

Руководство пользователя

АС УДИ ШД - модуля к программной платформе
T-FLEX-DOCS

Оглавление

Введение	3
Установка модуля	3
Импорт ddx файлов	3
Регистрация и настройка web-сервиса	6
Windows	7
Linux	10
Базовая конфигурация модуля	11
Конфигурация справочника «Шина данных».	11
Типы справочника	12
Пользовательский интерфейс справочника «Шина данных»	13
Макросы	15
Web-сервис	15
Метод ReceiveIncomingMessage	16
Метод GetIncomingMessageStatus	16
Конфигурационный файл.	17
Основные элементы.	17
Файл данных	17
Описание работы приложения	17
Алгоритмы обработки	18
API модуля	18
Добавление новой исходной системы для интеграции	19
Приложения	21
Приложение 1: XSD схема конфигурационного XML файла	21
Приложение 2: XSD схема xml файла входящего сообщения.	22

Введение

Программное обеспечение предоставляет возможность организовать миграцию данных из сторонних гетерогенных для T-Flex DOCs систем.

Отличительные особенности

- API для написания алгоритмов обработки любых исходных данных.
- Поддержка возможности предобработки входных данных сортировка, нормализация и т.п.
- Поддержка обработки в режиме транзакции с возможностью отката всего пакета данных в случае критической ошибки.
- Простая и понятная настройка на основании служебного справочника и файла конфигурации в формате XML.
- Наличие готовой базовой конфигурации для получения и обработки данных в predetermined формате.

Установка модуля

Для установки модуля необходимо загрузить базовую конфигурацию используя стандартный импорт конфигурации через ddx файлы.

Импорт ddx файлов

Импортировать ddx файлы необходимо в следующем порядке:

1. Шина данных
2. Файлы
3. Макросы
4. Сервис шины данных
5. Документы
6. ЭСИ

1. В T-FLEX DOCs запустить окно импорта данных, для этого в левом верхнем углу нажать «Вид» и выбрать «Импорт...».

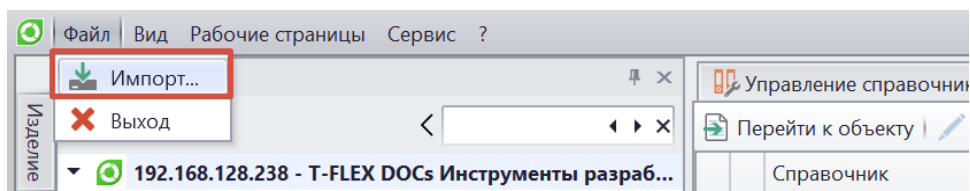


Рисунок 1 Запуск импорта

2. В открывшемся окне выбрать один из шести ddx файлов.

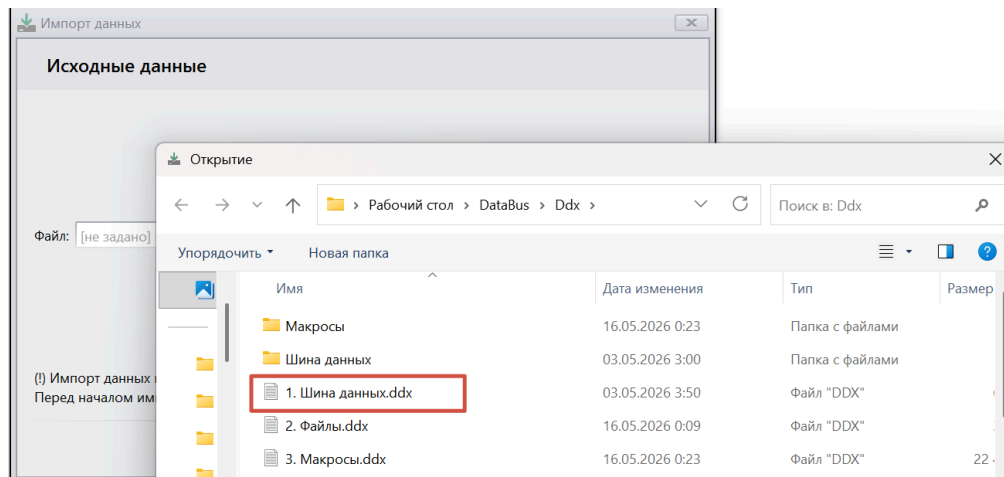


Рисунок 2 Окно выбора файла ddx

3. Нажать «Далее >».

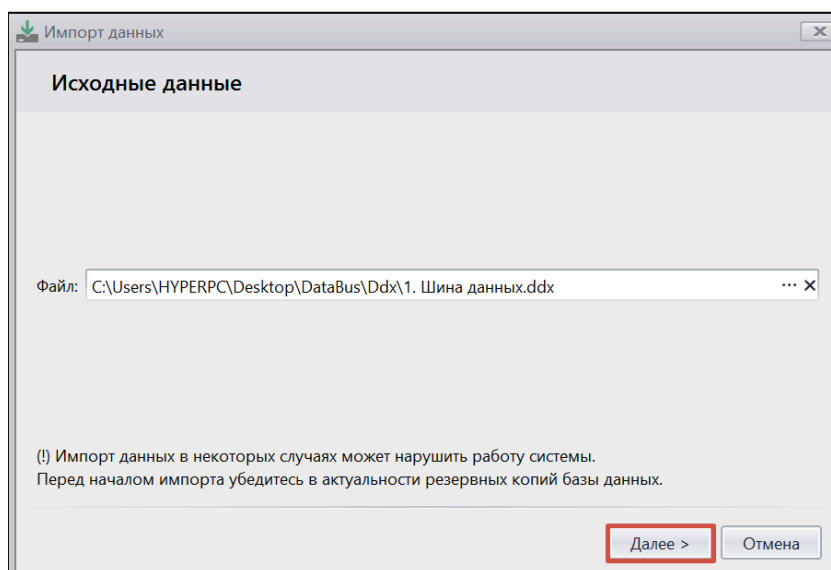


Рисунок 3

4. Поставить галочки, как показано на примере. Нажать «Далее >».

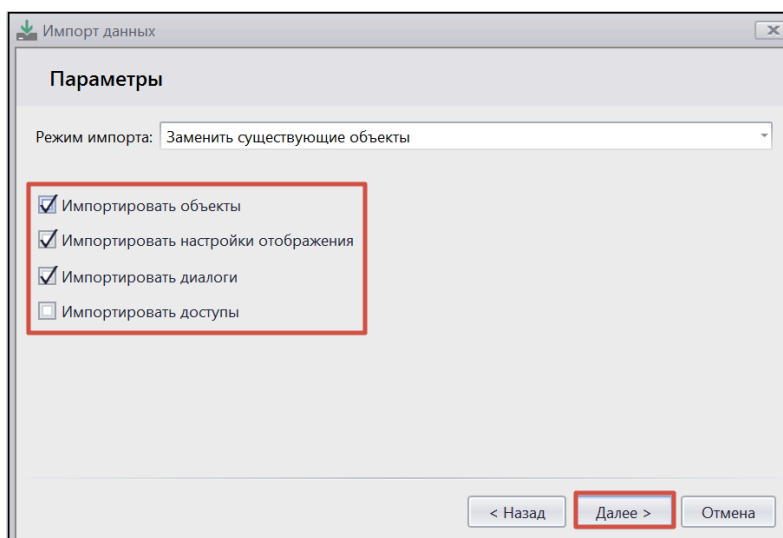


Рисунок 4 Настройка импорта

5. После успешного импорта нажать «Далее >».

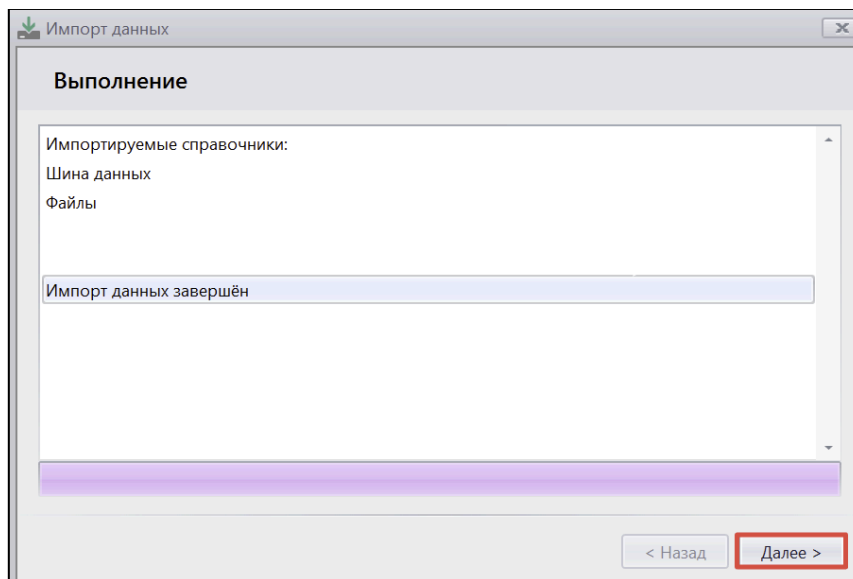


Рисунок 5 Процесс импорта

6. Для применения изменений импортированных объектов (если они были) нажать «Открыть».

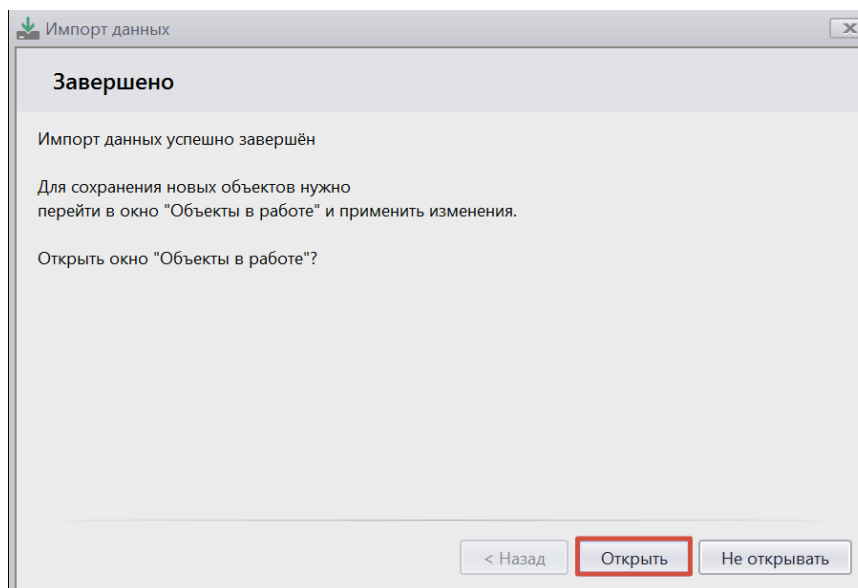


Рисунок 6 Завершение импорта

7. В открывшемся окне «Объекты в работе» необходимо выделить импортированные объекты (при их наличии) и нажать «Применить изменения» (иконка дискеты).

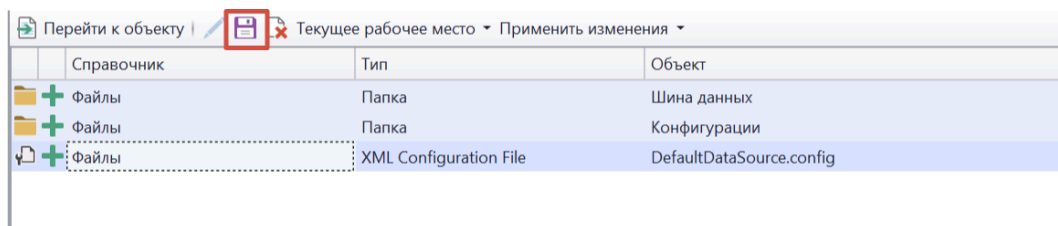


Рисунок 7 Объекты в работе

8. В открывшемся окне «Применение изменений» заполнить поле «Комментарий» и нажать «ОК».

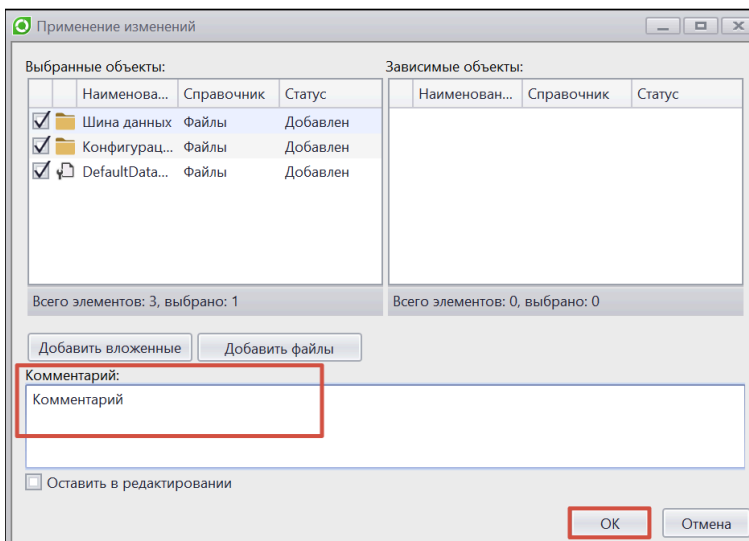


Рисунок 8 Окно применения изменений

Регистрация и настройка web-сервиса

1. Откройте справочник **Группы и Пользователи** в группе **Администраторы**. Создайте локального пользователя, под которым будет работать интеграция см. **Рисунок 1**. В качестве конфигурации для входа укажите **T-Flex DOCs Инструменты разработки**.

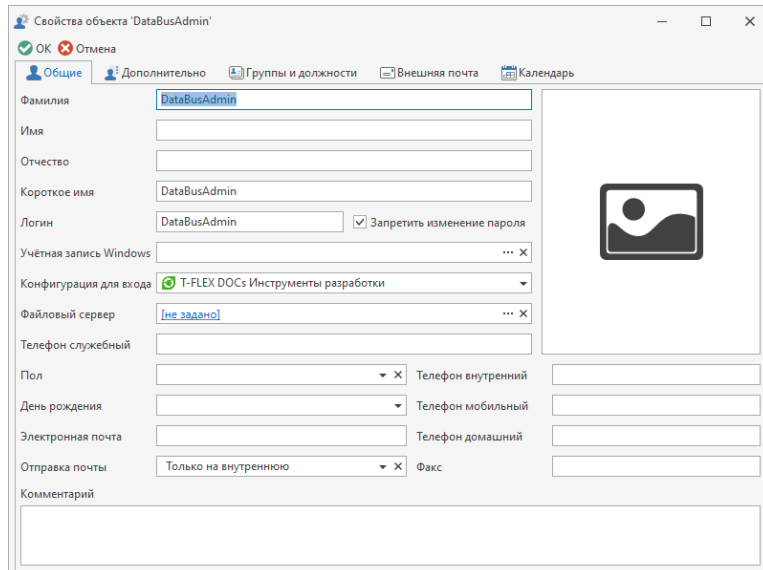


Рисунок 9 Создание пользователя

2. После создания пользователя, вызовите контекстное меню **Изменить пароль пользователя** и задайте пароль см. **Рисунок 2**.

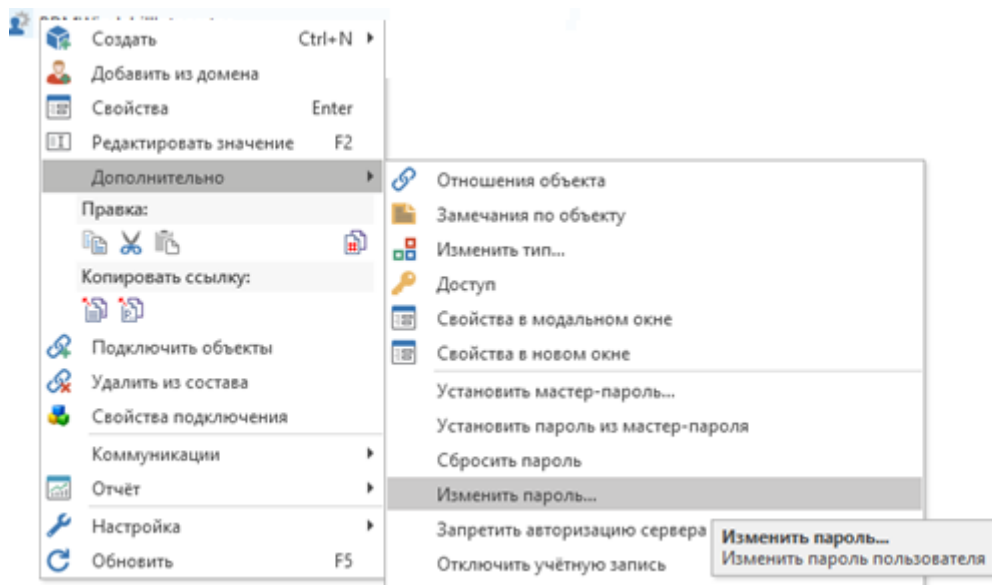


Рисунок 10 Задать пароль пользователя

Windows

1. Запустите клиент **T-Flex DOCs** с правами администратора.
2. Войдите в систему под учетной записью администратора с конфигурацией **T-Flex DOCs Инструменты разработки** см. Рисунок 3.

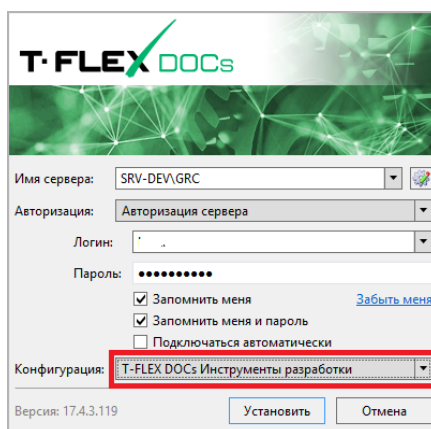


Рисунок 11

3. Откройте справочник **WEB Сервисы** см. Рисунок 4

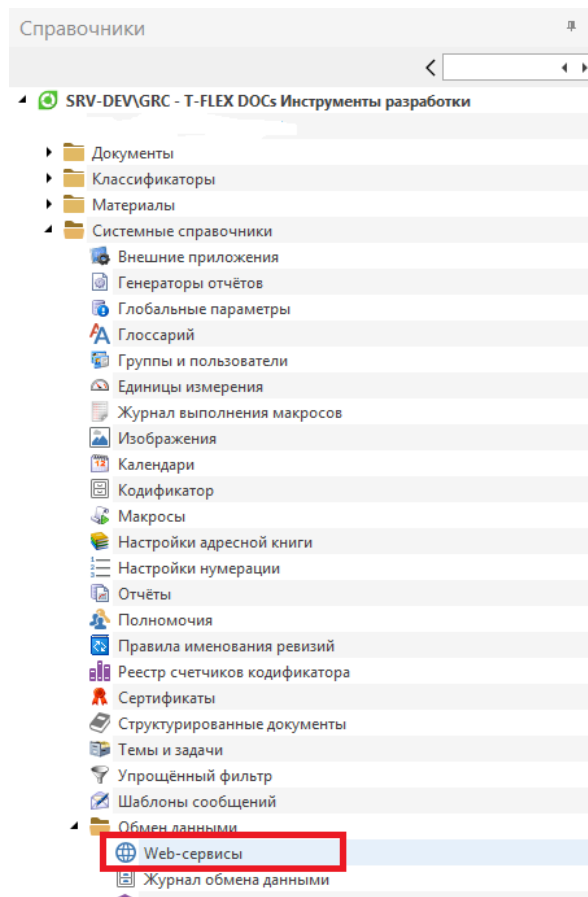


Рисунок 12 Справочник WEB Сервисы

4. В открывшемся окне справочника **WEB Сервисы** выберите Сервис шины данных см. *Рисунок 5*

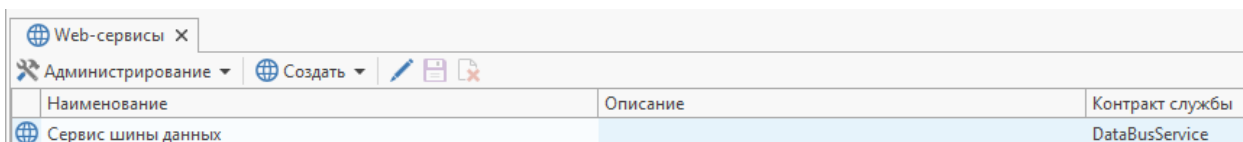


Рисунок 13 Сервис шины данных

5. Правой кнопкой вызовите контекстное меню и выберите пункт **Опубликовать**.

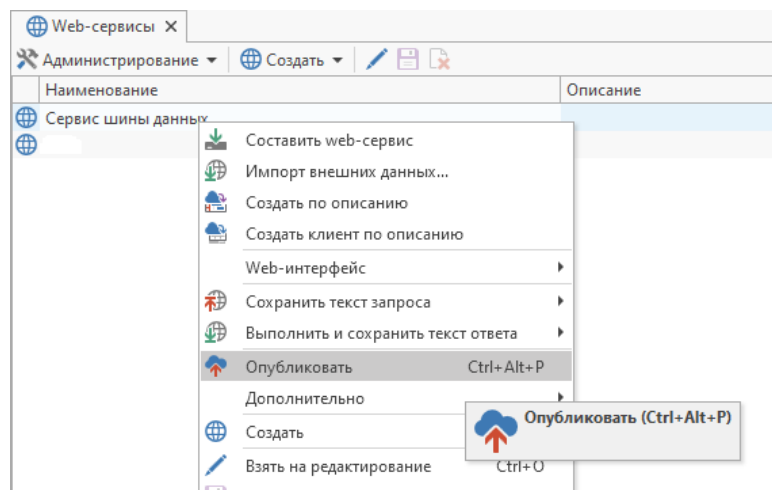


Рисунок 14

6. В открывшемся окне **Настройки сайта web-сервиса** укажите **порт** и **путь** для публикации сервиса, либо оставьте все без изменений.

Внимание: Указанный порт должен быть открыт в **брандмауэре!**

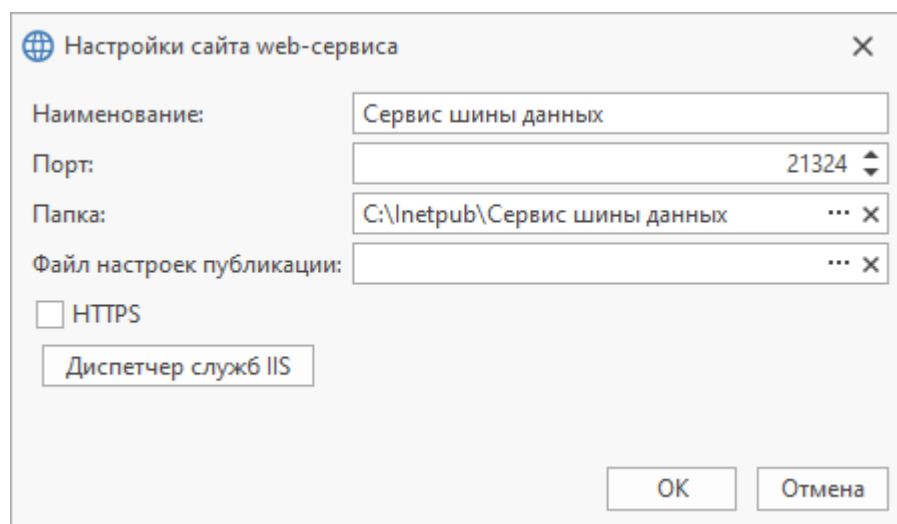


Рисунок 15 Окно настройки сайта web-сервиса

7. Нажмите **ОК**.
8. После успешной публикации будет предложено перейти на тестовую страницу сервиса.

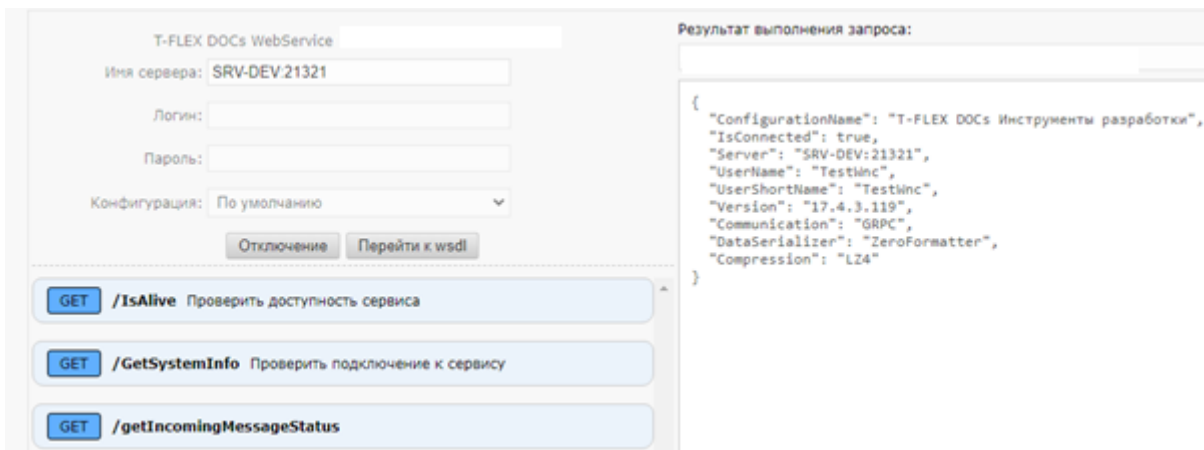


Рисунок 16 Тестовая страница сервиса

9. Перейдите в папку, в которую были установлены файлы сервиса, откройте файл **Web.config** в текстовом редакторе и укажите логин и пароль и конфигурацию входа пользователя под которым будет осуществляться работа сервиса.

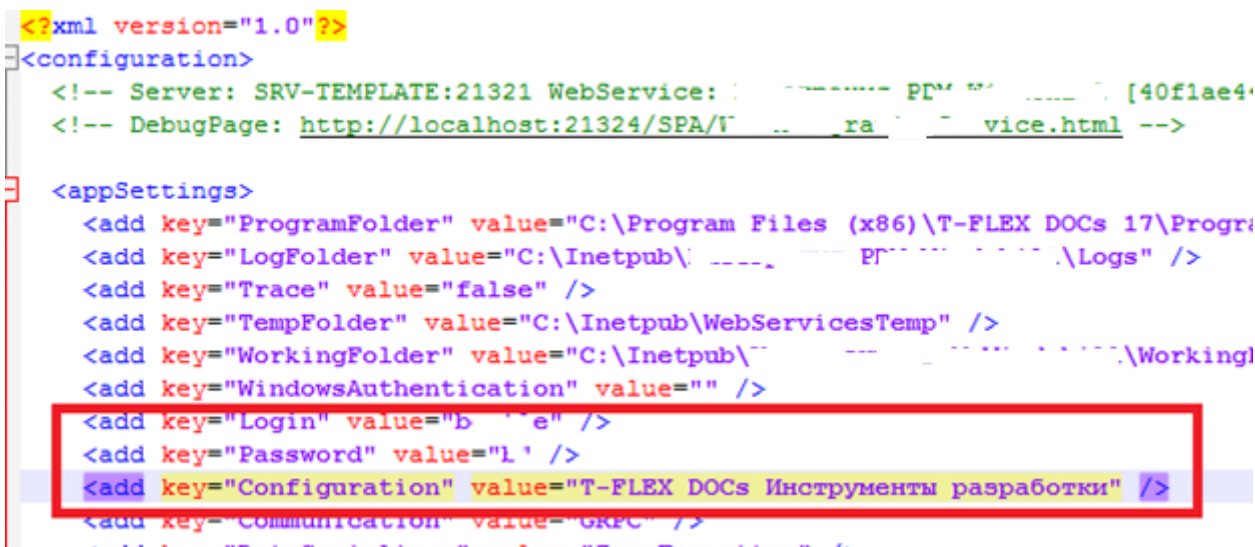


Рисунок 17 Файл Web.config

10. Нажмите **Win+R** и запустите **InetMgr.exe**. Откроется окно **Диспетчер служб IIS**. Выберите опубликованный сервис и измените правила привязки.

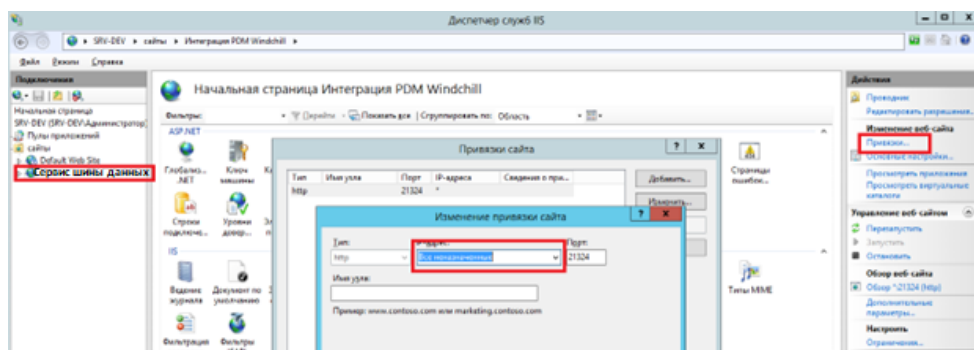


Рисунок 18 Окно Диспетчер служб IIS

11. Перезапустите сервис.

Linux

1. Установить пакет

```
sudo apt install ./databus-web-service.deb
```

2. В файле

```
/opt/t-flex-docs/web-services/databus/webservice.json
```

указать корректный ip и порт сервера T-FLEX-DOCs

```
"Server": "192.168.128.238:21321"
```

3. Сохранить файл

4. Выполнить команду

```
sudo systemctl restart t-flex-docs-databus.service
```

5. После успешной установки по адресу <http://<ip сервера>:<port сервера>/debug/standard> должна быть доступна страница **swagger**

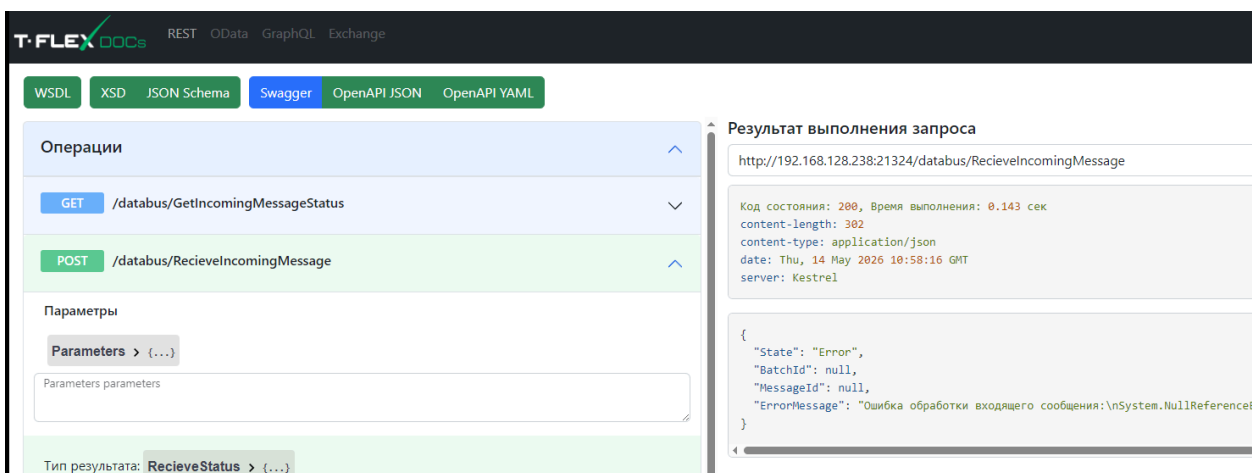


Рисунок 19

Базовая конфигурация модуля

Модуль расширяет конфигурацию базовой поставки T-Flex DOCs и включает в себя:

- справочник «Шина данных»
- макросы «Шина данных. Web-сервис», «Шина данных. Задачи»
- web-сервис «Сервис шины данных»
- задача «Transaction manager. Default»

Конфигурация справочника «Шина данных».

Справочник предназначен:

- для хранения и структурирования полученных данных
- формирования очереди обработки
- логирования и обеспечения целостности передаваемых данных
- мониторинга выполнения заданий по созданию объектов T-Flex DOCs.

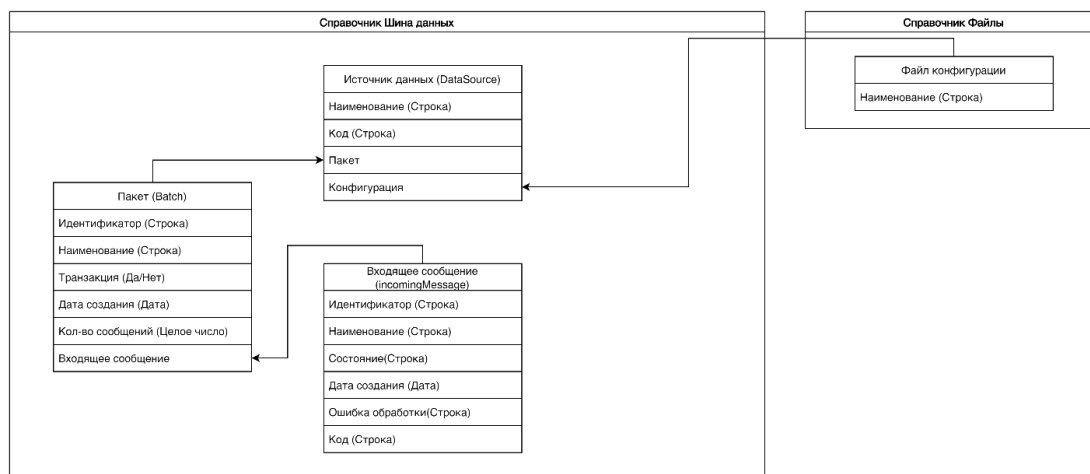


Рисунок 20 Структурная схема справочника Шина данных

Типы справочника

Источник данных (DataSource)

Корневой объект для определения исходной системы отправителя данных.

Используется для структурирования данных внутри справочника и определения конфигурации интеграции, а также алгоритмов обработки входящих данных из исходной системы.

Параметры

- Наименование (Строка) – отображаемое наименование.
- Код источника (Строка/Список).

Связи

- Конфигурация (Справочник Файлы/Файл) – ссылка на xml файл конфигурации интеграции для текущей системы.

Состоит из

- Пакет (Справочник Шина данных/Пакет)

Пакет (Batch)

Объект для структурирования входящих данных. Содержит набор входящих сообщений. Используется для предварительной обработки полученных данных (сортировка, нормализация...), а также для обработки в одной транзакции.

Параметры

- Идентификатор (Строка) – уникальный идентификатор

- Наименование (Строка) – отображаемое наименование
- Транзакция (Да/Нет) – указывает что пакет должен обрабатываться в одной транзакции с возможностью отката всех изменений в результате ошибки обработки очередного сообщения.
- Кол-во сообщений (Целое число) – ожидаемое кол-во сообщений, которое должно быть получено из исходной системы.
- Состояние (Строка/Список) – состояние жизненного цикла обработки пакета:
 - Создание
 - Выполнение
 - Завершено
 - Ошибка

Состоит из

- Входящее сообщение (Справочник Шина данных/ Входящее сообщение)

Входящее сообщение (IncomingMessage)

Объект для хранения файлов данных, полученных из системы источника. Хранит метаданные в формате xml/Json необходимые для создания объектов TFlexDOCs, а также любые файлы данных исходной системы. Содержит ссылки на созданные в результате объекты справочников TFlexDOCs.

Параметры

- Идентификатор (Строка) – уникальный идентификатор
- Идентификатор содержимого (Строка) – уникальный идентификатор, используется для типа Content, для хранения файлов.
- Наименование (Строка) – отображаемое наименование
- Номер п/п (Целое число) – порядковый номер сообщения
- Состояние (Строка) – содержит информацию о текущем состоянии обработки из предопределённого набора:
 - Получено
 - Обработка
 - Ошибка
 - Выполнено
- Тип данных (Строка/Список) – по умолчанию содержит 2 значения:
 - Content – для хранения любых файлов исходной системы
 - Metadata – для хранения xml файлов метаданных.
- Ошибка обработки (Строка) – содержит лог ошибки обработки.

Связи

- Файлы (Справочник Файлы/Файл) – ссылка на сохраненные файлы.

Пользовательский интерфейс справочника «Шина данных»

Пользовательский интерфейс предназначен для мониторинга процесса получения и обработки данных. Справочник отображается в виде древовидной структуры, состоящей из источников данных см. *Рисунок 2*.

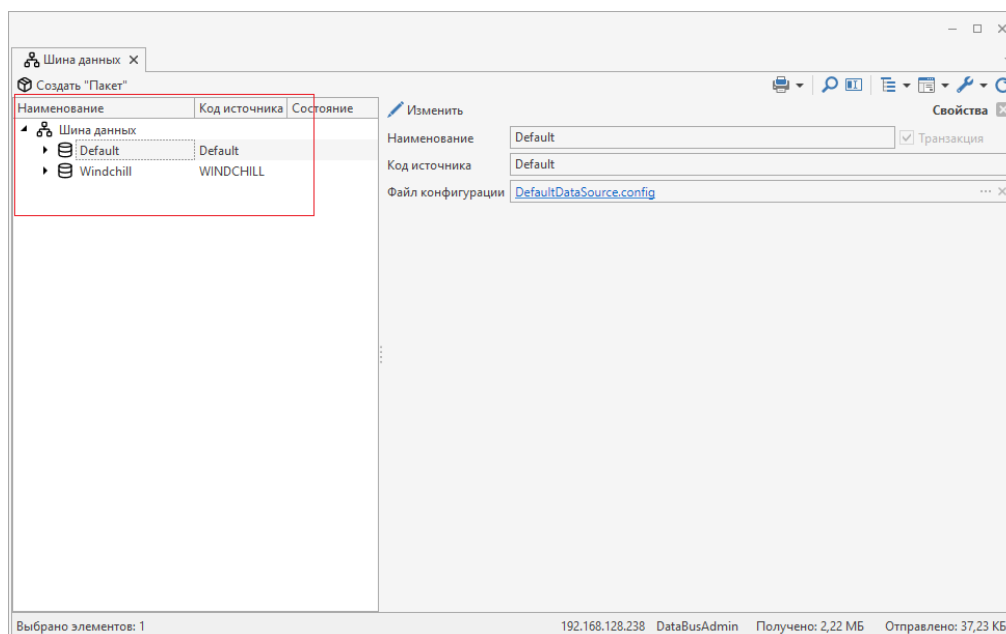


Рисунок 21 Источники данных

Каждый источник данных содержит список полученных Пакетов и состояние их обработки см. Рисунок 14 (1). При выборе Пакета доступен интерфейс отображающий его состав в виде списка Входящих сообщений их типов, статуса и ошибок обработки см. Рисунок 14 (2).

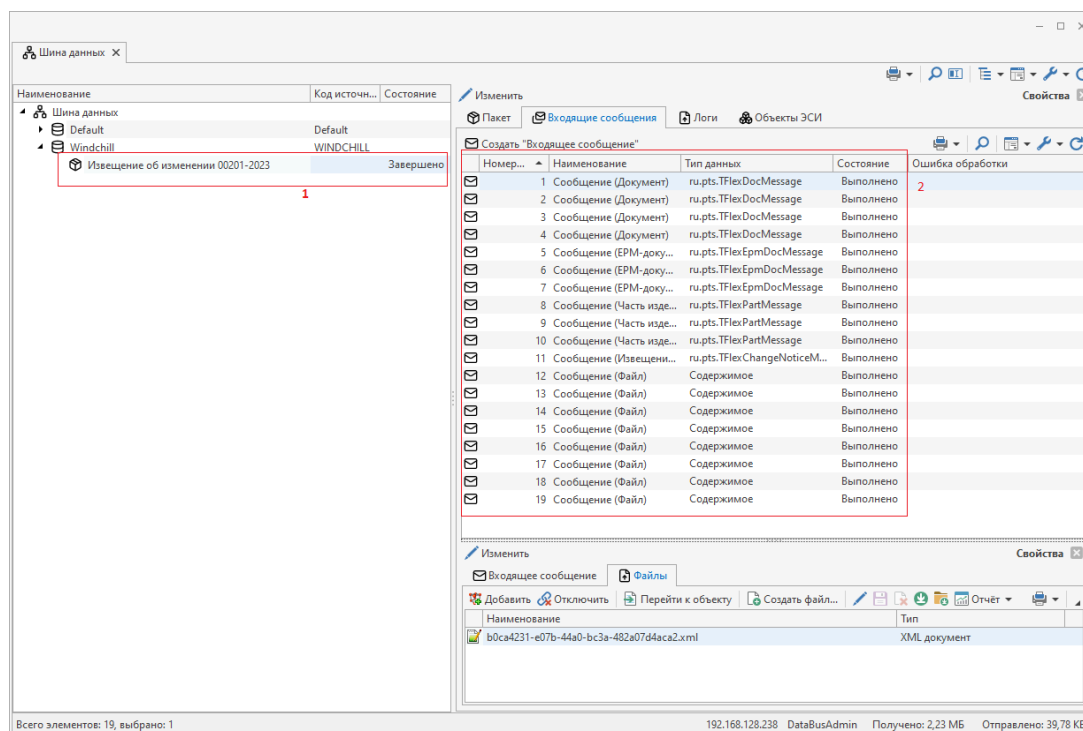


Рисунок 22 Пакет и входящие сообщения

В этом же окне доступны вкладки «Логи» и «Объекты ЭСИ». Вкладка «Логи» содержит ссылки на файлы логов процесса выполнения обработки каждого сообщения пакета см. *Рисунок 15*. На вкладке «Объекты ЭСИ» отображаются ссылки на все созданные объекты справочника ЭСИ см. *Рисунок 16*.

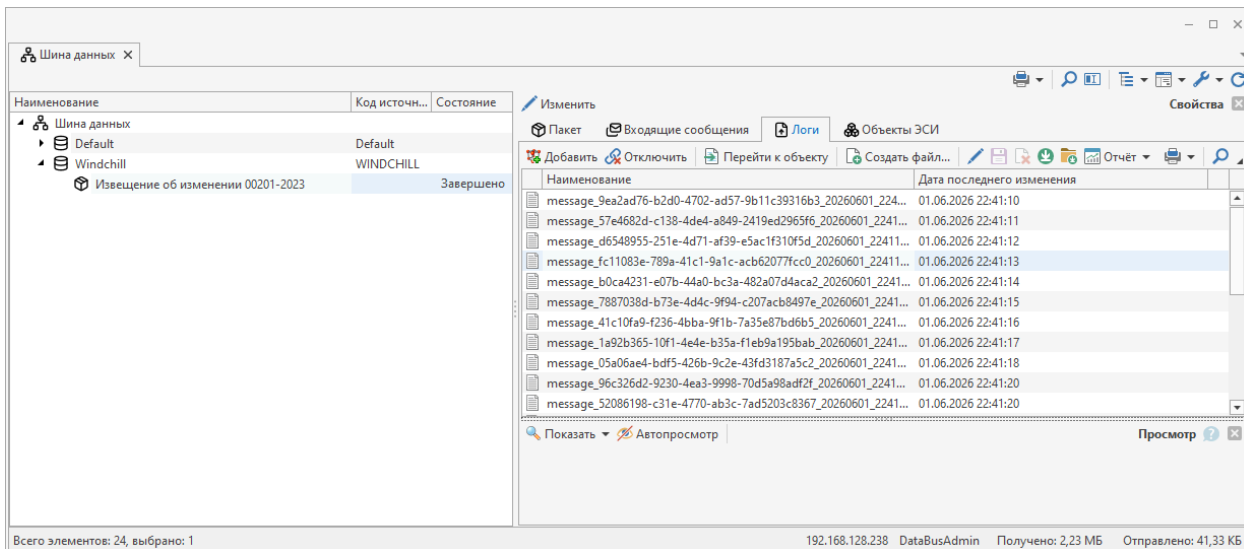


Рисунок 23 Логи обработки пакета

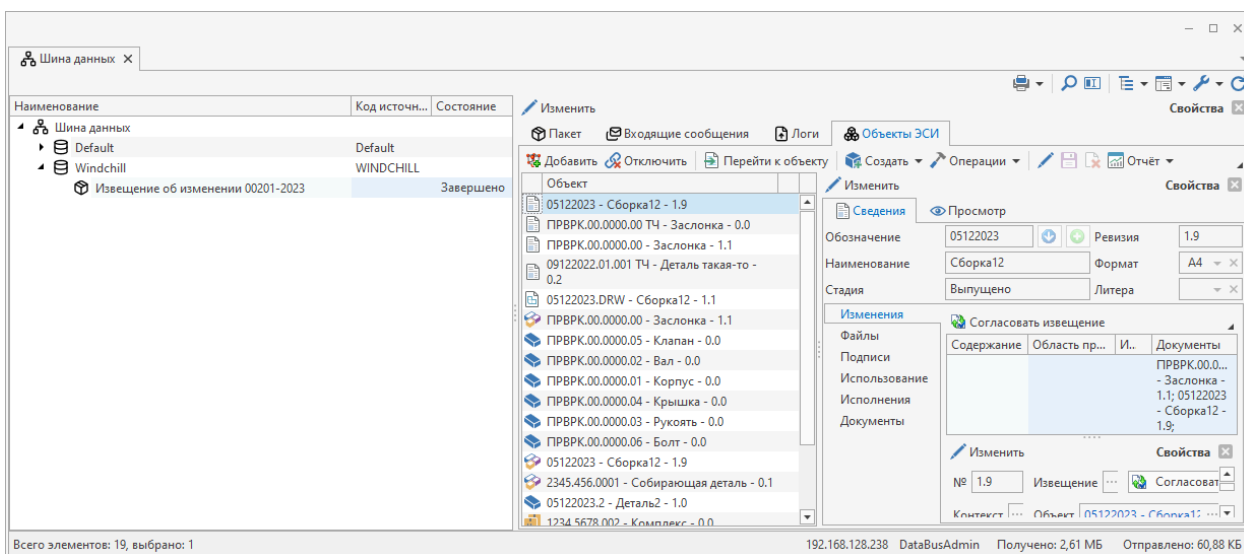


Рисунок 24 Объекты ЭСИ

Макросы

- Шина данных. Web-сервис – содержит методы определяющие входные точки web-сервиса.
- Шина данных. Задачи – содержит список методов, которые используются для создания заданий обработки пакетов по расписанию.

Web-сервис

Для получения данных из сторонних систем используется механизм web-сервиса.

Предустановленный веб сервис предоставляет базовый интерфейс:

- o POST /databus/RecieveIncomingMessage
в зависимости от набора полученных параметров создает пакет, связанное входящее сообщение в справочнике шина данных и файлы метаданных/содержимого
- o POST /databus/Connect
подключиться к серверу (стандартный метод **T-FLEX DOCS**)
- o GET /databus/GetIncomingMessageStatus
получает состояние пакета и входящего сообщения по их идентификаторам
- o GET /databus/IsAlive
проверяет доступность сервиса (стандартный метод **T-FLEX DOCS**)
- o GET /databus/GetSystemInfo
получить системную информацию о подключении (стандартный метод **T-FLEX DOCS**)
- o GET /databus/GetConfigurations
получить список конфигураций (стандартный метод **T-FLEX DOCS**)
- o GET /databus/Disconnect
отключить (стандартный метод **T-FLEX DOCS**)

Метод ReceiveIncomingMessage

Назначение:

Метод предназначен для приема входящего сообщения и связанного с ним файла от внешней информационной системы.

Сигнатура

ReceiveIncomingMessage(Parameters parameters)

Входные параметры

Метод принимает объект Parameters, содержащий следующие поля:

Параметр	Тип	Описание
DataSourceCode	string	Код системы-источника.
BatchId	string	Уникальный идентификатор пакета.
BatchName	string	Наименование пакета.
MessageId	string	Уникальный идентификатор входящего сообщения.
MessageName	string	Наименование входящего сообщения.
MessageType	string	Тип входящего сообщения.
MessageIndex	int	Порядковый номер сообщения в пакете.

MessageCount	int	Общее количество сообщений в пакете.
FileDataBase64	string	Содержимое файла в формате Base64.
FileName	string	Имя передаваемого файла.
Folder	string	Каталог для сохранения файла.
AdditionalParameters	List<Parameter>	Дополнительные параметры сообщения в виде списка «имя–значение».

Метод GetIncomingMessageStatus

Назначение:

Метод предназначен для получения текущего статуса обработки входящего сообщения и статуса пакета, в который данное сообщение входит.

Сигнатура

GetIncomingMessageStatus(string batchId, string incomingMessageId)

Входные параметры

Параметр	Тип	Описание
batchId	string	Идентификатор пакета.
incomingMessageId	string	Идентификатор входящего сообщения.

Конфигурационный файл.

Xml файл для определения алгоритмов обработки. Содержит информацию о сопоставлении атрибутов и типов данных исходной системы и TFlexDOCs. Схема файла базовой конфигурации приведена в [Приложении 1](#).

Основные элементы.

- stages - используется для сопоставления состояний объектов, стадий жизненного цикла.
- configSpec - используется для сопоставления разделов спецификаций конструкторской документации
- batchPreprocessors – препроцессоры для предварительной обработки полученных данных.
- mapRule – правило сопоставления типов
 - srcType – внутреннее обозначение исходного типа
 - docsType – GUID типа TFlexDOCs
 - mapAlgorithm – наименование класса, содержащего алгоритм обработки

- ruleAtt – правило сопоставления атрибутов
 - srcId – внутреннее обозначение исходного атрибута
 - docId – GUID параметра в справочнике TFlexDOCs
 - reference – указывает создание параметра в справочнике ЭСИ.

Файл данных

Комплект поставки содержит базовые настройки для получения данных из любой системы в predetermined xml формате. (xml схема формата указана в [Приложении 2.](#))

Описание работы приложения

- WEB – сервис получает данные от сторонней системы.
- В справочнике “Шина данных” создается пакет с набором входящих сообщений содержащих файлы метаданных (XML - по умолчанию) или файлы данных произвольного формата
- По заданному расписанию запускается задача обработки пакетов

Алгоритмы обработки

Входной точкой начала работы программы является повторяющееся по расписанию системное задание, настроенное в справочнике “Задачи и оповещения” системы TFlexDOCs.

Задание создается и запускается для каждого источника данных при регистрации данных автоматически.

Задание запускает настроенные процессы обработки полученных данных из справочника Шина данных.

API модуля

Обработка данных происходит по заранее определенным алгоритмам согласно файлу описания на основании зарегистрированной dll библиотеки для каждого источника данных реализующей следующий API модуля.

Класс `AbstractIncomingMessageHandler`.

Метод	Описание	Входные параметры (Аргументы)	Возвращаемое значение
HandleFiles	обработка xml-файлов сообщения определенного типа, используется в файле	files : IEnumerable<FileObject>	void

	конфигурации для обработки определенного типа входящих сообщений		
--	------------------------------------------------------------------	--	--

Класс `AbstractTypeMappingAlgorithm`.

Метод	Описание	Входные параметры (Аргументы)	Возвращаемое значение
<code>Execute</code>	логика обработки типа объекта другой системы, используется в файле конфигурации для обработки типа данных исходной системы	<code>xmlNode : XElement</code>	<code>void</code>

Класс `AbstractAttMappingAlgorithm`.

Метод	Описание	Входные параметры (Аргументы)	Возвращаемое значение
<code>Execute</code>	логика обработка атрибута объекта другой системы, используется в файле конфигурации для определения алгоритма обработки атрибута исходной системы	<code>mappingContext : MappingContext</code>	<code>void</code>

Добавление новой исходной системы для интеграции

1. Определите:

- a. код источника данных (`NEW_SYSTEM_CODE`),
- b. набор исходных типов входящих сообщений (`MessageType1, MessageType2...`)
- c. формат и тип файла метаданных

2. Создайте dll библиотеку для нового источника данных используя шаблон `DataBusSourceTemplate`:

- a. Определите обработчики файлов метаданных для каждого типа входящего сообщения переопределив класс `AbstractIncomingMessageHandler` для каждого типа входящего сообщения.
- b. Создайте классы обработчики для каждого типа исходной системы, указанного в файле метаданных переопределив класс `AbstractTypeMappingAlgorithm`
- c. При необходимости определите новые алгоритмы обработки атрибутов переопределив класс `AbstractAttMappingAlgorithm`
- d. Сохраните файл библиотеки в справочник Файлы

3. Настройка конфигурации:

- a. На основании базового файла конфигурации (`DefaultDataSource.Config`) создайте новый XML файл конфигурации сопоставления.
- b. Добавьте наименования классов обработчиков, созданные в п. 2.

```
<MessageHandlers>
...
  <MessageHandler MessageType="MessageType1"
Algorithm="NewSystemSource.Message.Handlers.NewMetadataIncomingMessageHandlerType1"/>
  <MessageHandler MessageType="MessageType2" Algorithm="
NewSystemSource.Message.Handlers.NewMetadataIncomingMessageHandlerType2"/>
...
</MessageHandlers>
```

- c. Добавьте описание сопоставления типов и атрибутов для xml файла метаданных, укажите алгоритм обработки типов из п.2. или используйте алгоритмы по умолчанию.

```
<TypesMap>
...
  <TypeMap Id="typeMapId" SrcType="SrcObjectType1" DocsType="" Algorithm="
NewSystemSource.Mapping.Type.Type1MappingAlgorithm">
...
    <AttMap SrcId="SrcTypeName" SrcName="Наименование"
DocsParam="7e115f38-f446-40ce-8301-9b211e6ce5fd" DocsName="Наименование"
Reference="Документы" Required="true" Type="string"
Algorithm="DataBus.Mapping.Attribute.DefaultAttMappingAlgorithm"/>
    <AttMap SrcId=" SrcTypeNameNumber" SrcName="Обозначение"
DocsParam="b8992281-a2c3-42dc-81ac-884f252bd062" DocsName="Обозначение"
Reference="Документы" Required="true" Type="string"
Algorithm="DataBus.Mapping.Attribute.DefaultAttMappingAlgorithm"/>
...
  </TypeMap>
...
</TypesMap>
```

При необходимости измените другие параметры конфигурации. Сохраните файл в TFlexDOCs.

4. Зарегистрируйте новый источник данных см. *Рисунок 6*.

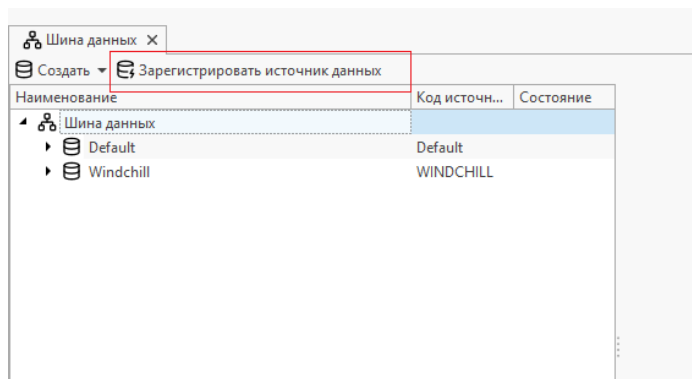


Рисунок 25 Регистрация источника данных

Укажите отображаемое наименование, код источника на латинице, файл конфигурации, созданный в п.3. и библиотеку обработчиков, созданную в п.2. см. Рисунок 7

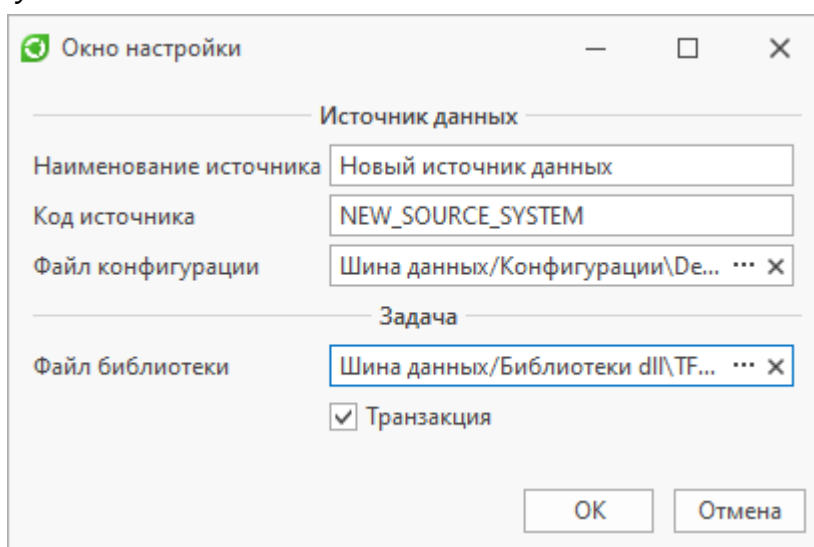


Рисунок 26 Параметры источника данных

В результате завершения работы мастера будет создан новый источник данных, в макрос «Шина данных.Задачи» добавится новый метод обработки и создастся новая задача по обработке полученных пакетов в справочнике «Задачи и оповещения».

Приложения

Приложение 1: XSD схема конфигурационного XML файла.

```
<?xml version="1.0" encoding="utf-8"?>
<xs:schema attributeFormDefault="unqualified" elementFormDefault="qualified"
xmlns:xs="http://www.w3.org/2001/XMLSchema">
  <xs:element name="config">
    <xs:complexType>
      <xs:sequence>
        <xs:element name="stages">
          <xs:complexType>
            <xs:sequence>
              <xs:element maxOccurs="unbounded" name="stageMap">
                <xs:complexType>
                  <xs:attribute name="srcState" type="xs:string" use="required" />
                  <xs:attribute name="docsStageGuid" type="xs:string" use="required" />
                </xs:complexType>
              </xs:element>
            </xs:sequence>
          </xs:complexType>
        </xs:element>
      </xs:sequence>
    </xs:complexType>
  </xs:element>
</xs:schema>
```

```

        </xs:element>
    </xs:sequence>
</xs:complexType>
</xs:element>
<xs:element name="configSpec">
    <xs:complexType>
        <xs:sequence>
            <xs:element maxOccurs="unbounded" name="configSpecMap">
                <xs:complexType>
                    <xs:attribute name="srcState" type="xs:string" use="required" />
                    <xs:attribute name="docsStageGuid" type="xs:string" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element name="batchPreprocessors">
    <xs:complexType>
        <xs:sequence>
            <xs:element name="preprocessor">
                <xs:complexType>
                    <xs:attribute name="clazz" type="xs:string" use="required" />
                </xs:complexType>
            </xs:element>
        </xs:sequence>
    </xs:complexType>
</xs:element>
<xs:element maxOccurs="unbounded" name="mapRule">
    <xs:complexType>
        <xs:sequence>
            <xs:choice maxOccurs="unbounded">
                <xs:element maxOccurs="unbounded" name="ruleAtt">
                    <xs:complexType>
                        <xs:attribute name="srcId" type="xs:string" use="required" />
                        <xs:attribute name="docsId" type="xs:string" use="required" />
                        <xs:attribute name="type" type="xs:string" use="required" />
                        <xs:attribute name="reference" type="xs:string" use="optional" />
                    </xs:complexType>
                </xs:element>
                <xs:element name="ruleAttMap">
                    <xs:complexType>
                        <xs:sequence>
                            <xs:element maxOccurs="unbounded" name="ruleAtt">
                                <xs:complexType>
                                    <xs:attribute name="srcId" type="xs:string" use="required" />
                                    <xs:attribute name="docsId" type="xs:string" use="required" />
                                    <xs:attribute name="type" type="xs:string" use="required" />
                                </xs:complexType>
                            </xs:element>
                        </xs:sequence>
                        <xs:attribute name="mapAlgorithm" type="xs:string" use="required" />
                    </xs:complexType>
                </xs:element>
            </xs:choice>
        </xs:sequence>
        <xs:attribute name="id" type="xs:string" use="required" />
        <xs:attribute name="srcType" type="xs:string" use="required" />
        <xs:attribute name="docsType" type="xs:string" use="required" />
        <xs:attribute name="mapAlgorithm" type="xs:string" use="optional" />
    </xs:complexType>
</xs:element>
</xs:sequence>
</xs:complexType>
</xs:element>
</xs:schema>

```

Приложение 2: XSD схема xml файла входящего сообщения.

```

<?xml version="1.0" encoding="UTF-8"?>
<xs:schema xmlns:xs="http://www.w3.org/2001/XMLSchema"
    elementFormDefault="qualified">
    <!-- Корневой элемент -->
    <xs:element name="data">
        <xs:complexType>
            <xs:sequence>
                <!-- objects -->
                <xs:element name="objects">

```

```

<xs:complexType>
  <xs:sequence>
    <xs:element name="object" minOccurs="0" maxOccurs="unbounded">
      <xs:complexType>
        <xs:sequence>
          <!-- attributes -->
          <xs:element name="attributes">
            <xs:complexType>
              <xs:sequence>
                <xs:element name="attr"
                  minOccurs="0"
                  maxOccurs="unbounded">
                  <xs:complexType>
                    <xs:attribute name="name"
                      type="xs:string"
                      use="required"/>
                    <xs:attribute name="value"
                      type="xs:string"
                      use="required"/>
                  </xs:complexType>
                </xs:element>
              </xs:sequence>
            </xs:complexType>
          </xs:element>
        </xs:sequence>
        <!-- object attributes -->
        <xs:attribute name="type"
          type="xs:string"
          use="required"/>
        <xs:attribute name="srcType"
          type="xs:string"
          use="required"/>
        <xs:attribute name="srcIdFile"
          type="xs:string"
          use="required"/>
        <xs:attribute name="srcId"
          type="xs:string"
          use="required"/>
        <xs:attribute name="srcMasterId"
          type="xs:string"
          use="required"/>
      </xs:complexType>
    </xs:element>
  </xs:sequence>
</xs:complexType>
</xs:element>
<!-- links -->
<xs:element name="links">
  <xs:complexType>
    <xs:sequence>
      <xs:element name="link" minOccurs="0" maxOccurs="unbounded">
        <xs:complexType>
          <xs:sequence>
            <!-- attributes -->
            <xs:element name="attributes">
              <xs:complexType>
                <xs:sequence>
                  <xs:element name="attr"
                    minOccurs="0"
                    maxOccurs="unbounded">
                    <xs:complexType>
                      <xs:attribute name="name"
                        type="xs:string"
                        use="required"/>
                      <xs:attribute name="value"
                        type="xs:string"
                        use="required"/>
                    </xs:complexType>
                  </xs:element>
                </xs:sequence>
              </xs:complexType>
            </xs:element>
          </xs:sequence>
        </xs:complexType>
      </xs:element>
    </xs:sequence>
  </xs:complexType>
</xs:element>

```

```
        </xs:complexType>
      </xs:element>

    </xs:sequence>

    <!-- link attributes -->
    <xs:attribute name="parentId"
      type="xs:string"
      use="required"/>

    <xs:attribute name="childId"
      type="xs:string"
      use="required"/>

    <xs:attribute name="srcType"
      type="xs:string"
      use="required"/>

    </xs:complexType>
  </xs:element>

  </xs:sequence>
</xs:complexType>
</xs:element>

  </xs:sequence>
</xs:complexType>
</xs:element>

</xs:sequence>
</xs:complexType>
</xs:element>

</xs:schema>
```