

3.5 Удаление горизонтальных связей между объектами в СУПД

Требования к работоспособности удаления горизонтальной связи объектов модулем: при получении модулем GUID двух связанных друг с другом объектов в СУПД - удаляется указанная горизонтальная связь.

3.6 Прикрепление исходного файла

Требования к работоспособности функции добавления исходных файлов к объекту модулем: при получении модулем GUID объекта, к которому необходимо прикрепить файл и пути исходного файла в формате Путь\Наименование файла.расширение - в СУПД создается родительская связь между объектом (родитель) и добавленным исходным файлом. Добавленный файл отображается во вкладке «Файлы».

3.7 Удаление исходного файла

Требования к работоспособности удаления исходного файла модулем: при получении модулем GUID исходного файла, который необходимо удалить, модуль удаляет исходный файл из СУПД.

3.8 Заморозка (и разморозка) документа

Требования к работоспособности заморозки (разморозки) объектов модулем: при получении модулем GUID объекта, который необходимо защитить от любых изменений (заморозить/разморозить), модуль замораживает (размораживает) объект в системе.

3.9 Прикрепление файла XPS к документу

Требования к работоспособности функции добавления файлов XPS к документу модулем: при получении GUID документа и пути файла XPS в виде Путь\Наименование файла.расширение (форматы: xps, pdf, png.tiff.tif.jpeg.jpg) - модуль загружает выбранный документ в окно предпросмотра документа. Изображения (.png.tiff.tif.jpeg.jpg) загружаются в файлы, заменяя существующие изображения и документы (xps и pdf)Для того, чтобы увидеть загруженный снова файл, нужно два раза кликнуть по объекту. В окне предпросмотра документ не отображается, для этого нужно внедрить механизм AutoImport.

3.10 Создание задания на ознакомление

Требования к работоспособности функции создания задания на ознакомление модулем: при получении модулем GUID объекта, который необходимо вложить в задание на ознакомление, Main Position исполнителя задания – модуль реализует функцию выдачи задания на ознакомление с вложением в виде выбранного объекта. Дополнительно есть возможность указать «Заголовок» и «Описание» к заданию.

3.11 Требования к работоспособности функции простановки подписи (ЭЦП) модулем: при получении модулем GUID объекта, который необходимо подписать, документ подписывается сертификатом пользователя (Администратора) с созданием запроса на подпись. Дополнительно можно задать роль подписанта.

# Требования к предоставлению результатов работы модуля

4.1 Все функции dll при возникновении ошибки должны возвращать код ошибки. 0 – без ошибок, 1 – ошибка.

4.2. Функции dll должны работать в условиях использования библиотеки dll при разработке в рамках среды программирования Embarcadero Delphi 10+ методом подключения и вызова функций из внешней dll.

# Стадии и этапы адаптации

Адаптация осуществляется по следующим этапам:

* 1. Разработка и согласование настоящего технического задания;
  2. Формирование комплексного решения согласно подписанному техническому заданию;
  3. Разработка инструкции по работе с библиотекой dll и модулем в СУПД;
  4. Разработка и согласование программы и методики испытаний (ПМИ) библиотеки dll и модуля в СУПД;
  5. Техническая поддержка по установке библиотеки dll в СУПД в информационной системе ФГУП «Комбинат «Электрохимприбор»;
  6. Техническая поддержка проведения испытаний в соответствии с ПМИ библиотеки dll в СУПД на ФГУП «Комбинат «Электрохимприбор».