

Утверждён РФ.DIS.00002-01 92 01-ЛУ

# Plus7 FormIT

РУКОВОДСТВО ПОЛЬЗОВАТЕЛЯ  
РФ.DIS.00002-01 92 01

Версия программы 1.2.1

Москва

2022

## ПРЕДИСЛОВИЕ

Plus7 FormIT - проверенное решение для управления исходящими коммуникациями организации с заказчиками, включая создание и распространение документов, направляемых в адрес клиентов, управления формированием выписок, отчётов, счетов, тарифов, рекламных рассылок, маркетинговых предложений.

Plus7 FormIT позволяет взаимодействовать по всем возможным каналам взаимодействия, включая обычную почтовую службу, электронную почту, телефонную связь, факс, SMS, социальные сети.

Модуль интеграции данных позволяет подключать широкий набор источников данных для последующего наполнения формируемых документов информацией.

Plus7 FormIT идеально подходит крупным банкам, телекоммуникационным компаниям, авиакомпаниям и другим предприятиям и организациям, имеющим распределённую сеть филиалов и представительств.

## СОДЕРЖАНИЕ

ПРЕДИСЛОВИЕ.....	2
ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ .....	4
ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ .....	5
1 НАЗНАЧЕНИЕ ПРОГРАММЫ.....	11
2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ.....	12
2.1 Аппаратные средства.....	12
2.2 Программные средства .....	12
3 НАЧАЛО РАБОТЫ.....	13
4 Инструменты разработчика .....	16
5 Типы полей.....	17
6 Модель состояния шаблона документа .....	18
7 Создание шаблона документа.....	19
8 Создание страниц .....	20
9 Создание контейнера и добавление поля в контейнер.....	21
10 Определение физических и логических страниц шаблона.....	25
11 Загрузка и сохранение.....	27
12 Получение шаблона документа .....	29
13 Работа с документами Word/PDF .....	30
14 Работа с абзацами .....	32
15 Предварительный просмотр.....	34
16 Сборка.....	35
17 Создание дерева сборки.....	36
18 Сохранение.....	37
19 Использование объектов .....	39
20 Модуль управления правилами.....	40
21 Создание правил.....	41
22 Модуль архивации.....	42
22.1 Элементы документа.....	42
22.2 Извлечение документов из архива .....	42
22.3 Таблицы архивов .....	42
22.4 Добавление запроса.....	42

## ОБОЗНАЧЕНИЯ И СОКРАЩЕНИЯ

В настоящем документе применяют следующие сокращения и обозначения:

Сокращение	Определение
АРМ	- автоматизированное рабочее место
БД	- база данных
ПО	- программное обеспечение
ДВ	- хранилище форматов

## ТЕРМИНЫ И ОПРЕДЕЛЕНИЯ

В документе использованы следующие термины с соответствующими определениями:

Термин	Определение
Вызов процедуры	Ссылка на процедуру с целью ее выполнения
Выражение	Языковая конструкция, определяющая некоторое значение в соответствии со значениями одного или более операндов
Динамическое свойство объекта	Свойство объекта, определяемое или доопределяемое в период выполнения программы
Идентификатор	Литерная цепочка, выступающая в определенном контексте в роли символа. Примечание - В обработке данных рассматривают идентификаторы порций данных, областей памяти, пространств памяти, томов и др., при этом они могут выражать определенные свойства этих объектов
Индексация (в языках программирования)	Механизм для доступа к компоненте массива данных посредством ссылки на массив и посредством одного или более выражений, значения которых определяют позицию компоненты массива
Область действия объявления	Часть текста программы, на который распространяется действие некоторого объявления объекта
Область памяти	Память, выделенная для размещения одной или нескольких порций данных
Объявление объекта	Языковая конструкция для создания языкового объекта
Оператор	Пользователь программы Plus7 FormIT
Пакет заданий	Последовательность предложений языка управления заданиями, представляющая последовательность заданий системе обработки информации и вводимая вместе с данными самих заданий одним процессом системного ввода
Пакетное задание	Задание системе обработки информации, выполняемое в режиме пакетной обработки
Параллельные процессы	Процессы обработки данных, у которых интервалы времени выполнения перекрываются за счет использования различных ресурсов одной и той же системы
Переменная (в языках программирования)	Языковой объект, который может принимать различные значения
Подпрограмма	Программа, являющаяся частью другой программы и удовлетворяющая требованиям языка программирования к структуре программы
Порция данных	Данные, представленные как целое в конкретном контексте их описания или обработки и неразрывно связанные со своим носителем. Примечание - Контексты существенно зависят от решаемых задач и этапов их решения и могут изменяться от задачи к задаче и от одного этапа к другому
Последовательность выполнения в программе	Порядок выполнения инструкций программы в процессе ее выполнения
Прикладная программа	Программа, предназначенная для решения задачи или класса задач в определенной области применения системы обработки информации
Присваивание	Механизм придания значения переменной в языке программирования

Термин	Определение
Программа	Данные, предназначенные для управления конкретными компонентами системы обработки информации в целях реализации определенного алгоритма
Программирование	Научная и практическая деятельность по созданию программ
Программное обеспечение	Совокупность программ системы обработки информации и программных документов, необходимых для эксплуатации этих программ
Программный блок	Синтаксически определенное составное предложение, образующее область действия объявленных в нем объектов
Программный модуль	Программа или функционально завершенный фрагмент программы, предназначенный для хранения, трансляции, объединения с другими программными модулями и загрузки в оперативную память
Процедура (в языках программирования)	Параметризуемый именованный программный блок, конкретное выполнение которого определяется вызовом процедуры
Процесс обработки данных	Система действий, реализующая определенную функцию в системе обработки информации и оформленная так, что управляющая программа данной системы может перераспределять ресурсы этой системы в целях обеспечения мультипрограммирования. Примечания: 1. Процесс характеризуется состояниями, которые определяются наличием тех или иных ресурсов в распоряжении процесса и, следовательно, возможностью фактически выполнять действия, относящиеся к процессу. 2. Перераспределение ресурсов, выполняемое управляющей программой, влияет на продолжительность процесса обработки данных, но не на его конечный результат. 3. Процесс оформляют с помощью специальных структур управляющих данных, которыми манипулирует управляющий механизм. 4. В конкретных системах обработки информации встречаются разновидности процессов, которые различаются способом оформления и составом ресурсов, назначаемых процессу и отнимаемых от него, и допускается вводить специальные названия для таких разновидностей, например задача в операционной системе ОС ЕС ЭВМ
Сеанс работы	Интервал времени, в течение которого пользователю выделены ресурсы системы обработки информации для выполнения его работ и связи с ним в интерактивном режиме
Составное предложение	Предложение, состоящее из последовательности предложений
Ссылка (в языках программирования)	Объект при его использовании в качестве имени
Статическое свойство объекта	Свойство объекта, определяемое до выполнения программы
Структурное программирование	Метод построения программ, использующий только иерархически вложенные конструкции, каждая из которых имеет единственную точку входа и единственную точку выхода. Примечание - В структурном программировании используются три вида структур, связанных с передачей управления: последовательная, условного перехода и циклическая
Тип данных	Множество значений вместе с множеством допустимых над ними операций

Термин	Определение
Трансляция программы	Преобразование программы, представленной на одном языке программирования, в программу на другом языке и в определенном смысле равносильную первой
Физический объект	Объект, рассматриваемый в аспекте взаимодействия логического объекта с техническими средствами
Цикл (в программе)	Последовательность команд в программе, которая должна исполняться неоднократно в результате перехода от начала последовательности к концу
Язык высокого уровня	Язык программирования, понятия и структура которого удобны для восприятия человеком
Язык программирования	Язык, предназначенный для представления программ. Примечание - К традиционным языкам программирования процедурного типа относят, как правило, языки для представления программ в виде последовательности предписания
Язык управления заданиями	Язык, предназначенный для идентификации заданий и описания их характеристик при передаче задания на выполнение в систему обработки информации. Примечание - Система обработки информации обычно имеет свой язык управления заданиями
Archive	Модуль для хранения отправленных документов в системе Plus7 FormIT Project для последующего извлечения
ASCII file	Выходной файл метаданных Plus7 FormIT, очищенный от тегов и свойств стиля в кодировке ASCII. ASCII - стандарт кодировки для обмена информацией.
Barcode	Поле контейнера для отображения штрих-кода
Batch Document	Документ, созданный процессом по заданному расписанию
Boolean Expression (Булево выражение)	Выражение, результатом которого является булево значение, т.е. TRUE или FALSE
Button	Поле контейнера для вызова определенного события (например, отправка значений из формы на веб-сервер)
Calculate	Специфичное свойство контейнера, указывающее, следует ли рассчитывать местоположение контейнера относительно другого контейнера
Checkbox	Поле контейнера, содержащее значение True/False
Collection	Процесс, содержащий извлечение данных документа из исходной системы в динамические таблицы данных и сохранение метаданных документа в хранилище
Collection Tree	Иерархическое представление тегов и полей документа, порядок которых учитывает расчет относительного расположения и динамическую вставку объектов, рассчитанных по правилам
CollectIT	Инструмент, используемый для построения иерархического дерева представления тегов, полей и определения их свойств
Combobox	Поле контейнера, содержащее набор значений, из которых пользователь должен выбрать одно
ComposeIT	Инструмент, используемый для создания шаблона метаданных документа
Container	Объект ComposeIT (аналогичный Tag в CollectIT) для объединения логически связанных объектов Container Field

Термин	Определение
Container Field	Объект ComposeIT (аналогичный Field в CollectIT) для идентификации типа элемента данных (статическое поле, поле динамического изображения и т.д.) и определения специфических и стилевых свойств объекта
Dataflow	Последовательность процессов извлечения, преобразованию и передачи данных
DateBox	Поле контейнера для приема строки в формате даты
Dependency	Специфическое свойство контейнера для определения положения контейнера на странице
Designer	Модуль конструирования документа (Теги, поля и т.д.)
Document	Выходной документ, основанный на данных из CollectIT и оформленный в соответствии с ComposeIT
Dynamic Field	Поле контейнера, заполняемое текстом, который может быть разным для каждого экземпляра документа (например, данные о начислениях клиента, адрес клиента)
Dynamic Image	Поле контейнера, заполняемое объектом изображения на основе правил для группы экземпляров документа (например, сегментированное рекламное изображение)
Dynamic Text	Поле контейнера, заполняемое текстовым объектом на основе правил для группы экземпляров документа (например, текстовые сегментированные сообщения)
Engine	Модуль выполнения процессов проекта и связанных систем-источников
Environment	Модуль создания рабочего пространства проекта
Extraction	Процесс объединения таблиц динамических данных и метаданных в XML-строку для записи в архивные таблицы
Field	Элемент данных
Fit_rep_xx	Текущая версия репозитория
Flat file	Текстовый файл содержащий информацию в кодировке ASCII
HTML	Язык разметки гипертекста для создания веб-контента
Insert	Специальная логическая страница документа для вставки в различные места документа в соответствии с логическими правилами, реализованных в инструменте Pagination Tool
IVR	Интерактивный голосовой помощник, возможный канал вывода документов Plus7 FormIT
Logical Page	Описание страницы, которое определяет логическую структуру данных в Документе и содержит однотипные данные
Mapping	Последовательность правил преобразования данных
Mapping Designer	Инструмент создания, редактирования правил преобразования и их последовательностей.
Mapplet	Простое правило преобразования данных с набором трансформаций, которое может быть многократно использовано в последовательностях правил преобразования данных.
Online Document	Процесс создания документа в реальном времени.
ORM (Object Rule Manager)	Инструмент управления правилами и объектами для заполнения различных полей документа
Page Break	Инструмент нумерации страниц и разделения страниц на основе тегов
Page Type	Инструмент описания типа содержимого физической страницы
Page Type Group	Группа типов страниц включает специальный тег для данных конечной группы



Термин	Определение
Pagination	Процесс формирования документа (как определено в CollectIT) в шаблон документа (как определено в ComposeIT)
Paragraph	Инструмент определения больших текстовых данных
Physical Page	Объект, определяющий специфические свойства размера и полей физической страницы
Plus7 FormIT	Комплексное решение для централизованного проектирования, разработки и выпуска большого объема сложноструктурированных документов
Plus7 FormIT Manager	Графический интерфейс для разработки проекта с использованием инструментов Plus7 FormIT
Popup	Поле контейнера (только в HTML) для отображения данных в различных окнах
Project	Набор определений связанного набора документов, которые будут созданы с помощью Plus7 FormIT
Repository	Модуль для хранения таблиц метаданных документов
Repository Manager	Графический интерфейс для просмотра и манипулирования метаданными проекта
Rule	Правило бизнес-логики, определяемое булевым выражением Rule Source для выбора объекта динамического изображения или поля динамического текста заполняемого поля
Rule Condition	Булево выражение для выбора объекта для заполнения поля динамического изображения или поля динамического текста (правило с несколькими элементами позволяет выбирать несколько объектов для приоритетного заполнения одного поля)
Rule Source	Поле объекта CollectIT, используемое в качестве переменной булева выражения для тестирования соответствия бизнес-логике ORM
SendIT	Модуль для рассылки документов (SMS, факс, электронная почта и т.д.)
Sorting	Процесс построения иерархии объектов проекта на основе уровней групп тегов, определенных в CollectIT
Source Analyzer	Приложение с графическим интерфейсом для импорта и редактирования определений источников
Source Qualifier	Трансформация, описывающая считываемые данные из источника
Static Field	Поле контейнера, заполняемое фиксированным текстовым элементом (например, итоговая сумма начислений)
Static Image	Поле контейнера заполняемое фиксированным элементом изображения (например, логотипом поставщика услуг)
Tag	Объект CollectIT (аналогичный Container в ComposeIT) для объединения логически связанных полей и определения иерархии объектов для целей сортировки
Tag Data Table	Таблица репозитория, заполненная метаданными документа
Target	Объект, приемник (файл или таблица) для записи данных после манипулирования ими
Task	Задание рабочего процесса, например, запуск сеанса сопоставления, установка значения для переменной рабочего процесса
Template	Уникальный макет документа, определенный в ComposeIT свойствами стиля объекта
Transformation	Объект репозитория, который генерирует, изменяет или передает данные
Uploading	Процесс, в ходе которого информация, введенная в клиенте, загружается в таблицы хранилища

Термин	Определение
URI	Универсальный относительный идентификатор веб-службы для рабочей станции Web
View IT	Инструмент, используемый для отображения документов Plus7 FormIT посредством запросов, в том числе запросов на поиск файлов в Архиве
Word Insert	Инструмент, используемый для импорта документа в формате doc/ pdf и использования его как текстовых данных
Workflow	Рабочий процесс, состоящий из заданий для пакетного выполнения на основе точек принятия решений и связей
Workflow Manager	Инструмент для создания и манипулирования рабочими процессами
Workflow Monitor	Инструмент для мониторинга рабочих процессов и задач.
Worklet	Часть рабочего процесса (подобно мапплету в маппинге), инкапсулированная и управляемая извне в целях повторного использования
XML	Расширяемый язык разметки. Используется для хранения данных и метаданных Plus7 FormIT

## 1 НАЗНАЧЕНИЕ ПРОГРАММЫ

1.1 Программа Plus7 FormIT представляет собой систему управления исходящими коммуникациями с заказчиком. Используется в качестве центрального инструмента для формирования сообщений согласно заранее подготовленным шаблонам.

1.2 Plus7 FormIT предназначена для служб по работе с заказчиками. Используется с целью создания индивидуализированных предложений, а также для информирования заказчика путём предоставления ему индивидуализированного отчёта об операциях, транзакциях, заказах и т.д.

1.3 В составе решения находятся модули подготовки шаблонов и интеграции. Все метаданные о структуре шаблонов и интеграционных процессов хранятся в репозитории.

1.4 Данное руководство предназначено для пользователей программы Plus7 FormIT. В руководстве содержится подробное описание действий пользователей, а также подробные снимки экранов с описанием всех возможностей и ограничений.

1.5 Навыки и компетенции обслуживающего персонала и пользователей программы Plus7 FormIT должны обеспечивать бесперебойную работу всех модулей программного обеспечения в режиме реального времени с технологическими перерывами на обслуживание оборудования и обновление программно-аппаратных средств. Обучение персонала производится на основе договоров поставки программного обеспечение и/или других соглашений.

1.6 Программа Plus7 FormIT разработана на языке программирования Java.

## 2 УСЛОВИЯ ВЫПОЛНЕНИЯ ПРОГРАММЫ

### 2.1 Аппаратные средства

2.1.1 Подпрограмма Plus7 FormIT Manager устанавливается на автоматизированные рабочие места операторов программы.

2.1.2 Рекомендуемые технические характеристики АРМ оператора подпрограммы Plus7 FormIT Manager:

- процессор с тактовой частотой не менее 1 ГГц;
- объём оперативной памяти не менее 8 Гб;
- объём жесткого диска не менее 200 Гб;
- клавиатура;
- монитор SVGA (графический режим должен иметь разрешение не менее 1024x768);
- USB-порт;
- 1 порт Ethernet 100Мбит/с;
- манипулятор типа мышь.

2.1.3 Рабочее место оператора должно быть подключено к сети Интернет со скоростью передачи информации по каналу связи не менее 100 Мбит/сек.

### 2.2 Программные средства

2.2.1 Программа Plus7FormIT устанавливается на следующие операционные системы:

- RedOS версии 7.3
- AstraLinux Common Edition
- Ubuntu версии 18.04/20.04
- Windows Server 2019/2016

2.2.2 Репозиторий Plus7 FormIT можно установить на базы данных:

- PostgreSQL версии 11+
- Oracle версии 21с, 19с, 18с, 12сR2

2.2.3 Рекомендуемые обозреватели:

- Google Chrome версии 100 и выше
- Mozilla Firefox версии 100 и выше
- Любой обозреватель на Chromium версии 98 и выше

### 3 НАЧАЛО РАБОТЫ

3.1 Работа с программой Plus7 FormIT начинается с введения параметров подключения и авторизации пользователя к системе (рисунок 1).

3.2 В поле Username и Password необходимо указать имя пользователя и пароль для последующей авторизации в системе.

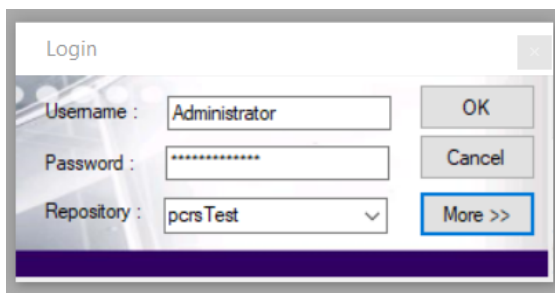
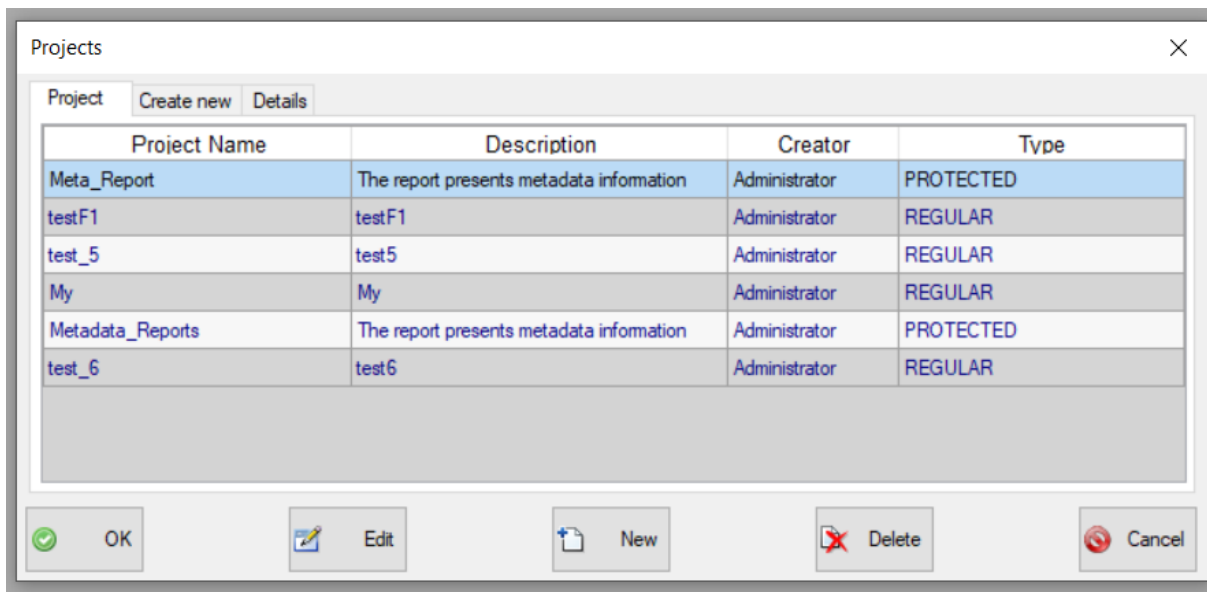


Рисунок 1 – Формы параметров подключения и авторизации

3.3 Для заполнения параметров подключения к системе обратитесь к системным администраторам.

3.4 После подключения к системе необходимо выбрать существующий проект и нажать кнопку «OK» или нажав кнопку «New» создать новый проект (рисунок 2).



Project Name	Description	Creator	Type
Meta_Report	The report presents metadata information	Administrator	PROTECTED
testF1	testF1	Administrator	REGULAR
test_5	test5	Administrator	REGULAR
My	My	Administrator	REGULAR
Metadata_Reports	The report presents metadata information	Administrator	PROTECTED
test_6	test6	Administrator	REGULAR

Рисунок 2 – Список проектов

3.5 При создании нового проекта необходимо указать имя проекта в поле «Project Name» и нажать кнопку «OK», после чего появится дополнительное окно подтверждения создания проекта, в котором необходимо подтвердить создание проекта нажав кнопку «OK» (рисунок 3).

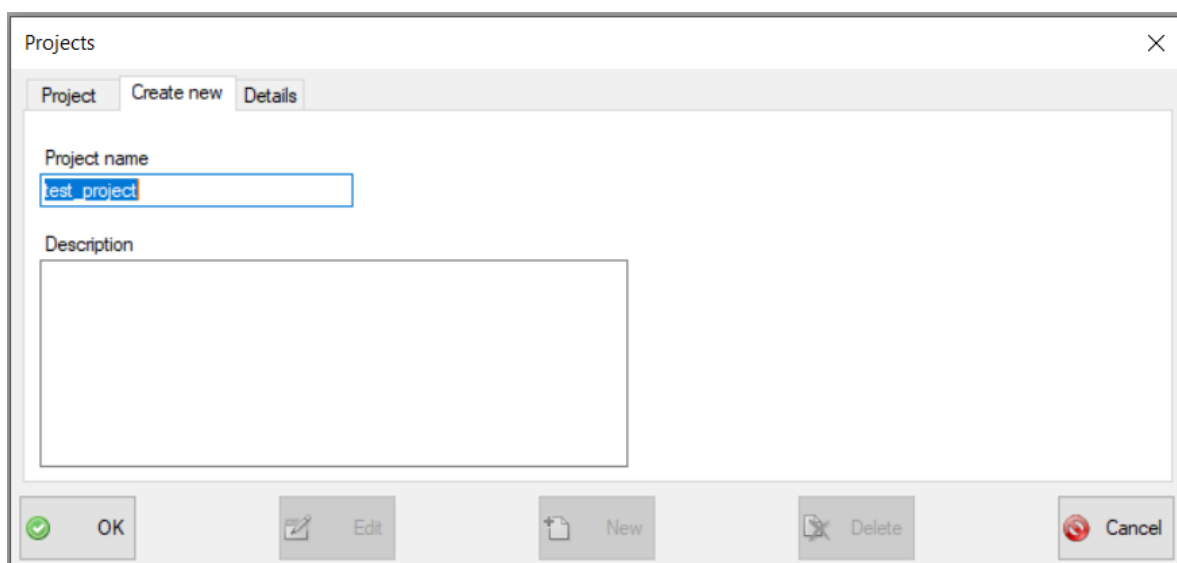


Рисунок 3 – Создание проекта

3.6 После создания проекта клиент автоматически открывает инструмент CollectIT (рисунок 4) в котором отображается базовое дерево объектов проекта. Если вы знаете структуру дерева сборки объектов проекта, то можете добавить необходимые объекты. Сохраните дерево объектов проекта нажав кнопку сохранить.

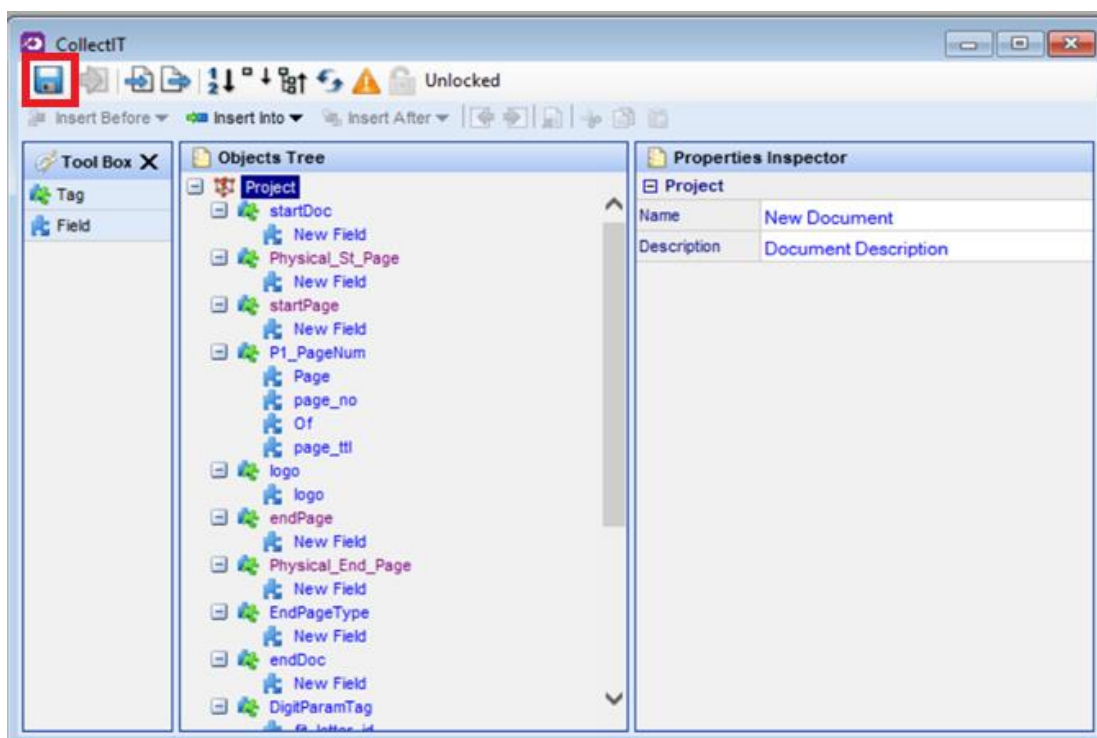


Рисунок 4 – Дерево сборки объектов проекта

3.7 В дереве объектов сборки проекта создаются теги и входящие в теги поля. В последствии при создании шаблона документа и наполнении шаблона контейнерами и полями, указывается связь тэгов в параметрах элементов шаблона, таких как поля, контейнеры, страниц и т.д. При настройке интеграционных процессов с источниками данных, теги используются в качестве идентификатора приемника данных.

3.8 После сохранения дерева объектов кнопка загрузки дерева в систему станет активной (смотри рисунок 5). Необходимо произвести загрузку дерева объектов проекта в систему нажав на данную кнопку.

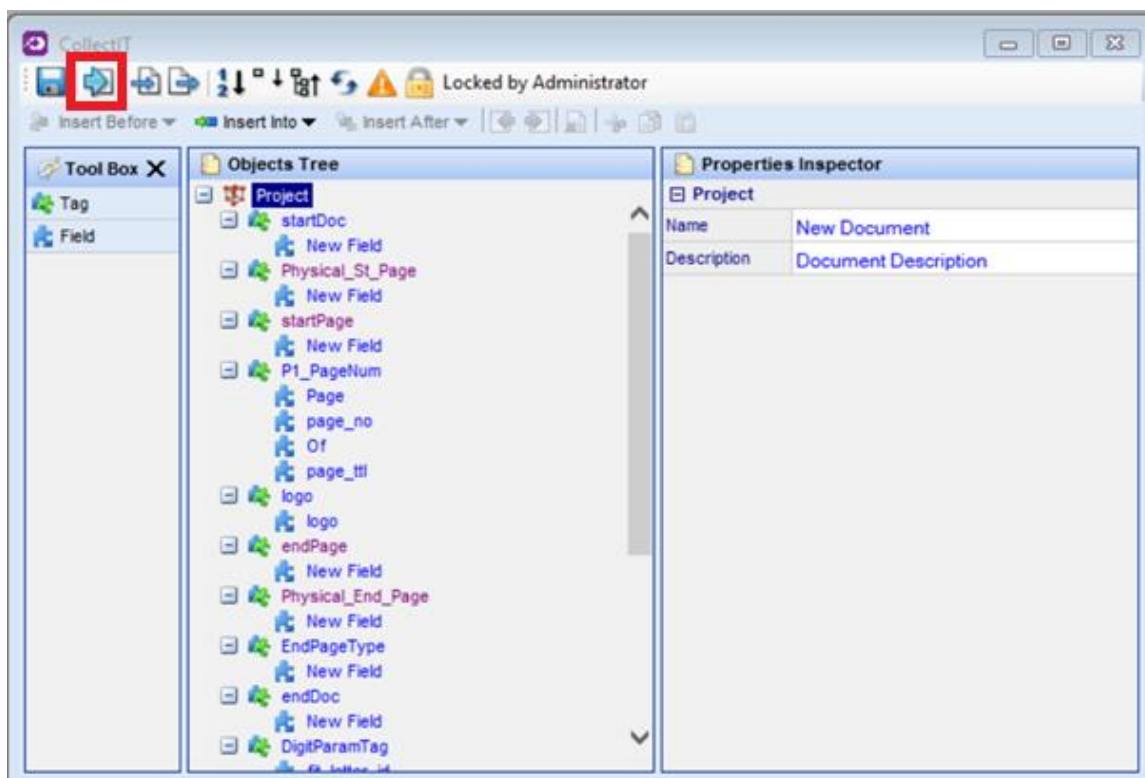


Рисунок 5 – Отображение дерева сборки объектов проекта с активной кнопкой загрузки в систему

3.9 Далее в рамках работы с проектом производится создание шаблонов документов в соответствии с процедурами, по которым будет осуществляться подготовка документов для индивидуализированной отправки. Вторым этапом выполняется загрузка интерфейсных таблиц, из которых пользователь будет получать информацию для формирования индивидуализированных документов на основании шаблонов.

## 4 Инструменты разработчика

4.1 Для разработчиков доступны инструменты разработки шаблонов и создания интеграционных процессов.

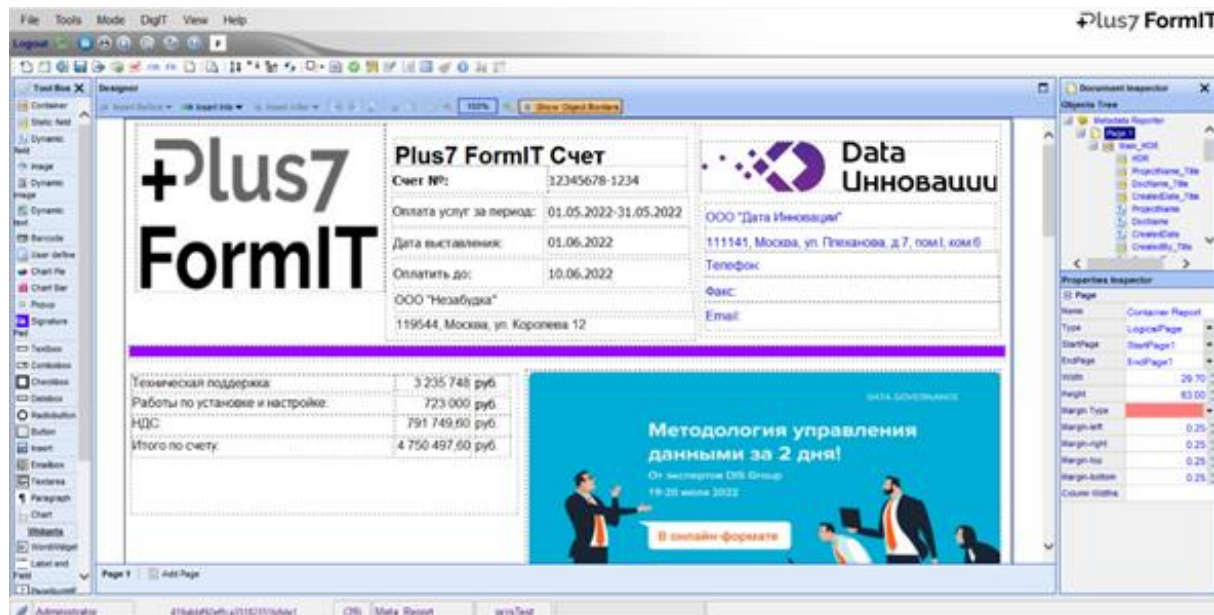


Рисунок 6 – Общий вид инструмента разработчика ComposeIT

4.2 Интерфейс разработки шаблонов представляет собой толстый клиент (см. рис. 6).

4.3 Составление документа — это процесс разработки макета документа и шаблонов документов.

4.4 Проект может содержать 1 или более шаблонов документов, созданных из одного и того же дерева коллекций.

4.5 Шаблон документа: визуальный макет элементов данных, которые должны быть представлены в документе вместе с их атрибутами дизайна и формата.

4.6 Контейнер: логическая группировка информации (статические/динамические данные и/или изображения), которая должна быть представлена в документе, содержащая атрибуты формата.

4.7 Логическая страница: отдельная страница документа, содержащая все контейнеры и поля, которые могут быть представлены на ней.

4.8 Одна логическая страница может создавать от одной до нескольких физических страниц, в зависимости от объема данных, которые обрабатываются в процессе сбора.

4.9 Логика разбивки верхних и нижних колонтитулов, разделов и страниц является частью процесса разбивки на страницы.



## 5 Типы полей

5.1 Поля (см. рис. 7) – это элементы шаблонов, которые представляют в себе информацию. Поля могут быть разных типов в зависимости от того, какая информация должна быть в них представлена (это может быть текстовая, цифровая, графическая и т.д).

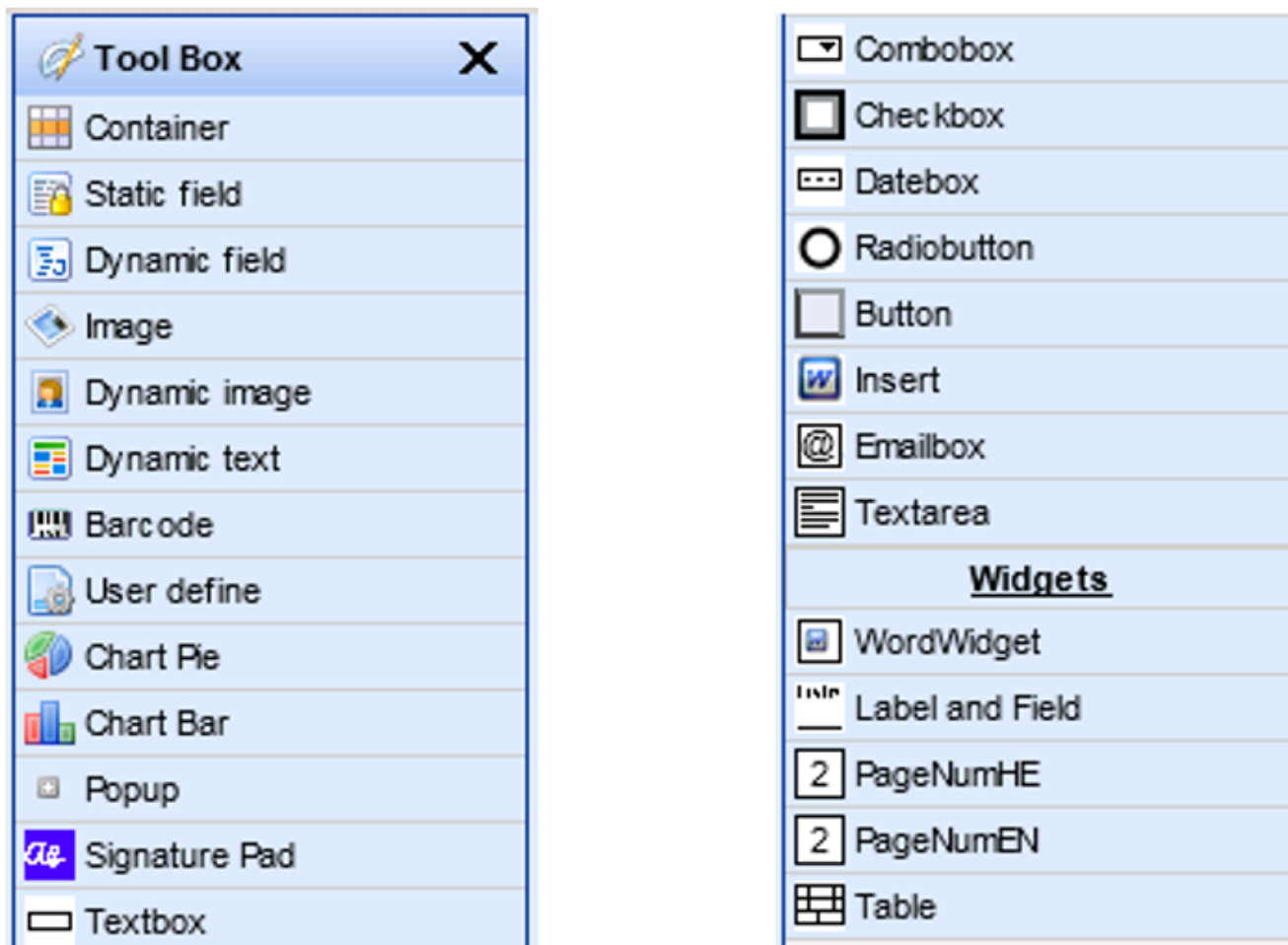


Рисунок 7 – Поля элементов

## 6 Модель состояния шаблона документа

- 6.1 Шаблон документа может находиться в трёх состояниях.
  - 6.1.1 Состояние W = Работа. В рабочем состоянии всегда есть 1 запись – при работе с ComposeIT
  - 6.1.2 Состояние P = Производство. Шаблон создается после выполнения загрузки и не может быть изменён. Данный шаблон будет использоваться для создания документов при следующем запуске. В БД может существовать только одна запись “P”.
  - 6.1.3 Состояние H = История. В данное состояние шаблон переходит после того, как была выполнена еще одна загрузка и шаблон находился в состоянии W. Шаблон не может изменяться. Одна или несколько таких записей могут существовать в базе данных.
- 6.2 Пример списка шаблонов представлен на рисунке 8.

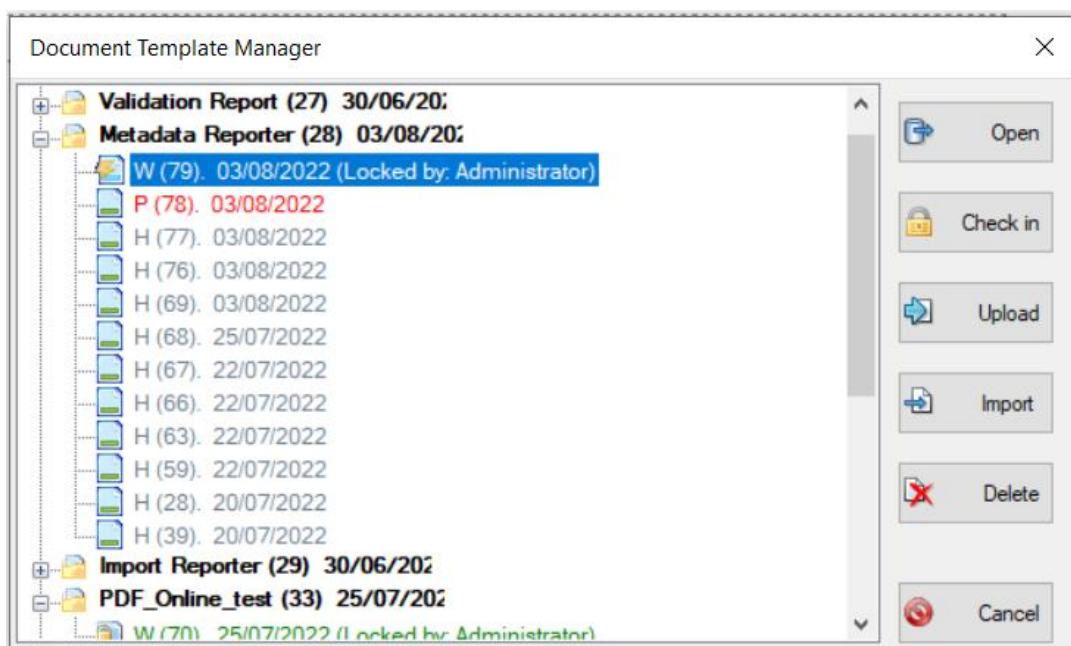


Рисунок 8 – Отображение списка шаблонов проекта со статусом

## 7 Создание шаблона документа

7.1 Шаблоны документов создаются в составной рабочей области подпрограммы Plus7 ComposeIT (Mode → Project Structure → ComposeIT (смотри рисунок 9)).

7.2 При входе в рабочую область по умолчанию используется пустой шаблон с одной страницей. Дайте шаблону осмысленное имя и описание. Назначьте теги начального документа и конечного документа в атрибутах Ссылка на начальный тег и Ссылка на Конечный тег.

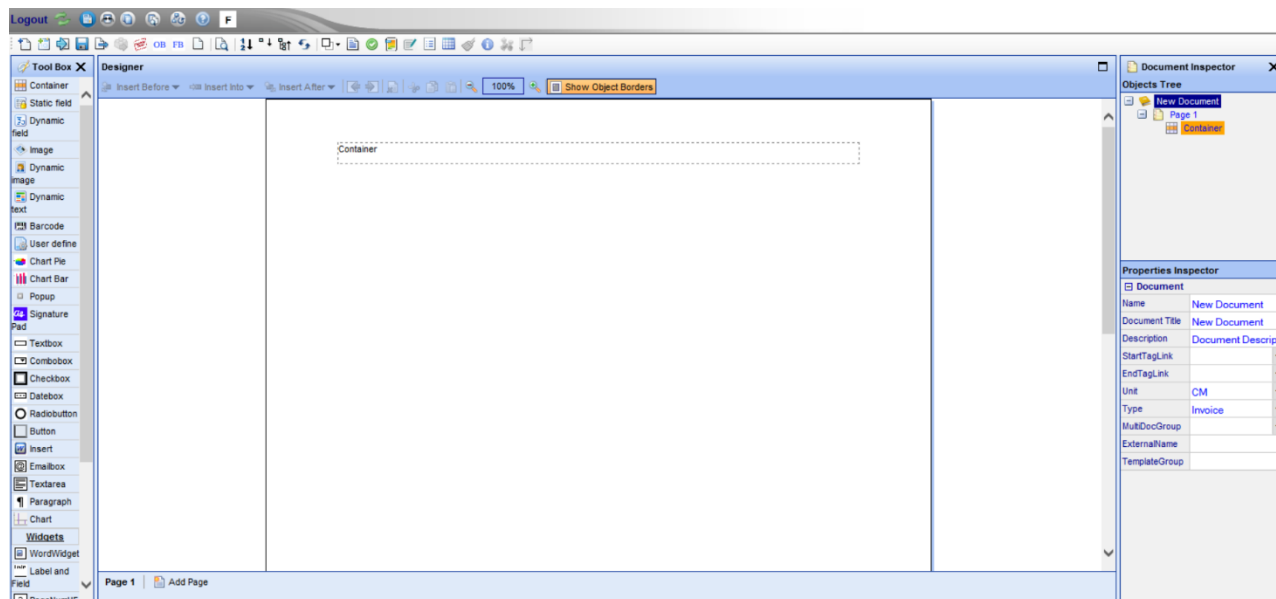


Рисунок 9 – Создание шаблона документа

## 8 Создание страниц

8.1 Документ «Страница 1» отображается по умолчанию (смотри рисунок 10). Нажмите на кнопку «Добавить страницу», чтобы добавить новую страницу. Дайте каждой странице соответствующее название. Назначьте тег страницы в атрибуте Ссылки на тег. Определите все другие атрибуты страницы форматирования по мере их применения к страницам.

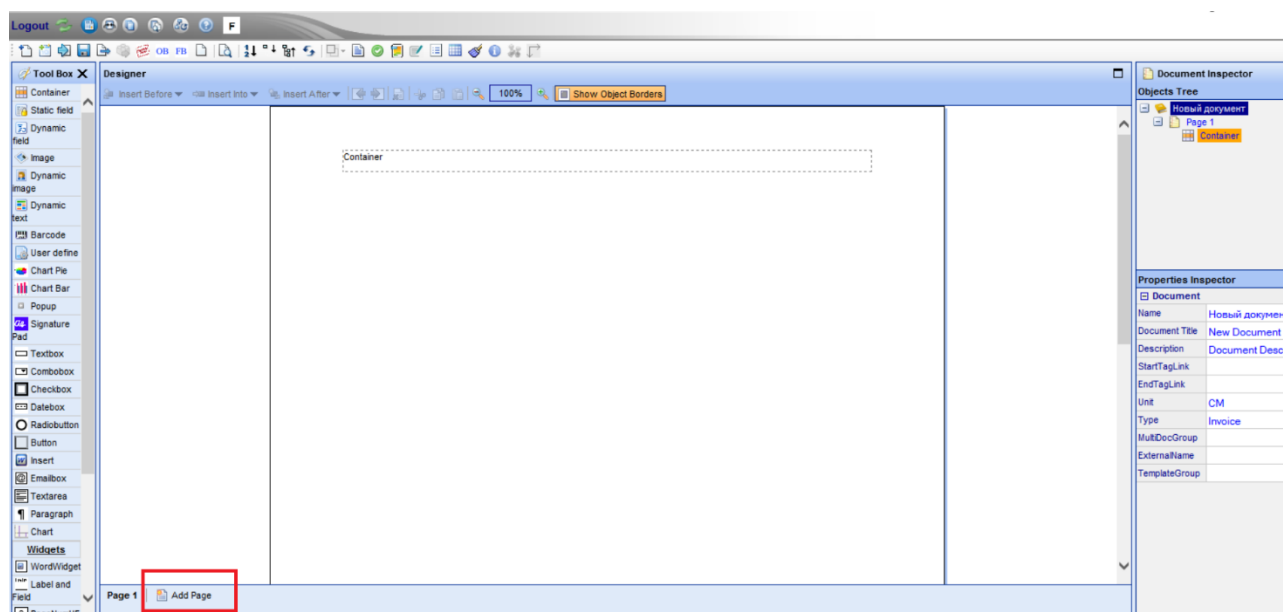


Рисунок 10 – Создание страниц документа

## 9 Создание контейнера и добавление поля в контейнер

9.1 Выберите страницу в Plus7 Document Inspector на которую будет добавляться контейнер, нажмите правую кнопку мыши и добавьте контейнер, также контейнер можно добавить, установив курсор мышки на свободное место шаблона, нажав правую кнопку мыши (смотри рисунок 11).

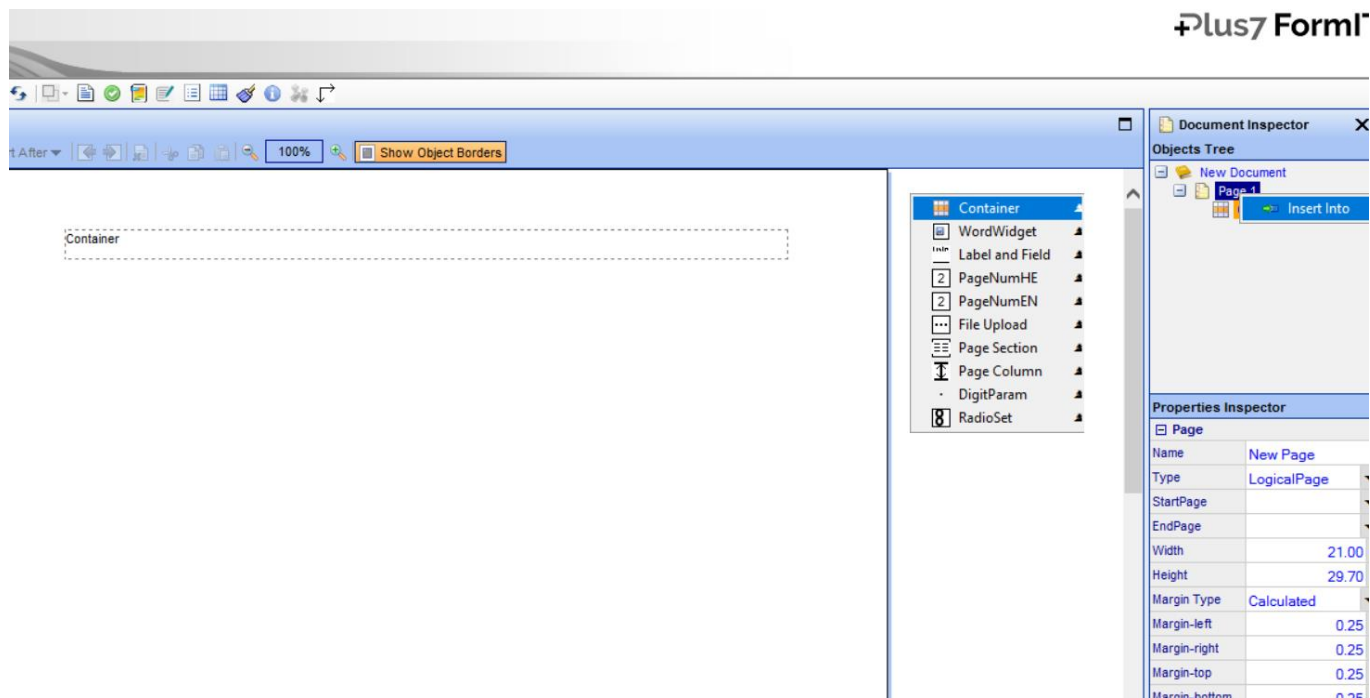


Рисунок 11 – Добавление контейнера на страницу документа

9.2 Создайте необходимое количество контейнеров и расположите их согласно предполагаемому целевому дизайну документа.

9.3 При добавлении поля в контейнер выберите контейнер, в который будет добавляться поле, нажмите правую кнопку мыши, выберите тип поля из списка и добавьте в контейнер (см. рис. 12).

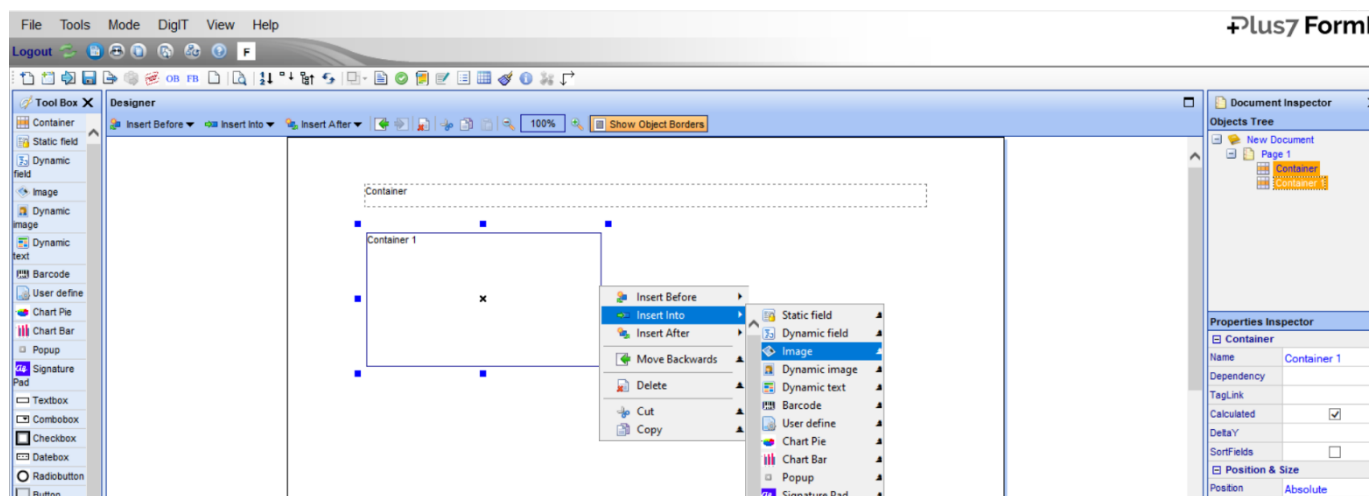


Рисунок 12 – Добавление графического поля (image) в контейнер

9.4 В свойствах поля в строке Image выберите графический элемент из доступных и нажмите кнопку Apply. Таким образом в каждый контейнер можно добавить до 20 различных полей (см. рис. 13).

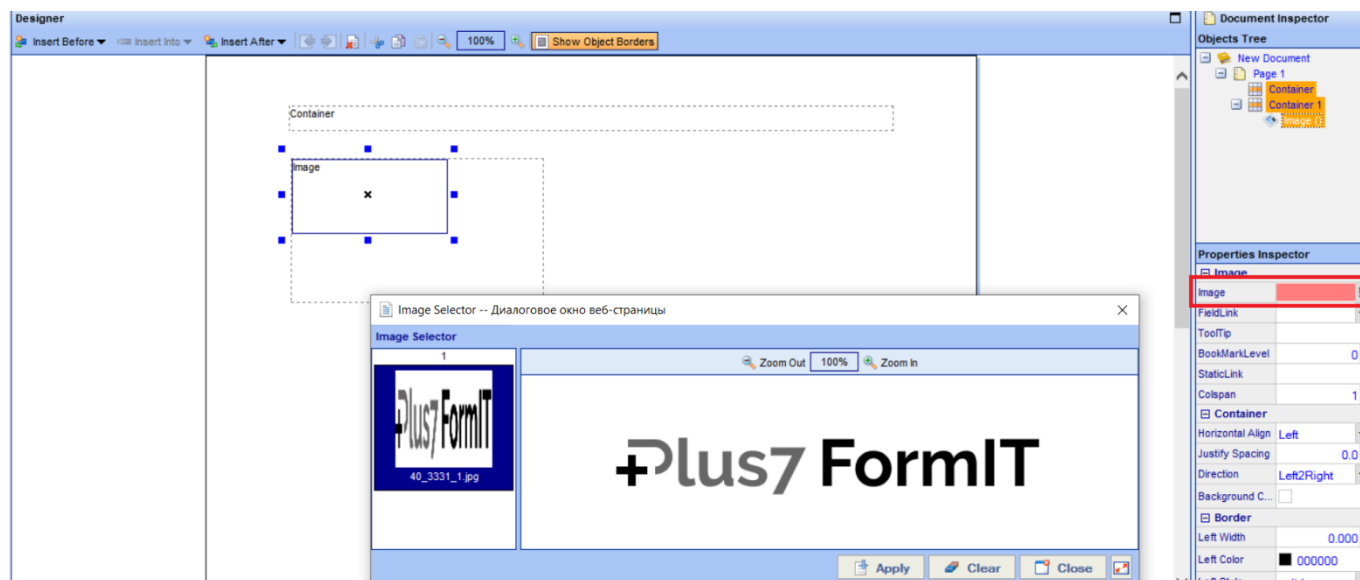


Рисунок 13 – Выбор графического объекта (image)

9.5 Далее в соответствии с планируемым дизайном шаблона документа добавьте дополнительные поля и контейнеры на страницу шаблона документа. Один контейнер может содержать не более 20 полей.

9.6 Для каждой страницы документа (Page 1, 2 и т.д.) необходимо дать название, указать тип (Type) – LogicalPage, StartPage – startPageX, EndPage – endPageX, проставить размеры логической страницы и отступы. На каждой странице необходимо оставить место для Header и Footer (если они нужны) в характеристиках Margin-top / Margin-bottom, см. рисунки 14, 15 и 16. Форматирование верхнего и нижнего колонтитулов происходит только на первой странице, а правила дублирования устанавливаются позже, в интерфейсе Pagination.

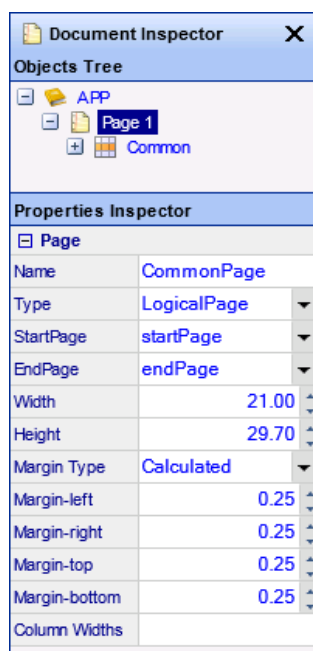


Рисунок 14 – Определение параметров страницы

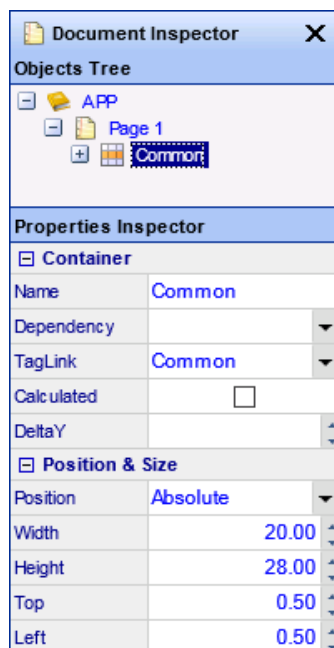


Рисунок 15 – Определение параметров контейнера

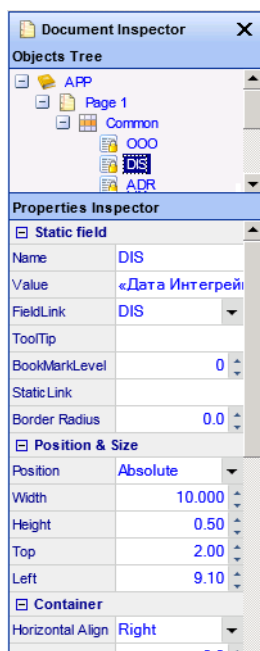


Рисунок 16 – Определение параметров поля



## 10 Определение физических и логических страниц шаблона

10.1 Необходимо открыть инструмент Pagination (Mode -> Project Structure -> Pagination).

10.2 Данная система основана на определенных индикаторных тэгах, которые предоставляют информацию о желаемом формате страницы (смотри рисунок 17). Эти тэги отмечают начало и конец: физической страницы (Physical Page), логической страницы (Logical Page), таблицы (Table) и буфера (Buffer).

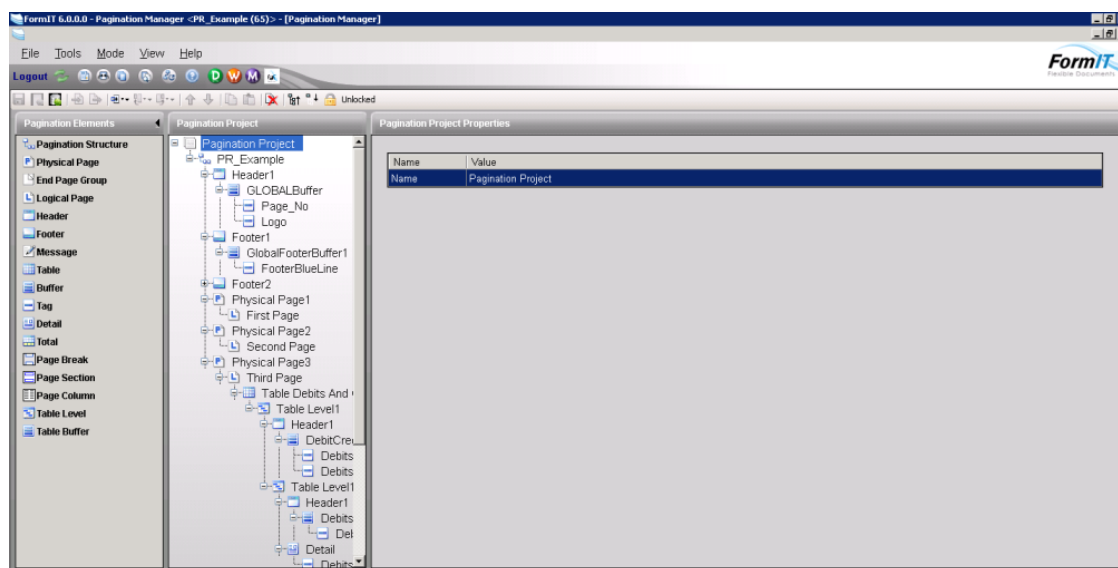


Рисунок 17 – Внешний вид инструмента Pagination

**Pagination Structure** – правила разбивки на страницы, которые ссылаются на конкретный документ.

**Physical Page** – описание физической страницы.

**Logical Page** – страница, которая была определена в ComposeIT.

**Header** – колонтитул, который появляется в начале каждой страницы.

**Footer** – колонтитул, который появляется в конце каждой страницы.

**Message** – текстовые сообщения, которые должны отображаться вместе на одной странице.

**Table** – данные, которые отображаются в табличной форме.

**Buffer** – сохраняет данные перед записью в документ. Буфер может быть вставлен в документ несколько раз. Он очищается на основе определенного тэга.

**Tag** – контейнер для данных.

**Detail** – часть таблицы, содержащая все данные.

**Total** – вычисляемые поля в таблице.

10.3 Перед тем, как создавать Pagination, необходимо, чтобы структура CollectIT была выгружена в репозиторий. Необходимо помнить, что физическая страница открывает новую страницу, в отличие от логической. Физическая страница может содержать несколько логических страниц.

10.4 Необходимо указать название документа как он указан в CollectIT, см. рисунок 18.

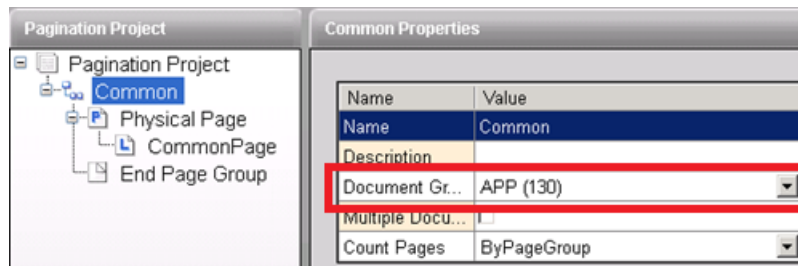


Рисунок 18 – Указание названия документа в Pagination

10.5 Выбрать тип страницы и начальный и конечный тэги, см. рисунок 19.

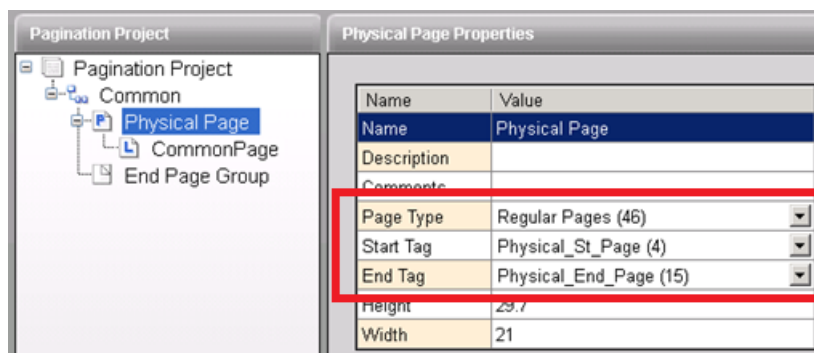


Рисунок 19 – Выбор типа страницы и тэгов в Pagination

10.6 Выбрать название страницы как указано в ComposeIT, см. рисунок 20.

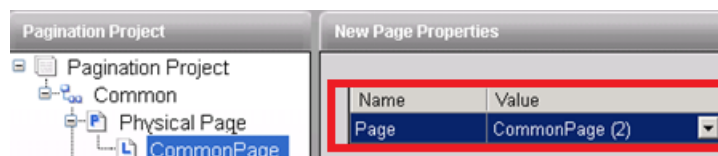


Рисунок 20 – Выбор типа страницы и тэгов в Pagination

10.7 Выбрать группу страниц из модуля Page Types, связать End Page Group с тэгом End Page Group, как определено в CollectIT, см. рисунок 21.

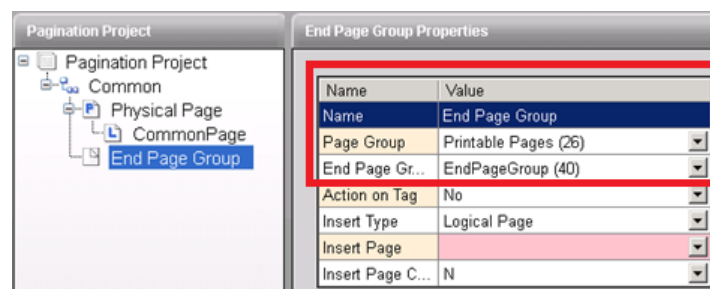


Рисунок 21 – Выбор типа страницы и тэгов в Pagination

## 11 Загрузка и сохранение

11.1 Шаблоны документов могут быть сохранены в XML-файле или в базе данных хранилища форматов. Метаданные шаблона документа сохраняются в формате XML. Как только шаблон документа будет завершен и готов к развертыванию, выполните загрузку. Разбивает XML на несколько реляционных таблиц, используемых в процессе извлечения в архив. Продвигает записи шаблона документа в базе данных репозитория на основе модели состояния:

- W (работа);
- P (производство);
- H (история).

11.2 Для сохранения документа нажмите кнопку «сохранить» (см. рис. 22)

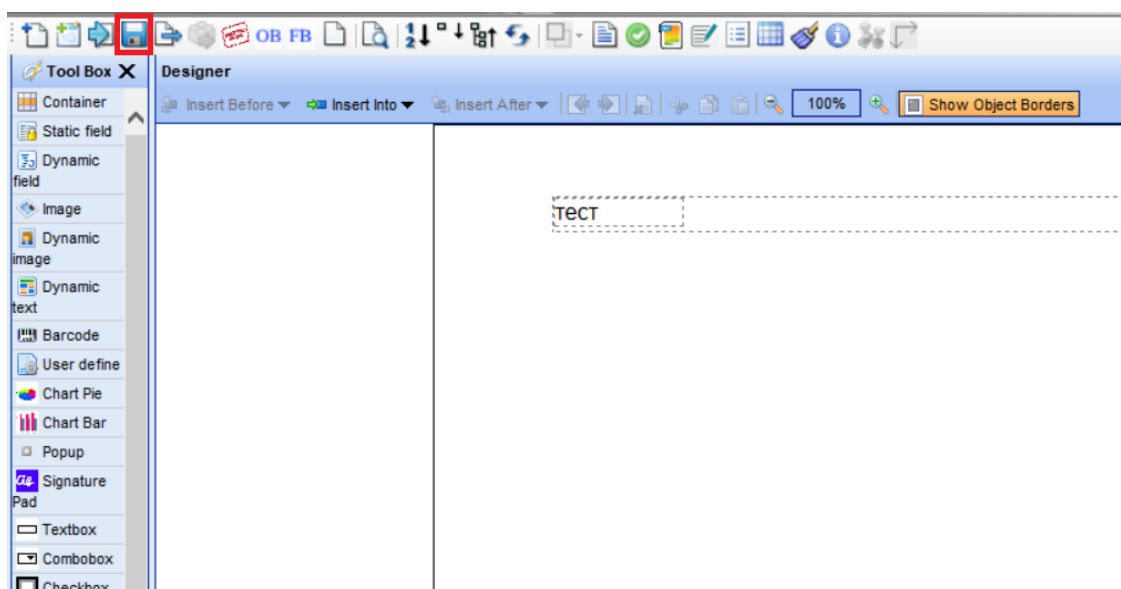


Рисунок 22 – Сохранение шаблона документа

11.3 После сохранения шаблона документа необходимо загрузить его в систему (см. рисунки 23 и 24).

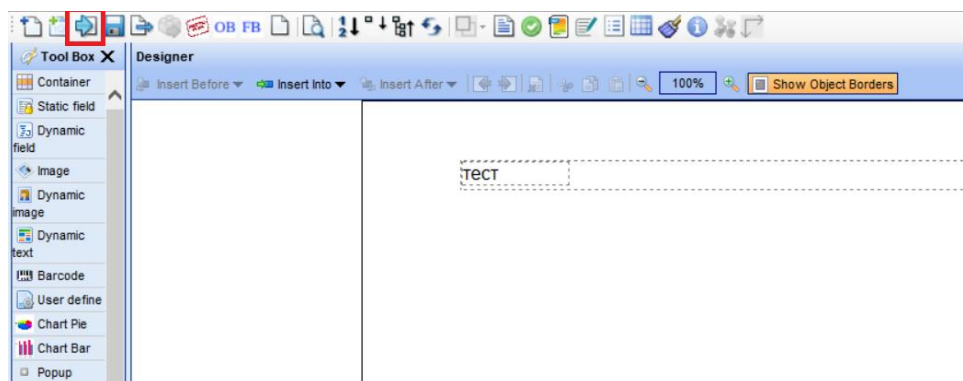


Рисунок 23 – Загрузка шаблона в систему шаг 1



Рисунок 24 – Загрузка шаблона в систему шаг 2

## 12 Получение шаблона документа

12.1 Документы извлекаются нажатием на значок. Появится список всех документов в проекте вместе с их статусом (смотри рисунок 25). Возврат заблокированного шаблона документа в состояние W. Проверьте доступный шаблон документа в состоянии W. Шаблоны документов в этом состоянии могут быть открыты только в режиме только для чтения. Импорт шаблона документа из XML-файла.

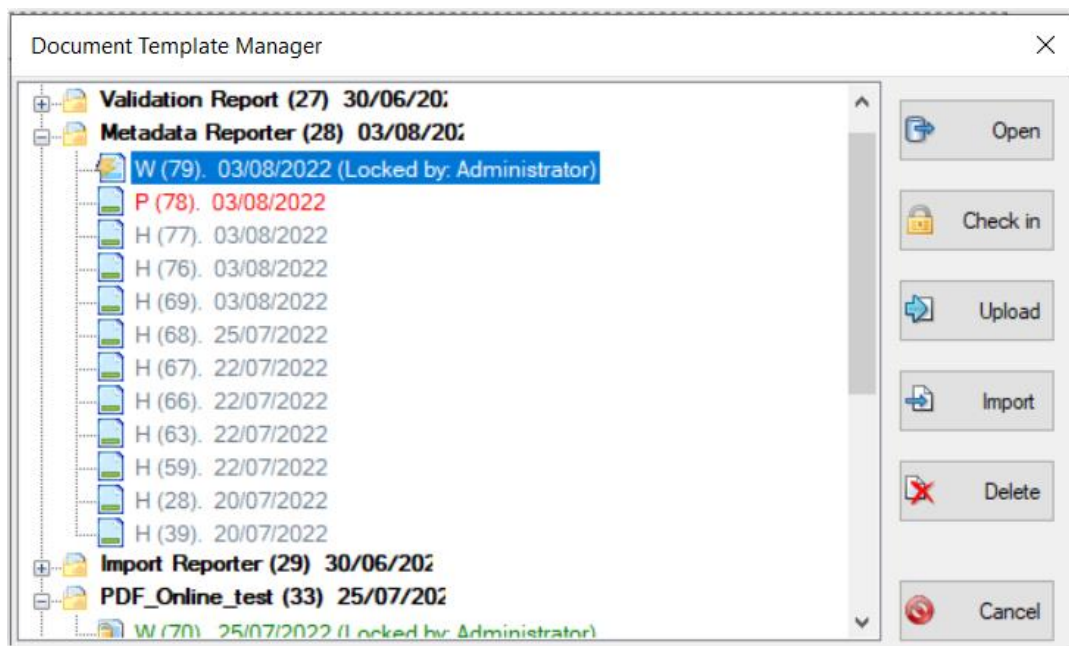


Рисунок 25 – Открытие шаблона документа

## 13 Работа с документами Word/PDF

13.1 В шаблон можно вставлять документ Word и PDF. Документ сохраняется в репозитории и может быть изменено из Composer редактором Word. Вы можете использовать всю страницу или часть страницы. Вы можете использовать несколько вставок Word из того же документа Word.

13.2 Для добавления документа Word или PDF необходимо добавить в контейнер поле «Insert» (см. рисунок 26)

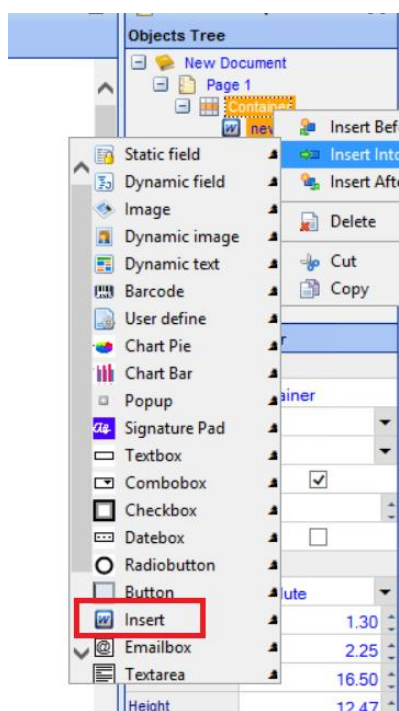


Рисунок 26 – Добавление элемента документа в контейнер шаблона

13.3 После добавления поля «Insert» в меню появляются кнопки добавления документов Word и PDF, см. рисунок 27.

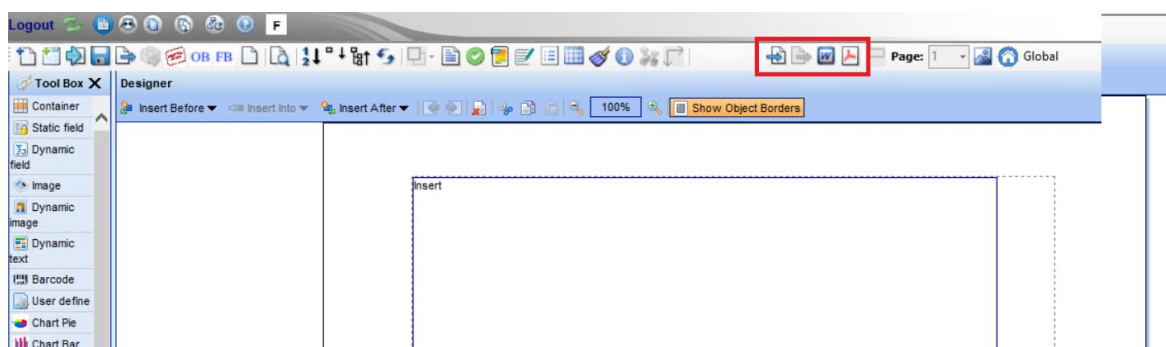


Рисунок 27 – Меню добавления добавления текста из документов Word и PDF

13.4 Нажав кнопку Word или PDF откроется документ Word или PDF в котором можно добавить текст и форматировать текст средствами редактора. По окончании редактирования документа необходимо

сохранить документ Word и в редакторе Word или PDF в меню выбрать раздел FormIT и сохранить информацию в систему, см. рисунок 28.

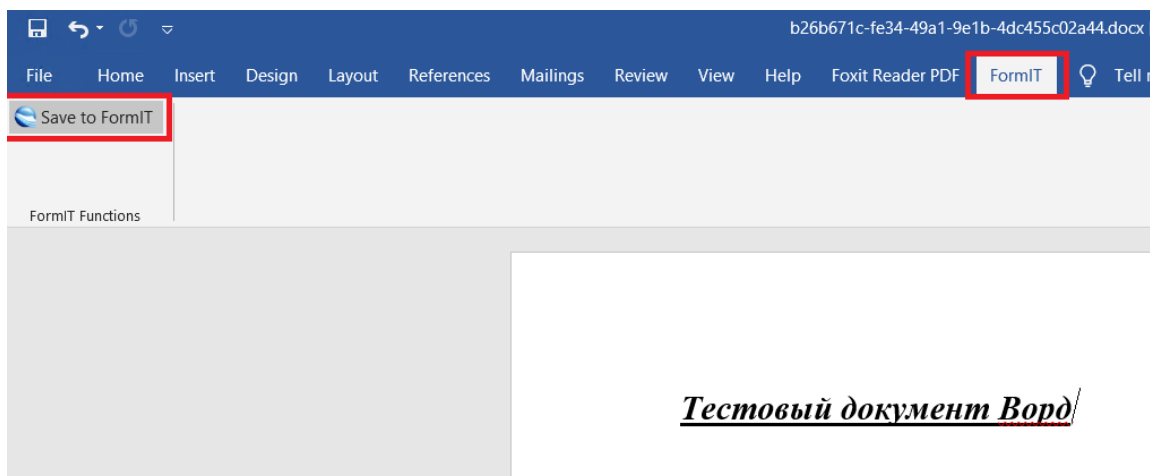


Рисунок 28 – Сохранение документа Word и PDF в Plus7 FormIT

13.5 Текст из сохраненного документа добавляется в созданный документ, см. рисунок 29.

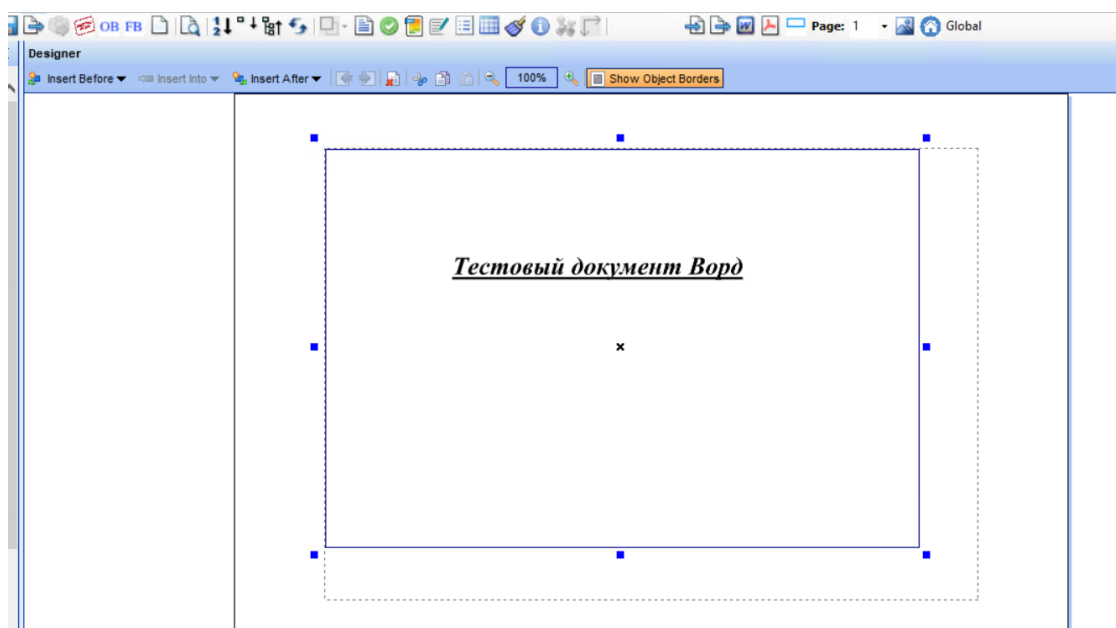


Рисунок 29 – Отображение текста документа Word и PDF в шаблоне документа

## 14 Работа с абзацами

14.1 Для того, чтобы добавить в контейнер абзац необходимо при выборе типа добавляемого поля выбрать «Paragraph», см. рисунок 30.

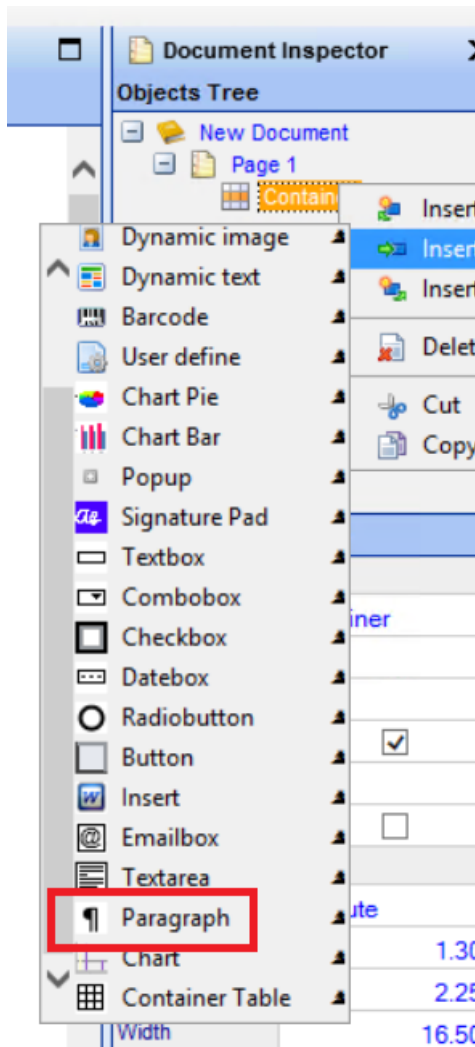


Рисунок 30 – Пример создания абзаца, шаг 1

14.2 После добавления поля «Paragraph» в контейнер в верхнем меню появится кнопка «Paragraph», см. рисунок 31.



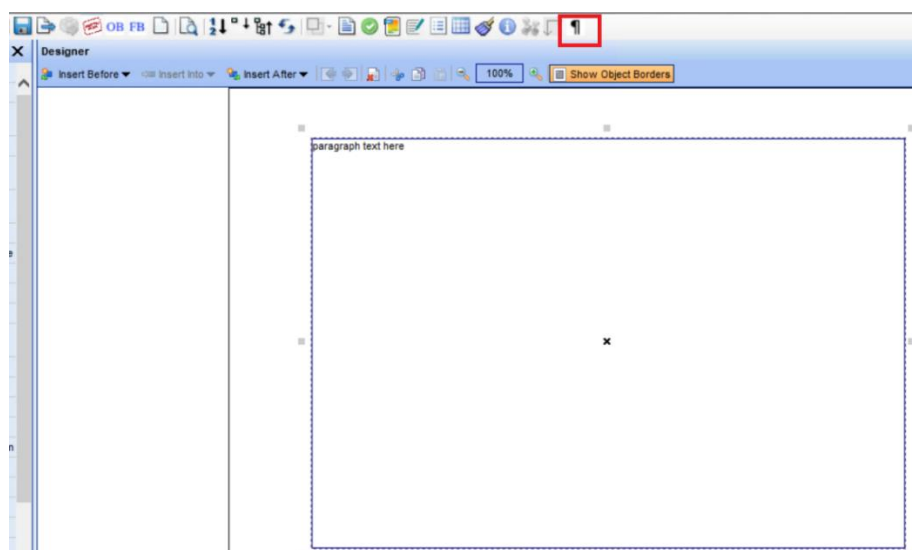


Рисунок 31 – Пример создания абзаца, шаг 2

14.3 При нажатии кнопки «Paragraph» открывается редактор текстовых форм, см. рисунок 32.

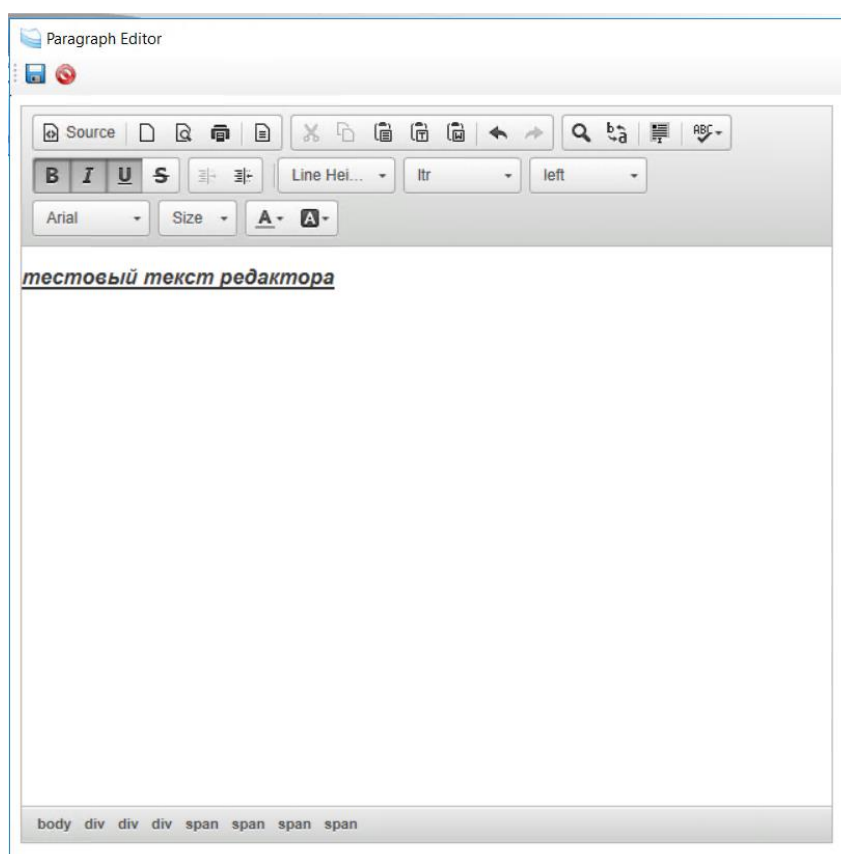


Рисунок 32 – Редактор текстовых форм

14.4 После сохранения отредактированного текста, он добавляется в шаблон документа.

## 15 Предварительный просмотр

15.1 Предварительный просмотр (смотри рисунок 33) шаблона документа с образцами данных (например, предварительный просмотр печати). Моделирование данных можно использовать для отладки и проверки на наличие ошибок. Для этого можно нажать кнопку быстрого просмотра или кнопку просмотра документа онлайн. При просмотре документа онлайн есть два вариант отображения документа PDF или HTML.

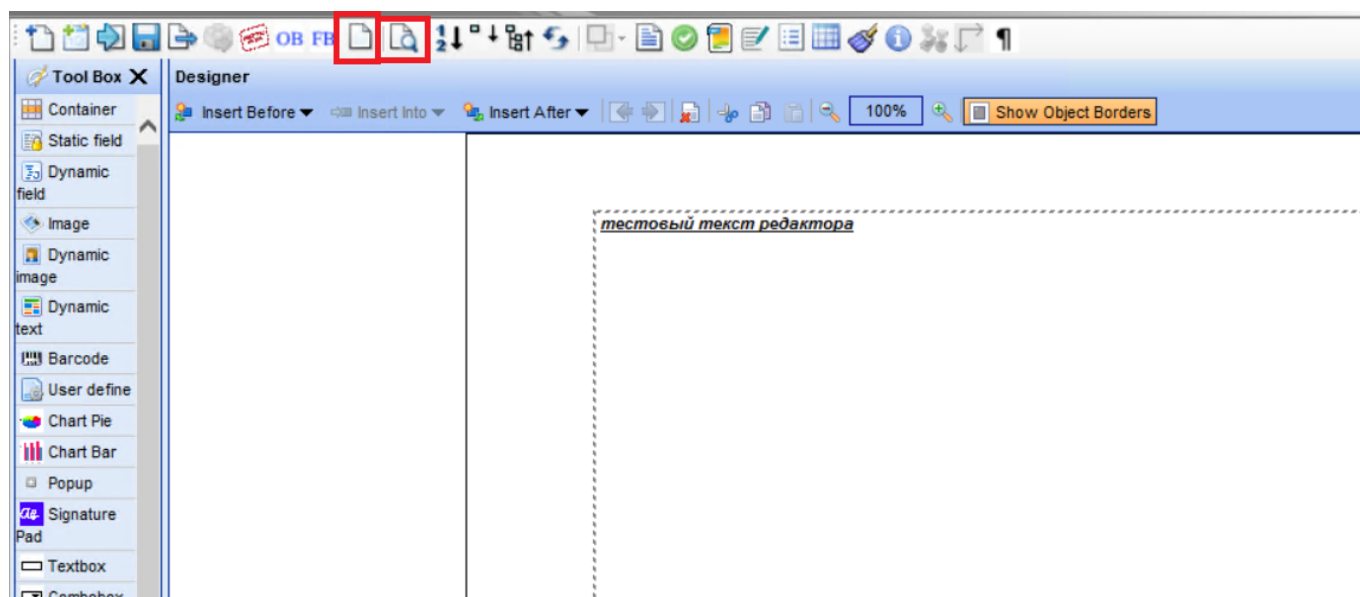


Рисунок 33 – Предварительный просмотр документа

## 16 Сборка

16.1 Сборка - это процесс идентификации и сопоставления всех статических и динамических элементов, которые должны отображаться в документе (данные и объекты).

16.2 В процессе сбора данные извлекаются из всех соответствующих систем источников, интегрируются, форматируются и загружаются в хранилище FormIt в формате XML-файла

16.3 Сборка состоит из двух частей:

- дерево сборки;
- карты сборки.

## 17 Создание дерева сборки

17.1 Деревья сборки создаются в рабочей области коллекции.

17.2 Сохраняется в формате XML-файла.

17.3 Существует одно дерево сборки для одного проекта (смотри рисунок 34).

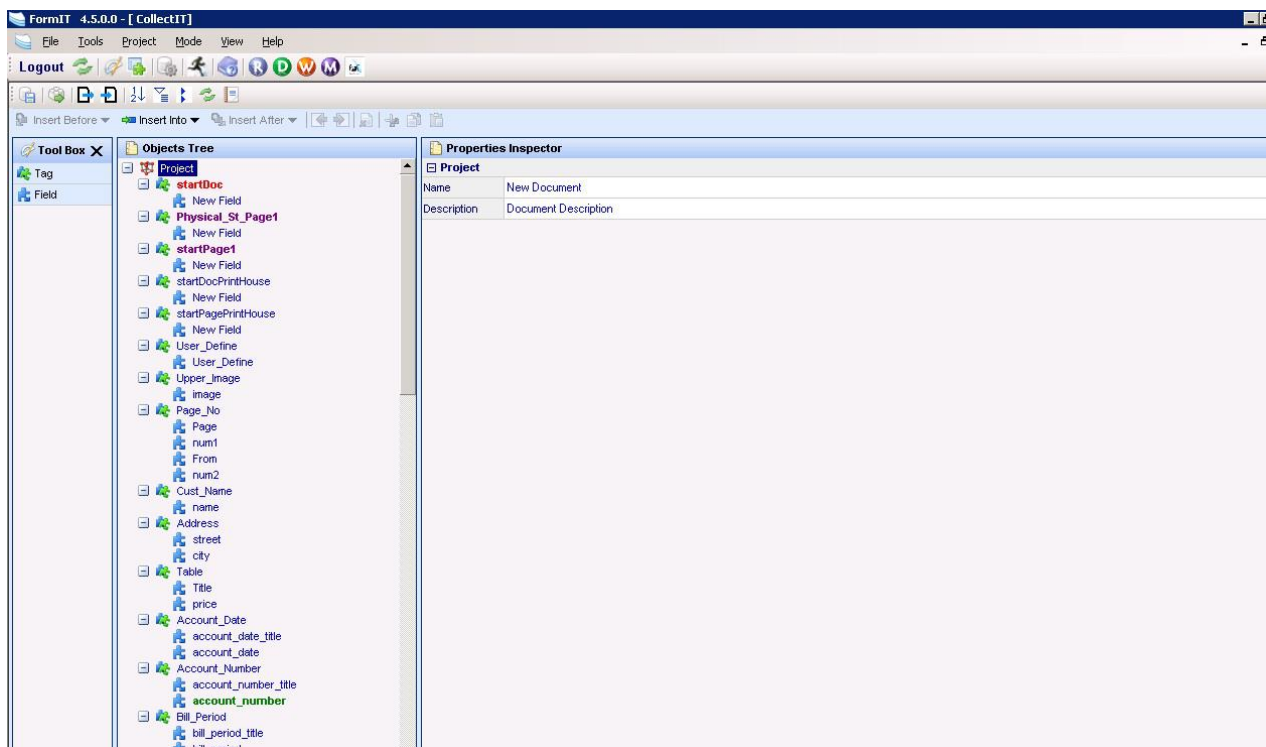


Рисунок 34 – Пример дерева сборки проекта

## 18 Сохранение

18.1 Деревья сборки могут быть сохранены в XML-файле или в формате репозитория DB.

18.2 Как только дерево сборки будет завершено и готово к развертыванию, сохраните данные, а затем выполните загрузку (см. рисунки 35 и 36):

18.2.1 Разбивает XML на несколько реляционных таблиц БД, используемых в процессах сбора данных

18.2.2 Продвигает записи коллекции в базе данных репозитория на основе модели состояния:

- W (работа);
- P (производство);
- H (история).

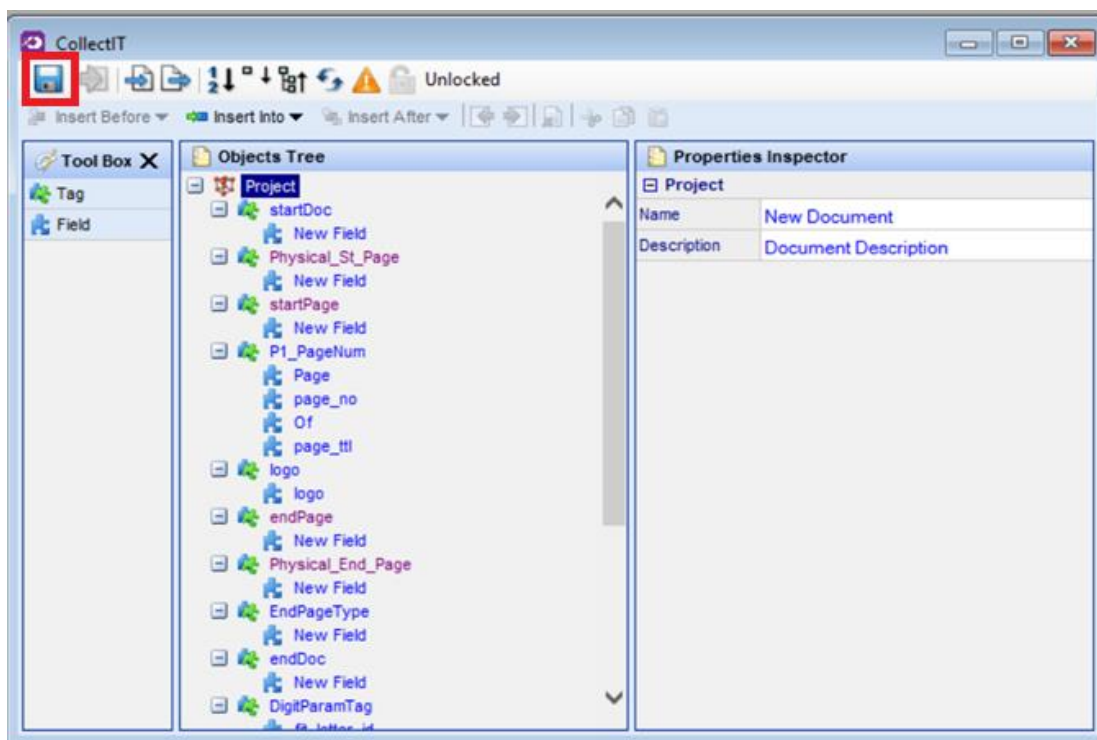


Рисунок 35 – Сохранение дерева сборки объектов проекта

18.3 После сохранения дерева объектов кнопка загрузки дерева в систему станет активной, см. рисунок 36. Необходимо произвести загрузку дерева объектов проекта в систему нажав на данную кнопку.

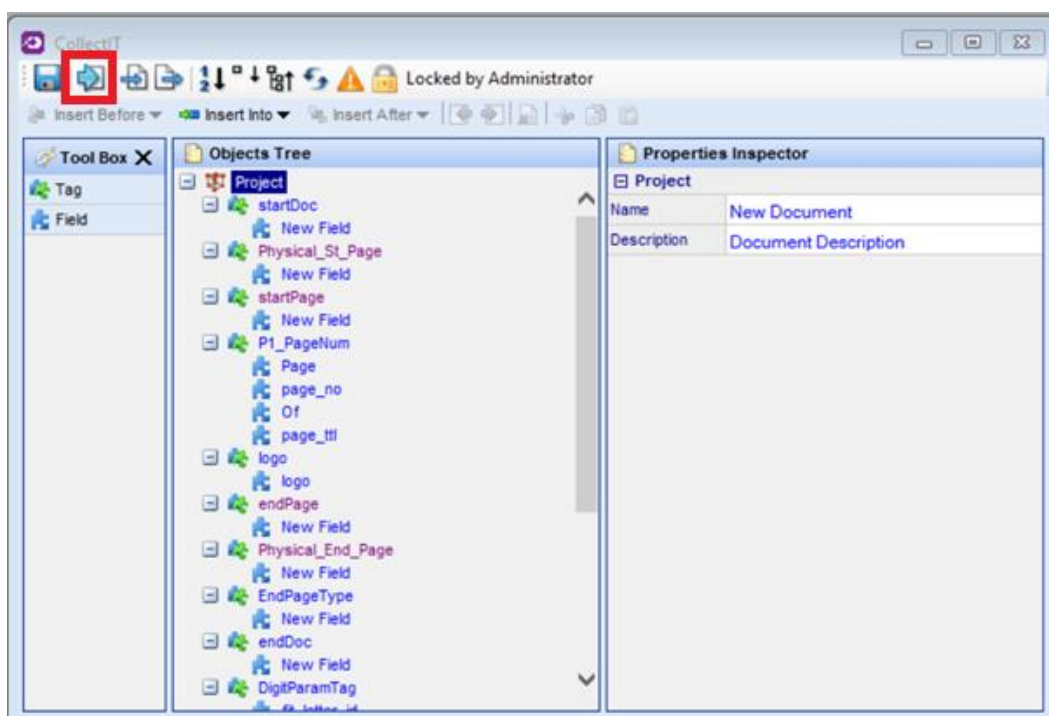


Рисунок 36 – Отображение дерева сборки объектов проекта с активной кнопкой загрузки в систему

## 19 Использование объектов

19.1 Объект — это любое статическое или динамическое изображение или сообщение, которое должно быть представлено в документе.

19.2 Доступ к рабочей области ФОРМЫ осуществляется с помощью меню Режим => ORM.

19.3 Все загруженные объекты регистрируются в базе данных репозитория, а физические объекты загружаются в IMG\_DIR в соответствии с системным параметром.

## 20 Модуль управления правилами

20.1 Данный модуль управляет всеми объектами (ресурсами) и всеми правилами, которые используются в проекте (смотри рисунок 37).

20.2 Модуль представляет собой инструмент маркетинга для персонализации динамического содержимого документов для конечных пользователей.

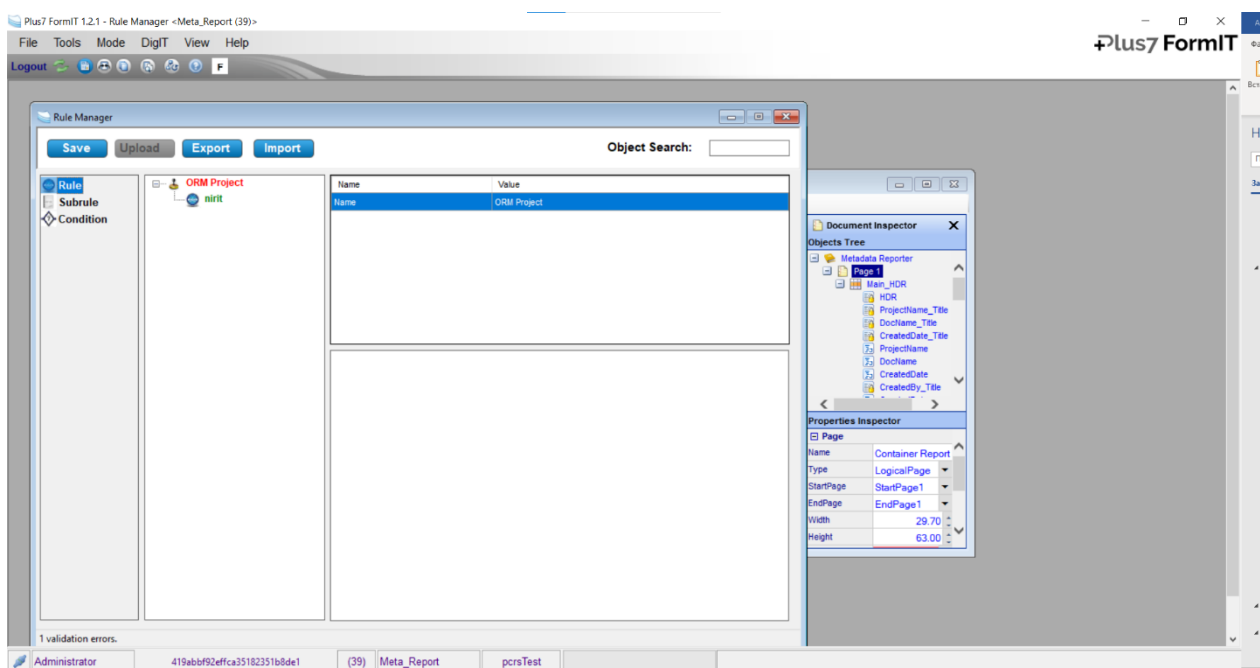


Рисунок 37 - Окно правил использования объектов



## 21 Создание правил

21.1 Правило - это бизнес-логика, связанная с данными, которая запускает размещение определенного динамического объекта в документе при его выполнении.

21.2 Определите правило на вкладке (смотри рисунок 38):

- правила в ORM;
- щёлкните правой кнопкой мыши, чтобы добавить новое правило;
- дайте значимые имена;
- щёлкните правой кнопкой мыши, чтобы добавить подправило.

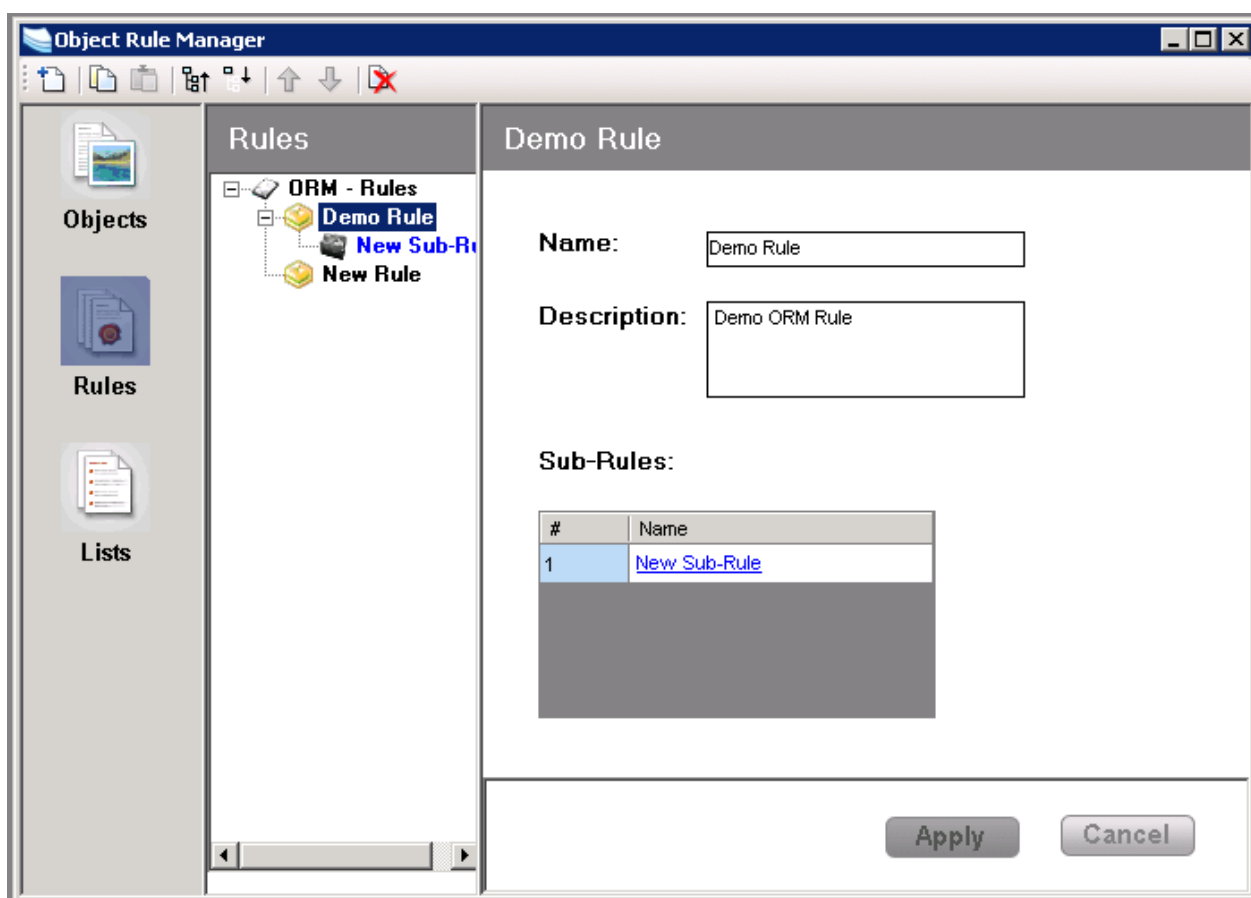


Рисунок 38 - Пример заполнения правил

## 22 Модуль архивации

Модуль сохраняет документы в макете шаблона, который был действительным/активным при их создании.

### 22.1 Элементы документа

#### 22.1.1 Документы состоят из:

- динамических данных, которые изменяются в документе;
- метаданных шаблона документа, определенных в ComposeIT;
- стиля (набора) преобразований XSL, которые определяют формат вывода документа (например, HTML, PDF, CSV, PopUp HTML).

22.1.2 Все элементы хранятся в виде XML-записей в репозитории (в отличие от хранения физических файлов).

### 22.2 Извлечение документов из архива

#### 22.2.1 Вы можете извлекать документы из архива следующими способами:

- используя приложение Format, через модуль архивирования (см. рис. 39);
- путём интеграции формата с внешними приложениями через поставляемый веб-сервис.

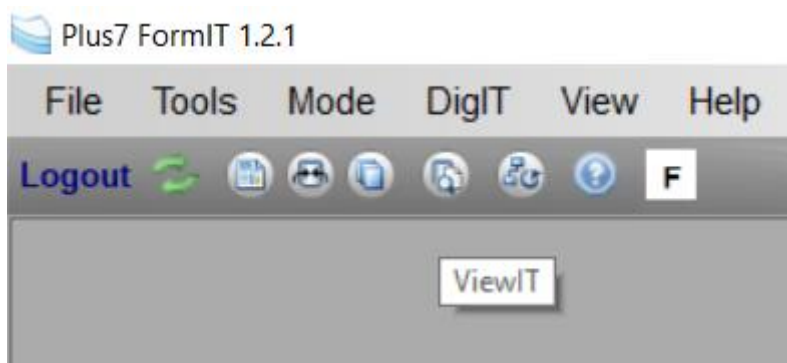


Рисунок 39 – Кнопка вызова модуля ViewIT

### 22.3 Таблицы архивов

22.3.1 Обычно для создаваемого архива создается отдельная схема – FORMIT\_ARCHIVE или FIT\_ARC

22.3.2 Схема состоит из 2 основных таблиц:

- PF\_ARCHIVE\_INDEX
- PF\_ARCHIVE\_DATA

22.3.3 Для каждой строки в PF\_ARCHIVE\_INDEX может быть много строк в PF\_ARCHIVE\_DATA

22.3.4 Связь между таблицами - это ключ PF\_ARC\_INDEX\_ID

### 22.4 Добавление запроса

22.4.1 Для получения документа можно использовать запросы (смотри рисунок 40).

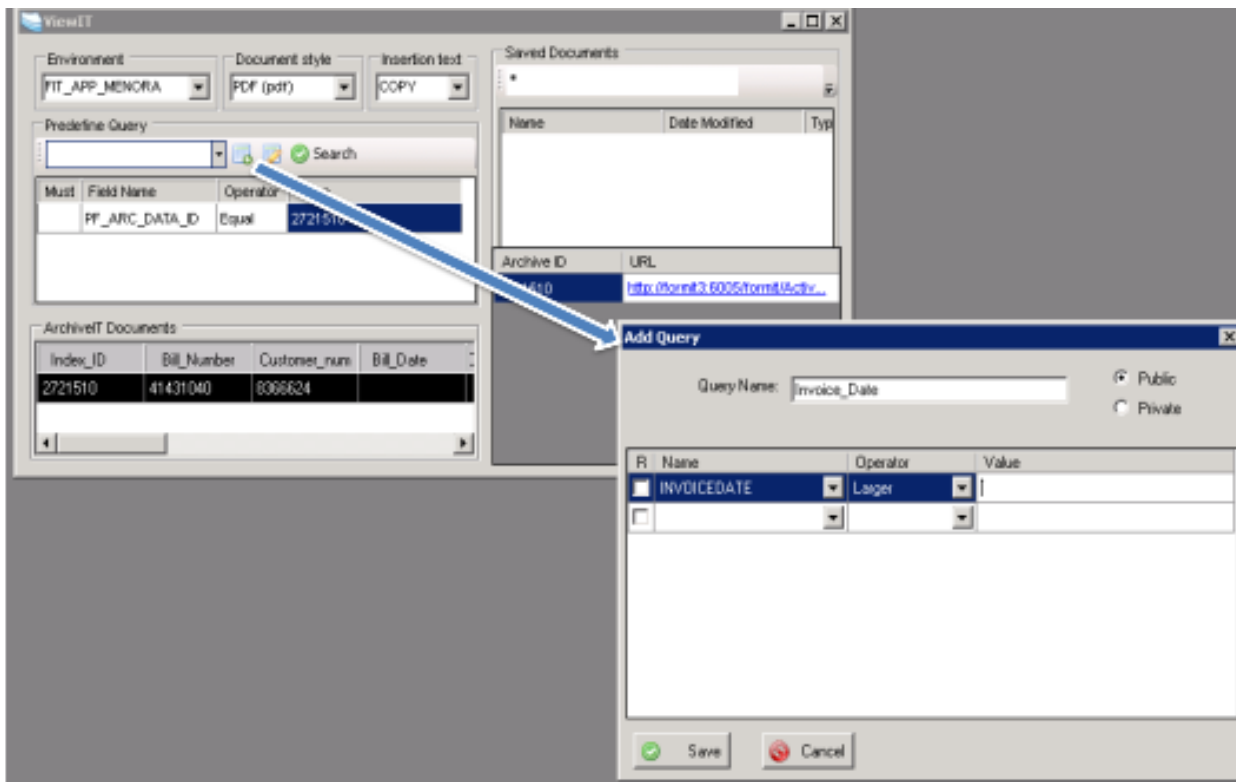


Рисунок 40 – Пример использования запросов