

**ОБЩЕСТВО С ОГРАНИЧЕННОЙ ОТВЕТСТВЕННОСТЬЮ  
«ПРОГРЭДИС»  
(ООО «ПРОГРЭДИС»)**

УТВЕРЖДАЮ

Генеральный директор

ООО «Прогрэдис»

\_\_\_\_\_ В.В. Субботин

«\_\_» \_\_\_\_\_ .2016

**СИСТЕМА УПРАВЛЕНИЯ РАЗРАБОТКОЙ  
НОРМАТИВНЫХ ДОКУМЕНТОВ  
(РАНДОК.НОРМАТИВНЫЕ ДОКУМЕНТЫ)  
ЭКСПЛУАТАЦИОННАЯ ДОКУМЕНТАЦИЯ**

**Руководство по инсталляции**

Листов 23

2016

## ОГЛАВЛЕНИЕ

1. ВВЕДЕНИЕ .....	3
1.1. Идентификация системы.....	3
1.2. Наименования организации Разработчика.....	3
1.3. Цели и назначение использования системы .....	3
1.4. Краткое описание возможностей .....	4
1.5. Наименование функций системы, реализуемых на данном оборудовании, количественные характеристики работы системы .....	4
2. ОБЛАСТЬ ПРИМЕНЕНИЯ .....	6
3. НОРМАТИВНЫЕ ССЫЛКИ .....	7
3.1. Основание для проведения работ.....	7
3.2. Перечень нормативно-технических документов, методических материалов, использованных при разработке системы .....	7
4. ПОРЯДОК УСТАНОВКИ.....	8
4.1. Требования к установке системы .....	8
4.2. Установка системы .....	8
4.3. Настройка системы.....	11
ПРИЛОЖЕНИЕ 1. СОДЕРЖАНИЕ КОНФИГУРАЦИОННОГО ФАЙЛА APPLICATION.CONF .....	13

## **1. ВВЕДЕНИЕ**

### **1.1. Идентификация системы**

Полное наименование системы: Система управления разработкой нормативных документов (далее РАНДОК или Система).

Сокращенное наименование системы: РАНДОК.

### **1.2. Наименования организации Разработчика**

Общество с ограниченной ответственностью «Прогрэдис» (ООО «Прогрэдис»); адрес: 123557, г. Москва, Большой Тишинский переулок, дом 26, корпус 13/14.

### **1.3. Цели и назначение использования системы**

Цель создания системы - обеспечение участников разработки нормативных документов специализированным инструментом для управления разработкой сложных нормативных документов различного вида, обеспечения удобных и быстрых коммуникаций между участниками разработки, мониторинга хода разработки нормативных документов и автоматического формирования оперативной и сводной отчетности.

Система предназначена для:

- управления процессами планирования, разработки, согласования, утверждения нормативных документов, а также мониторинга и контроля;
- предоставления единого интерфейса (рабочего места), с помощью которого выполняются все работы по управлению разработкой нормативных документов;
- единого хранилища данных и вспомогательных документов, формируемых в ходе работ по разработке, согласованию, утверждению нормативных документов;
- формирования системы отчетности по ходу разработки нормативных документов, позволяющей формировать визуальные и табличные отчеты в различных разрезах и степени детализации.

#### **1.4. Краткое описание возможностей**

Использование системы РАНДОК позволяет:

- 1) Управлять разработкой нормативных документов:
  - Автоматическое формирование рабочего плана разработки;
  - Настройка участников разработки нормативного документа;
  - Автоматическая постановка задач исполнителям в соответствии с ролью участника разработки;
  - Автоматический контроль сроков исполнения;
  - Список задач и нормативных документов в разработке которых участвует пользователь.
- 2) Выполнять мониторинг разработки нормативных документов, включая:
  - просмотр сводной информации по разработке нормативных документов;
  - просмотр сводной информации по задачам на исполнение;
  - просмотр перечня разрабатываемых нормативных документов,
  - просмотр деталей по нормативному документу;
- 3) Вести список организаций, сотрудников организаций и групп пользователей;
- 4) Просматривать свойства организаций и сотрудников организаций, вносить изменения в значения свойств организаций и сотрудников организаций при наличии достаточных прав доступа;

#### **1.5. Наименование функций системы, реализуемых на данном оборудовании, количественные характеристики работы системы**

Система РАНДОК обеспечивает реализацию следующих основных функций/задач:

- Управление разработкой нормативных документов;
- Постановка и исполнение задач пользователями;
- Управление организациями и группами пользователей;
- Управление пользователями и их ролями;
- Мониторинг процессов разработки нормативных документов;

- Формирование отчетности по разработке нормативных документов;

Система РАНДОК поддерживает многопользовательский режим, который обеспечивает одновременную работу нескольких сотен пользователей. Поддерживаемое количество одновременных пользователей – не менее 200 человек.

Система РАНДОК обеспечивает функционирование в круглосуточном режиме с допустимыми перерывами на профилактику, и перенастройку, и простоями, связанными с неисправностью оборудования, не более 48 часов в год при среднем времени устранения вызвавшей простой неисправности, не более 3 часов. Коэффициент готовности Системы РАНДОК равен 0,995.

Поддерживаемое количество одновременно разрабатываемых нормативных документов – не менее 1 000 единиц.

## **2. ОБЛАСТЬ ПРИМЕНЕНИЯ**

Система РАНДОК обеспечивает управление разработкой, согласованием, утверждением сложных нормативных документов на основе процессного подхода и безбумажных технологий.

### **3. НОРМАТИВНЫЕ ССЫЛКИ**

#### **3.1. Основание для проведения работ**

Разработка системы РАНДОК ведется на основании следующих документов:

- **Приказ №01-ИС от 27.12.2012 г.** «О начале работ по проекту «Система разработки нормативных документов (РАНДОК)»

#### **3.2. Перечень нормативно-технических документов, методических материалов, использованных при разработке системы**

При разработке системы были использованы следующие нормативные документы:

- **ГОСТ 34.602-89.** Информационная технология. Комплекс стандартов на автоматизированные системы. Техническое задание на создание автоматизированной системы.

## 4. ПОРЯДОК УСТАНОВКИ

### 4.1. Требования к установке системы

Комплекс технических средств должен удовлетворять минимальным требованиям, приведенным в таблице 1:

Таблица 1. Требования к комплексу технических средств

Процессор	Intel Xeon 2 ядра, 2.3ГГц
Оперативная память	2 Гб
Жесткий диск	300 Гб
Операционная система	Ubuntu 14.04.4 LTS

### 4.2. Установка системы

#### 4.2.1. Состав и структура дистрибутива dcs.zip:

- Директория backup предназначена для хранения резервных копий по умолчанию (при необходимости с помощью файла настройки может быть изменено);
- Директория www содержит файлы приложения:
  - директория bin содержит скрипты запуска приложения;
  - директория conf содержит файлы конфигурации application.conf (см. Приложение 1), а также три директории:
    1. evolutions с миграциями базы данных;
    2. META-INF с настройками ORM;
    3. update с загружаемыми данными;
  - директория lib включает библиотеки java необходимые для работы приложения;
  - директория processes содержит модели процессов разработки нормативных документов;
  - директория public содержит данные для отображения пользовательского интерфейса приложения;

- директория reports содержит шаблоны отчетов.
- Скрипты для формирования структуры СУБД и заполнения справочников:
  - dcsdbstructure.sql – структура таблиц системы;
  - dcsdbdata.sql – справочники системы;
  - dcsdbdata2.sql – справочники системы (продолжение).

4.2.2. Установка приложения выполняется в следующей последовательности:

- Установка ОС выполняется согласно документации на ОС Ubuntu 14.04.4 LTS;
- Установка БД выполняется согласно документации на MySQL Community Server 5.6;
- Установка Oracle Java 8 выполняется согласно документации по установке Oracle Java 8 на Ubuntu;
- Установка шрифтов, используемых в шаблонах отчетов, выполняются следующие команды через терминал:
  - «sudo apt-get install fontconfig»
  - «apt-get install ttf-mscorefonts-installer»
  - «fc-cache -v»
- Установка исполняемого приложения выполняется в следующей последовательности:
  - перенос дистрибутива dcs.zip на сервер, на котором будет установлена Система в папку /srv;
  - распаковка дистрибутива путем выполнения команды: «unzip dcs.zip».
- Настройка БД выполняется в следующей последовательности:
  - Запуск консоли mysql: «mysql -u root -p»;
  - Создание базы данных: «CREATE DATABASE dcs CHARACTER SET utf8 COLLATE utf8\_general\_ci;»

- Создание пользователя: «GRANT ALL ON dcs.\* TO dcsuser@localhost IDENTIFIED BY '12345678';» и «flush privileges;».

где '12345678' пароль, используемый по умолчанию, следует задать более сложный пароль и изменить его в файле настройки приложения – application.conf.

- Выполнение скриптов БД
  - Запуск консоли mysql: «mysql -u root -p»;
  - Выполнение команд:
    1. «source /srv/dcs/dcsdbstructure.sql»;
    2. «source /srv/dcs/dcsdbdata.sql»;
    3. «source /srv/dcs/dcsdbdata2.sql».
- Настройка запуска Системы, как сервиса ОС:
  - Выполнение команд:
    1. «sudo ln -s /srv/dcs/www/bin/init.d /etc/init.d/dcs»;
    2. «sudo update-rc.d dcs defaults»;
    3. «sudo chmod -R 755 /srv/dcs/www/bin».
  - Проверка работоспособности сервиса:
    1. «sudo service dcs start»;
    2. «sudo service dcs stop».

4.2.3. Установка клиентской части приложения не требуется.

4.2.4. Определение пути доступа по сети к серверу.

4.2.4.1. Подключение к Системе по умолчанию осуществляется через http-порт 9000. Если требуется использовать другой порт, то необходимо внести изменения в application.conf.

4.2.5. Загрузка необходимых компонентов системы

4.2.5.1. Загрузка дополнительных компонентов для работы РАНДОК не требуется.

4.2.6. Проверка функционирования программно-технического комплекса

4.2.6.1. Для проверки функционирования программно-технического комплекса необходимо запустить WEB-браузер на ПК пользователя (при помощи штатных средств ОС) и в строке ввода адреса указать адрес входа в

РАНДОК вида «server.url:9000», где server.url параметр, задаваемый в конфигурационном файле application.conf, если не задан, то http://ip сервера:9000. После чего в главном окне WEB-браузера должно появиться приглашение на авторизацию в Систему (см. Рисунок 1).

Рисунок 1. Приглашение на авторизацию в Систему

По умолчанию в системе существует пользователь с правами администратора системы:

Логин – ndadmin@ndadmin.ru

Пароль - ndadmin@ndadmin.ru

### 4.3. Настройка системы

#### 4.3.1. Настройка серверной части

4.3.1.1. Настройка приложения осуществляется через конфигурационный файл application.conf:

- Прописать URL приложения в параметре «server.url = ""» раздела # Application url configuration;
- Проверить настройки раздела # Application storage folder configuration, в котором указана папка, в которой будут храниться файлы, загружаемые в систему;
- Проверить настройки раздела # Database configuration;
- Для рассылки уведомлений по почте настроить раздел # Email configuration.

#### 4.3.2. Настройка клиентской части

4.3.2.1. Настройка клиентской части для Системы не требуется.

4.3.2.2. Для работы клиентского компонента РАНДОК необходимо наличие на ПК пользователя одного из WEB-браузеров: Microsoft Internet Explorer 9 и выше, Mozilla Firefox 17 и выше, Google Chrome 20 и выше. В настройках WEB-браузера должно быть обязательно разрешено использование JavaScript и Cookies.

4.3.2.3. Для первоначальной установки и настройки браузеров необходимо обратиться к руководству пользователя и/или инструкции по установке соответствующего WEB-браузера.

4.3.2.4. WEB-браузер Microsoft Internet Explorer можно скачать по ссылке: <http://windows.microsoft.com/ru-ru/internet-explorer/ie-11-worldwide-languages>.

4.3.2.5. WEB-браузер Mozilla Firefox можно скачать по ссылке: <https://www.mozilla.org/ru/firefox/new/>.

4.3.2.6. WEB-браузер Google Chrome можно скачать по ссылке: <https://www.google.ru/chrome/browser/desktop/index.html>.

4.3.2.7. Для запуска РАНДОК необходимо наличие адреса системы.

## ПРИЛОЖЕНИЕ 1. СОДЕРЖАНИЕ КОНФИГУРАЦИОННОГО ФАЙЛА APPLICATION.CONF

```
# Application info
application {
  name = "dcs"
  version = "1.14.5_b5"
  global = "play.PlayGlobalSettings"
}

# Application url configuration

server.url = ""

# Application storage folder configuration

storage.folder = ${?user.dir}/storage

# Play framework configuration

# Database configuration
db {
  default {
    driver = "com.mysql.jdbc.Driver"
    host = "localhost"
    db = "dcs"
    url = "jdbc:mysql://"${db.default.host}":3306/"${db.default.db}"?characterEncoding=UTF-8"
    username = "dcsuser"
    password = "12345678"
    partitionCount = 3
    maxConnectionsPerPartition = 5
    jndiName = "DefaultDS"
    #logStatements = true
  }
}

play {
  # Crypto configuration
  crypto {
    secret = "<=8[he_qWNuoK:Ym5x?b31Wt5dG4:u3/eoYWI0:NwDKkv_mjoay[=f>rwxFg>yrF"
  }
  # Internationalisation configuration
  i18n {
```

```

# The languages supported by this application
langs = ["ru"]
# A path to prefix message file loading with. Use this if you want to place your messages resources at
some path
# other than the root application path.
path = null
# The name of the cookie to store the Play language in. This cookie is set when Langs.setLang is invoked,
and
# read when the preferred lang is loaded.
langCookieName = "PLAY_LANG"
# Whether the language cookie should be secure or not
langCookieSecure = false
# Whether the HTTP only attribute of the cookie should be set to true
langCookieHttpOnly = false
}
http {
# The application context.
# Must start with /.
context = "/"
# The error handler.
# Used by Play's built in DI support to locate and bind a request handler. Must be the FQCN of a Play
router.
# If null, will attempt to load a class called Routes in the root package, otherwise if that's not found, an
empty
# router will be bound.
router = null
# The request handler.
# Used by Play's built in DI support to locate and bind a request handler. Must be one of the following:
# - A FQCN that implements play.api.http.HttpRequestHandler (Scala).
# - A FQCN that implements play.http.HttpRequestHandler (Java).
# - provided, indicates that the application has bound an instance of play.api.http.HttpRequestHandler
through some
# other mechanism.
# If null, will attempt to load a class called RequestHandler in the root package, otherwise if that's
# not found, will default to play.api.http.GlobalSettingsHttpRequestHandler, which delegates to legacy
request
# handling methods on Global.
requestHandler = null
# The error handler.
# Used by Play's built in DI support to locate and bind an error handler. Must be one of the following:
# - A FQCN that implements play.api.http.HttpErrorHandler (Scala).
# - A FQCN that implements play.http.HttpErrorHandler (Java).
# - provided, indicates that the application has bound an instance of play.api.http.HttpErrorHandler
through some
# other mechanism.

```

```

# If null, will attempt to load a class called ErrorHandler in the root package, otherwise if that's
# not found, will default to play.api.http.GlobalSettingsHttpErrorHandler, which delegates to legacy
error handling
# methods on Global.
errorHandler = null
# The filters.
# Used by Play's built in DI support to locate and bind a class to provide filters. Must be one of the
following:
# - A FQCN that implements play.api.http.HttpFilters (Scala).
# - A FQCN that implements play.http.HttpFilters (Java).
# - provided, indicates that the application has bound an instance of play.api.http.HttpFilters through
some
# other mechanism.
# If null, will attempt to load a class called Filters in the root package, otherwise if that's not found, will
# default to play.api.http.NoHttpFilters, which provides no filters.
filters = null
# Parsing configuration
parser = {
  # The maximum amount of a request body that should be buffered into memory
  maxMemoryBuffer = 100k
  # The maximum amount of a request body that should be buffered into disk
  maxDiskBuffer = 10m
}
# Action composition configuration
actionComposition = {
  # If annotations put directly on Controller classes should be executed before the ones put on action
methods
  controllerAnnotationsFirst = false
}
# Cookies configuration
cookies = {
  # Whether strict cookie parsing should be used. If true, will ignore the entire cookie header if a single
invalid
  # cookie is found, otherwise, will just ignore the invalid cookie if an invalid cookie is found. The reason
  # dropping the entire header may be useful is that browsers don't make any attempt to validate cookie
values,
  # which may open opportunities for an attacker to trigger some edge case in the parser to steal cookie
  # information. By dropping the entire header, this makes it harder to exploit edge cases.
  strict = true
}
# Session configuration
session = {
  # The cookie name
  cookieName = "PLAY_SESSION"
  # Whether the secure attribute of the cookie should be set to true

```

```

secure = false
# The max age to set on the cookie.
# If null, session cookies are used (that is, the cookie expires when the user closes their browser).
# An important thing to note, this only sets when the browser will discard the cookie. Play will consider
any
# cookie value with a valid signature to be a valid session forever. To implement a server side session
timeout,
# you need to put a timestamp in the session and check it at regular intervals to possibly expire it.
maxAge = null
# Whether the HTTP only attribute of the cookie should be set to true
httpOnly = true
# The domain to set on the session cookie
# If null, does not set a domain on the session cookie.
domain = null
}
# Flash configuration
flash = {
# The cookie name
cookieName = "PLAY_FLASH"
# Whether the flash cookie should be secure or not
secure = false
# Whether the HTTP only attribute of the cookie should be set to true
httpOnly = true
}
}
# Evolutions configuration
evolutions {
# Whether evolutions are enabled
enabled = true
# Whether evolution updates should be performed with autocommit or in a manually managed transaction
autocommit = false
# Whether locks should be used when apply evolutions. If this is true, a locks table will be created, and
will
# be used to synchronise between multiple Play instances trying to apply evolutions. Set this to true in
a multi
# node environment.
useLocks = false
# Whether evolutions should be automatically applied. In prod mode, this will only apply ups, in dev
mode, it will
# cause both ups and downs to be automatically applied.
autoApply = false
# Whether downs should be automatically applied. This must be used in combination with autoApply,
and only applies
# to prod mode.
autoApplyDowns = false

```

```

# Db specific configuration. Should be a map of db names to configuration in the same format as this.
db {
  default {
    enabled = false
    autocommit = false
    useLocks = false
    autoApply = true
    autoApplyDowns = true
  }
}
}
server {
  # The root directory for the Play server instance. This value can
  # be set by providing a path as the first argument to the Play server
  # launcher script. See `ServerConfig.loadConfiguration`.
  dir = "."
  dir = ${?user.dir}

  # HTTP configuration
  http {
    # The HTTP port of the server. Use a value of "disabled" if the server
    # shouldn't bind an HTTP port.
    port = 9000
    port = ${?http.port}
    # The interface address to bind to.
    address = "0.0.0.0"
    address = ${?http.address}
  }

  # HTTPS configuration
  https {
    # The HTTPS port of the server.
    port = ${?https.port}
    # The interface address to bind to
    address = "0.0.0.0"
    address = ${?https.address}
    # The SSL engine provider
    engineProvider = "play.core.server.ssl.DefaultSSLEngineProvider"
    engineProvider = ${?play.http.sslengineprovider}
    # HTTPS keystore configuration, used by the default SSL engine provider
    keyStore {
      # The path to the keystore
      path = ${?https.keyStore}
      # The type of the keystore
      type = "JKS"
    }
  }
}

```

```

    type = ${?https.keyStoreType}
    # The password for the keystore
    password = ""
    password = ${?https.keyStorePassword}
    # The algorithm to use. If not set, uses the platform default algorithm.
    algorithm = ${?https.keyStoreAlgorithm}
  }
  # HTTPS truststore configuration
  trustStore {
    # If true, does not do CA verification on client side certificates
    noCaVerification = false
  }
}

# The path to the process id file created by the server when it runs.
# If set to "/dev/null" then no pid file will be created.
pidfile.path = ${play.server.dir}/RUNNING_PID
pidfile.path = ${?pidfile.path}
}
}

# Backup configuration

backup {
  cron = "0 0 1 * * ?"
  makeBackupOnServerShutdown = true
  local {
    enabled = true
    directory = "/srv/dcs/backup"
  }
  # sftp {
  #   enabled = false
  #   host =
  #   port =
  #   user =
  #   password =
  #   directory =
  #   config.StrictHostKeyChecking = "no"
  # }
  # scp {
  #   enabled = false
  #   host =
  #   port =
  #   user =
  #   password =

```

```
# directory =
# config.StrictHostKeyChecking = "no"
# }
}

# Logger configuration

akka {
  loggers = ["akka.event.slf4j.Slf4jLogger"]
  loglevel = "INFO"
}

# User session configuration
session {
  duration = 8
}

# JPA configuration
jpa.default = "defaultPersistenceUnit"

# Email configuration
#smtp {
# mock = true
# host = ""
# port =
# ssl = true
# tls = false
# user = ""
# password = ""
#}

# Error report recipient configuration
report {
  from = "Система управления разработкой нормативных документов" <${smtp.user}>
  recipient = ""
}

# Support email configuration
#support.recipient = ""

#Auto job configuration
jobs {
  cleaner {
    cron = "0 0 */6 * * ?"
    keep = 6hours
  }
}
```

```
    report = "enabled"
  }

# Frequency checking emails
mailer {
  cron = "0 */5 * * * ?"
}

# Frequency checking new tasks
taskloader {
  cron = "*/5 * * * * ?"
}

# Frequency checking licence notification
notification {
  licenceInfo {
    enabled = true
    cron = "0 */5 * * * ?"
    attention {
      daysBefore = 28
      frequency = 7
    }
    overdue {
      daysAfter = 28
      frequency = 7
    }
  }
}
}

gzipFilter = false

parsers.MultipartFormData.maxLength = 10240K

updateplugin = enabled

# String quotes configuration

convertQuotes = false

# Process configuration

businessProcessEngine {
  setMainCuratorWhenProcessVariableIsEmpty = true
}
```

## СПИСОК СОКРАЩЕНИЙ

<b>Сокращение</b>	<b>Обозначение</b>
БД	База данных
ГОСТ	Государственный стандарт
ООО	Общество с ограниченной ответственностью
ПК	Персональный компьютер
ПО	Программное обеспечение
РАНДОК	Автоматизированная система управления разработкой нормативных документов
РФ	Российская Федерация



