

# HP Service Manager

Для поддерживаемых операционных систем Windows® и UNIX®

Версия ПО: 9.30

---

## Руководство Document Engine

Дата выпуска документа: июль 2011 г.

Дата выпуска ПО: июль 2011 г.



## Правовые уведомления

### Гарантия

Гарантии на продукты и услуги HP формулируются только в заявлениях о прямой гарантии, сопровождающих эти продукты и услуги. Никакая часть настоящего документа не может быть истолкована как дополнительная гарантия. Компания HP не несет ответственности за содержащиеся здесь технические или редакционные ошибки.

Приводимые здесь сведения могут быть изменены без какого-либо уведомления.

### Пояснение об ограниченных правах

Конфиденциальное компьютерное программное обеспечение. Для обладания, использования или копирования необходима действующая лицензия от компании HP. В соответствии с нормами FAR 12.211 и 12.212, коммерческое компьютерное программное обеспечение, документация на компьютерное программное обеспечение и технические данные для коммерческих позиций лицензируются государственным организациям США на условиях стандартной коммерческой лицензии поставщика.

### Заявление об авторских правах

© Hewlett-Packard Development Company, L.P. 1994-2011

### Заявления о товарных знаках

Adobe™ является товарным знаком компании Adobe Systems Incorporated. Java™ и все товарные знаки и логотипы на основе Java являются товарными знаками или зарегистрированными товарными знаками компании Sun Microsystems, Inc. в США и других странах.

Java™ является товарным знаком компании Sun Microsystems, Inc. в США.

Microsoft® и Windows® являются зарегистрированными в США торговыми знаками корпорации Microsoft Corporation.

Oracle® является зарегистрированным в США торговым знаком корпорации Oracle Corporation, Редвуд, Калифорния.

UNIX® является зарегистрированным торговым знаком The Open Group.

## Обновления документации

На титульном листе настоящего документа приведены следующие идентификационные данные.

- Номер версии программного обеспечения для указания версии ПО.
- Дата выпуска документа, которая меняется при каждом обновлении документа.
- Дата выпуска ПО, которая указывает дату выпуска текущей версии программного обеспечения.

Чтобы проверить наличие обновлений или убедиться в том, что используется последняя редакция документа, откройте веб-сайт

**<http://h20230.www2.hp.com/selfsolve/manuals>**

Для данного сайта требуется регистрация HP Passport и вход в систему. Чтобы зарегистрироваться для получения идентификатора HP Passport ID, перейдите на веб-сайт

**<http://h20229.www2.hp.com/passport-registration.html>**

Можно также выбрать ссылку **New users - please register** на странице входа в HP Passport.

Оформление подписки в службе поддержки соответствующего продукта также позволит получать обновленные и новые редакции. Обратитесь в торговое представительство компании HP для получения подробной информации.

## Поддержка

Используйте веб-сайт технической поддержки программного обеспечения компании HP по адресу

**<http://www.hp.com/go/hpsoftwaresupport>**

На этом веб-сайте приведена контактная информация и сведения о продуктах, услугах и поддержке компании HP Software.

Веб-сайт технической поддержки программного обеспечения компании HP предоставляет возможности самостоятельного решения проблем. Это позволяет быстро и эффективно получить доступ к интерактивным средствам технической поддержки, необходимым для управления компанией. Каждый клиент службы поддержки может пользоваться следующими преимуществами, предоставляемыми веб-сайтом технической поддержки:

- поиск интересующих документов базы знаний;
- отправка и контроль описаний конкретных случаев и расширенных запросов для получения технической поддержки;
- загрузка исправлений ПО;
- управление договорами на техническую поддержку;
- поиск контактов в HP для технической поддержки;
- проверка сведений о доступных услугах;
- участие в обсуждениях различных вопросов с другими заказчиками ПО;
- исследование определенных проблем и регистрация для обучения программному обеспечению.

В большинстве случаев для получения поддержки требуется регистрация HP Passport, а также договор на услуги технической поддержки. Чтобы зарегистрироваться для получения идентификатора HP Passport ID, перейдите на веб-сайт

**<http://h20229.www2.hp.com/passport-registration.html>**

Для получения дополнительных сведений об уровнях доступа перейдите на веб-сайт:

**[http://h20230.www2.hp.com/new\\_access\\_levels.jsp](http://h20230.www2.hp.com/new_access_levels.jsp)**

---

## Содержание

Руководство Document Engine.....	1
Содержание.....	5
Обзор средства Document Engine.....	8
Что такое «Средство Document Engine»?.....	8
Преимущества модульного подхода.....	8
Единообразие.....	9
Сокращение времени разработки.....	9
Гибкость.....	9
Доступ к средству Document Engine.....	9
Объекты.....	10
Создание и обновление объектов.....	11
Форма и поля определения объекта.....	11
Описания полей вкладки «Информация объекта».....	12
Описания полей вкладки «Блокировка».....	14
Описания полей вкладки «Модификации».....	15
Описание полей вкладки «Переменные и глобальные списки».....	15
Описания полей вкладки «Операции».....	16
Описания полей вкладки «Предупреждения».....	17
Описания полей вкладки «Утверждения».....	18
Описания полей вкладки «Управление очередями».....	20
Описания полей вкладки «Представления и шаблоны».....	22
Описания полей вкладки «Уведомления».....	22
Описания полей вкладки «Конфигурация поиска».....	23
Вкладка «Определенные запросы».....	23
Вкладка «Диапазоны».....	24
Состояния.....	26
Поиск.....	26
Списки записей.....	26

Просмотр одной записи.....	26
Обзор записи.....	26
Советы по интеграции для приложения отображения.....	27
Создание и обновление состояний.....	27
Описания полей определения состояния.....	27
<b>Процессы.....</b>	<b>30</b>
Создание и изменение процесса.....	30
Форма «Определение процесса» и описание полей.....	30
Вкладка «Исходные выражения».....	31
Вкладка «Исходный Javascript».....	32
Вкладка RAD.....	32
Вкладка «Финальные выражения».....	32
Вкладка «Финальный Javascript».....	33
Вкладка «Следующий процесс».....	33
<b>Ресурсы средства Document Engine.....</b>	<b>34</b>
Объект DEFAULT.....	34
Приложения RAD.....	34
se.search.engine.....	35
se.list.engine.....	35
se.view.engine.....	35
Поток приложений RAD.....	35
Советы по интеграции для приложения отображения.....	36
Базовые функции в se.search.engine.....	37
Базовые функции в se.view.engine.....	38
Базовые функции в se.list.engine.....	41
Локальные переменные.....	41
<b>Обзор поиска и устранения неполадок.....</b>	<b>44</b>
Исследование пути приложения в Document Engine.....	44
Поиск используемого dbdict или объекта.....	44
Поиск состояния записей.....	44
Поиск имени процесса.....	45
Исследование ошибок приложений.....	45

Печать значений переменных или результатов выражений .....	46
<b>Обзор примера заказа на работу.....</b>	<b>48</b>
Создание таблицы.....	49
Добавление ключевых полей к таблице.....	50
Создание формы.....	51
Создание копии формы.....	52
Создание связи для формы заказа на работу.....	53
Создание файла последовательных номеров.....	54
Создание определения объекта.....	54
Создание определения процесса инициализации.....	56
Создание определения экрана приложения отображения.....	56
Создание определений параметров приложения отображения.....	58
Создание определения состояния.....	64
Добавление кнопки закрытия заказа на работу.....	65
Создание мастера для заказа на работу.....	67
Добавление записи определения процесса.....	70
Создание формы ввода для мастера.....	71
Изменение форм закрытия и обновления инцидента.....	73
Создание псевдонима в таблице probsummary для связи.....	75
Изменение определения процесса im.set.close.....	76
Тестирование примера заказа на работу.....	78

# Глава 1

---

## Обзор средства Document Engine

Средство Document Engine представляет собой средство расширения функциональных возможностей, которое включает возможность настройки системы без необходимости изменения RAD. Это средство предоставляет централизованный способ установки привилегий и поведения для стандартных действий, таких как представление в виде списка, просмотр и поиск. Это средство позволяет обеспечить единообразие работы различных модулей. Три основных компонента средства Document Engine — объекты, состояния и процессы. Процессы могут использоваться повторно для повышения степени модульности программирования, а интеграция не требует дополнительных операций, что позволяет сократить время разработки.

### Что такое «Средство Document Engine»?

Средство Document Engine представляет собой набор инструментов и методологий для разработки и изменения рабочих процессов Service Manager. Средство Document Engine было исходно предназначено для разработки основополагающего набора базовых функциональных возможностей с поддержкой различных модулей в составе Service Manager, повышения целостности пользовательского интерфейса различных модулей и сокращения объема кода, необходимого для создания нового модуля.

Средство Document Engine расширяет возможности приложений отображения за счет упрощенных и более мощных действий, в частности, включающих вызовы нескольких приложений. Кроме того средство Document Engine поддерживает использование объединенных таблиц и вызовы основного управления форматами. Это средство разработано с учетом потребностей большинства пользователей в стандартной конфигурации с одновременным сохранением гибкости. Используются иерархические отношения между объектами, состояниями и процессами.

Средство Document Engine управляет поведением с помощью объектов. Ссылка на объект используется при открытии формы; при этом объект определяет поведение для состояния формы (открытие, список, поиск и т. п.). Объекты определяют общее поведение таблицы. В рамках объекта состояние описывает состояние записи в ее жизненном цикле (открытие, список, поиск и т. п.). В рамках состояния выполняются различные процессы в зависимости от действий с записью, инициированных пользователем. Состояния также определяют способ отображения записи в системе и доступные варианты в определенное время или в определенных обстоятельствах. Например, состояния могут определять действие, такое как сохранение, на основе прав доступа пользователя.

Процессы вызываются из состояний на основе действия пользователя. Процессы используют выражения RAD, JavaScript и вызовы существующих приложений RAD для выполнения действий с текущей записью.

### Преимущества модульного подхода

Преимущества модульного построения включают единообразие разработки, сокращение времени разработки и гибкость.

## Единообразие

Тот факт, что система позволяет выполнять все приложения с использованием одних и тех же базовых приложений RAD, позволяет добиться единообразия пакета приложений Service Manager. Базовые функциональные возможности, такие как блокировка, предупреждения, утверждения и использование функциональных возможностей списков записей, работают одинаково во всех модулях, так как все они используют один и тот же базовый код.

## Сокращение времени разработки

Модульное построение средства Document Engine обеспечивает возможность повторного использования существующего кода и процессов.

## Гибкость

Средство Document Engine использует записи процессов в качестве механизма для изменения поведения модулей в рамках приложений Service Manager. Можно создать новый процесс, работа которого отличается от работы базовой системы; при этом не требуется изменять или удалять исходный процесс из системы. Кроме того базовые процессы системы могут быть переопределены пользовательскими процессами для достижения большей гибкости для системных разработчиков при расширении функциональных возможностей Service Manager в соответствии с конкретными требованиями организации.

## Доступ к средству Document Engine.

Чтобы открыть средство Document Engine, выполните следующие действия.

1. Запустите клиент Service Manager и войдите в систему как администратор.
2. В системном навигаторе System Manager щелкните **Расширение функционала**.
3. Щелкните **Средство Document Engine**. Отсюда доступны три основные области для определения объектов, процессов и состояний. Также можно настроить предупреждения и утверждения, которые будут вызываться из объектов, или создать записи конфигурации поиска, которые используются объектами.

## Глава 2

---

### Объекты

Объекты представляют собой базовые наборы определений, которые определяют поведение записей и устанавливают определения и действующие правила. Объекты однозначно соответствуют записям каталога баз данных (dbdict) в Service Manager. Если в таблице отсутствует отдельная запись объекта, механизм работы с документами использует параметры в объекте DEFAULT. Для всех объектов должно быть определено состояние по умолчанию и состояние списка. Если не указано иное, используются состояния по умолчанию db.browse, db.list, db.search и db.view.

**ПРИМЕЧАНИЕ.** Не **изменяйте** и не **удаляйте** запись объекта DEFAULT, так как это приведет к непредсказуемым результатам.

Запись объекта устанавливает определения и действующие правила для поведения таблицы в средстве Document Engine. К ним относятся следующие элементы.

- Приложение, используемое для создания профиля пользователей в этом объекте, которое определяет действия, которые пользователь может выполнять с записями таблицы.
- Записи состояния, используемые в определенных обстоятельствах (дополнительные сведения см. в разделе «Состояния»).
- Имена категории, фазы и страничного файла для объекта.
- Имя записи с номером, которая будет использоваться для объекта.
- Использование блокировки объектом.
- Настройка модификаций записей в таблице.
- Переменные, доступные процессам, использующим этот объект.
- Глобальные списки, постоянно доступные при использовании этого объекта.
- Использование записей активности.
- Обработка предупреждений для этого объекта.
- Обработка утверждений для этого объекта.
- Настройки для очередей работ.
- Возможность указания личных или глобальных представлений и шаблонов по умолчанию.
- Уведомления о добавлении, обновлении и удалении записи в этом объекте.
- Возможность настройки дополнительных вариантов поиска для этого объекта.

Чтобы просмотреть список стандартных объектов, нажмите **Поиск** в форме «Определение объекта».

Чтобы просмотреть список полей и описаний полей для определений объектов, см. раздел ["Форма и поля определения объекта" \(on page 11\)](#).

## Создание и обновление объектов

Чтобы создать объект, выполните следующие действия.

1. Перейдите к средству Document Engine. Описание шагов см. в разделе "[Доступ к средству Document Engine.](#)" (on page 9).
2. Дважды щелкните **Объекты**. Откроется форма «Объекты».
3. Используйте вкладки формы «Объекты», чтобы заполнить поля, необходимые для создания объекта, который будет выполнять требуемые функции. Дополнительные сведения см. в описаниях полей.

Чтобы обновить определение объекта, выполните следующие действия.

1. Перейдите к средству Document Engine. Описание шагов см. в разделе "[Доступ к средству Document Engine.](#)" (on page 9).
2. Введите имя объекта, который требуется обновить, в поле «Объект», или щелкните **Поиск**, чтобы выполнить поиск объекта.

## Форма и поля определения объекта

Ниже приведены описания полей определения объекта.

Имя поля	Описание
Имя файла <i>file.name</i>	Имя dbdict для объекта, т. е. имя dbdict, соответствующее имени данного объекта. (Обязательное поле.)
Общее имя <i>message</i>	Система заполняет это поле на основе политики данных. Это — общее имя для объекта. Общее имя может быть простым именем, например «Заказ на работу».
Уникальный ключ <i>unique.field</i>	Система заполняет это поле на основе данных dbdict. Это — уникальный ключ объекта.

Эта форма содержит также следующие вкладки. Поля на этих вкладках описаны в таблицах описаний полей для соответствующих вкладок.

- "[Описания полей вкладки «Информация объекта»](#)" (on page 12). Вкладка содержит общие свойства и поведение объекта.
- "[Описания полей вкладки «Блокировка»](#)" (on page 14). Вкладка определяет поведение блокировки для объекта.
- "[Описания полей вкладки «Модификации»](#)" (on page 15). Вкладка определяет модификации для объекта.

- "["Описание полей вкладки «Переменные и глобальные списки»" \(on page 15\)](#). Вкладка содержит локальные и глобальные переменные, используемые объектом.
- "["Описания полей вкладки «Операции»" \(on page 16\)](#). Вкладка определяет регистрацию объектов.
- "["Описания полей вкладки «Предупреждения»" \(on page 17\)](#). Вкладка определяет, где устанавливать предупреждения и условия для создания предупреждений.
- "["Описания полей вкладки «Утверждения»" \(on page 18\)](#). Вкладка определяет параметры утверждения для объекта.
- "["Описания полей вкладки «Управление очередями»" \(on page 20\)](#). Вкладка управляет отображением и работой потоков очередей, а также правами создания представлений (списков).
- "["Описания полей вкладки «Представления и шаблоны»" \(on page 22\)](#). Вкладка определяет возможность пользователя создавать глобальные и личные представления и шаблоны для объекта.
- "["Описания полей вкладки «Уведомления»" \(on page 22\)](#). Вкладка определяет уведомления, отправляемые автоматически, для операций добавления, удаления или обновления какой-либо записи объекта.
- "["Описания полей вкладки «Конфигурация поиска»" \(on page 23\)](#). Вкладка управляет доступными параметрами на вкладке «Дополнительные варианты» экрана поиска.

## Описания полей вкладки «Информация объекта»

На этой вкладке указываются общие свойства и поведение объекта.

Ниже приведены описания полей для вкладки «Информация объекта».

Имя поля	Описание
Поле описания <i>desc.field</i>	Краткое описание объекта.
Приложение профиля <i>profile.appl</i>	Приложение RAD, создающее профиль, который определяет возможность выполнения пользователем определенных функций, таких как добавление и удаление. Пример: db.environment. (Обязательное поле.)
Переменная профиля <i>profile.variable</i>	Переменная, которая доступна в любое время при вызове этого объекта без доступа к записи среды. Пример: \$L.env. (Обязательное поле.)
Имя записи с номером <i>number.record</i>	Класс номера для объекта, который доступен из вызова приложения getnumb посредством вызова процесса, управления форматами или RAD. Может использоваться для извлечения порядкового номера в качестве уникального ключа записи. Пример: EXWorkOrder.

Имя поля	Описание
Имя таблицы категорий <i>category.file.name</i>	Имя таблицы, которое связано с таблицей категорий, соответствующей значению категории данного объекта. При отображении записи этого типа, если существует поле с именем <i>category</i> , объект будет выполнять поиск имени таблицы категорий и будет искать запись с соответствующим именем. При успешном завершении поиска система будет хранить имя файла категорий в переменной: <i>\$L.category</i> .
Имя таблицы фаз <i>phase.file.name</i>	Имя таблицы, которое связано с таблицей фаз, связанной с данным объектом (если применимо). При отображении записи этого типа, если существует поле с именем <i>phase</i> , объект выполняет переход к имени таблицы фаз и выбирает запись с соответствующим именем. При успешном завершении поиска система будет хранить имя таблицы фаз в переменной: <i>\$L.phase</i> .
Имя таблицы пейджинга <i>paging.file</i>	Имя таблицы, используемой для хранения страниц. Страницы создаются как полные копии текущей записи перед применением последних обновлений. Это делается при каждом обновлении записи, что позволяет обеспечить подробное отслеживание аудита.
Управление основными форматами <i>master.fc</i>	Имя записи управления основными форматами, если такая запись существует для записи объекта. Управление основными форматами позволяет определить в одной записи инструкции управления форматами, которые относятся ко всем фазам в области, например фазам запроса управления изменениями. Как правило, имя управления основными форматами — это имя <i>dbdict</i> или всех категорий в управлении инцидентами, управлении проблемами или службе поддержки пользователей.
Определение соединения <i>joindef</i>	Имя записи определения соединения, используемой для объединения нескольких таблиц, например <i>joincomputer</i> , или выражение, значение которого является допустимым именем определения соединения, например определение соединения в <i>\$L.category</i> .
Поле статуса <i>statusField</i>	Имя поля, которое содержит сведения о статусе записи.
Поля назначенных <i>assignedToFields</i>	Имя поля, которое содержит поле имени назначенного для данного объекта. Это поле будет использоваться при проверке функциями прав доступа к папкам того, что запись назначена оператору, вошедшему в систему.
Поля рабочей группы <i>workgroupFields</i>	Имя поля, которое содержит поле группы назначенных для данного объекта. Это поле будет использоваться при проверке функциями прав доступа к папкам того, что запись назначена одной из рабочих групп оператора, вошедшего в систему.

Имя поля	Описание
Открытое состояние <i>open.state</i>	Запись определения состояния, которая будет использоваться при открытии новой записи.
Закрытое состояние <i>close.state</i>	Запись определения состояния, которая будет использоваться для закрытия обработки существующей записи.
Состояние списка <i>list.state</i>	Запись определения состояния, которая используется для отображения списка записей.
Состояние по умолчанию <i>default.state</i>	Имя состояния, которое используется в качестве состояния по умолчанию для объекта. Для изменения записи в объекте используется состояние по умолчанию.
Состояние возможности поиска <i>search.state</i>	Запись определения состояния, используемая для поиска.
Состояние возможности просмотра <i>browse.state</i>	Запись определения состояния, которая используется при использовании блокировки для записей. По существу это поле определяет состояние только для чтения, когда запись заблокирована другим пользователем.
Установка состояния вручную <i>manual.states</i>	Массив состояний, которые могут быть использованы с данным объектом в дополнение к состояниям жизненного цикла «открыто», «закрыто», «список», «просмотр», «поиск» и «обзор».

## Описания полей вкладки «Блокировка»

Вкладка «Блокировка» определяет поведение блокировки для объекта. Это значит, что при активном обновлении записи пользователем система не разрешает другому пользователю обновлять эту же запись.

Ниже приведены описания полей для вкладки «Блокировка».

Поле	Описание
Использовать блокировку <i>use.locking</i>	Установите этот флажок, чтобы включить блокировку.
Блокировка во время отображения <i>lock.on.display</i>	Если это поле имеет значение true или содержит условие с значением true, Service Manager выполняет попытку блокировки записи при ее отображении. Чтобы это поле учитывалось, должен быть установлен флажок «Использовать блокировку».

Поле	Описание
Блокировка родительской записи <i>lock.parent</i>	Блокировка текущей записи и ее родительской записи. Например, если этот флажок установлен, и выполняется обновление задачи изменения, также блокируется запрос на изменение для этой задачи.
Поле кода родителя <i>parent.id</i>	Имя поля в текущей записи, которое содержит код родительской записи для блокировки.
Родительское имя файла/объект <i>parent.object</i>	Имя таблицы, содержащей родительскую запись.
Просмотреть переменные <i>watch.variables</i>	Поле «Просмотреть переменные» используется при проверке средством Document Engine наличия изменений записи. Поле «Просмотреть переменные» должно иметь значение NULL при первом отображении записи.

### Описания полей вкладки «Модификации»

Вкладка «Модификации» определяет работу с модификациями для объекта.

Ниже приведены описания полей для этой вкладки.

Имя поля	Описание
Имя таблицы модификаций <i>revision.file</i>	Имя таблицы, в которой должны храниться модификации записей данного объекта.
Макс. число модификаций <i>max.revisions</i>	Общее число модификаций, допустимое для данного объекта. Если значение не указано, верхний предел не устанавливается.

### Описание полей вкладки «Переменные и глобальные списки»

Вкладка «Переменные и глобальные списки» содержит описания локальных переменных и глобальных переменных, используемых объектом.

**Примечание.** Глобальные переменные создаются и хранятся в памяти.

Ниже приведены описания полей вкладки «Переменные и глобальные списки».

Имя поля	Описание
Локальные переменные <i>local.variables</i>	Список локальных переменных, которые могут использоваться в процессах. Локальные переменные определяются в приложении. Локальные переменные назначаются создаваемому объекту и доступны всем процессам и состояниям, связанным с объектом. Другим объектам локальные переменные недоступны.
Глобальные списки <i>global.lists</i>	После создания глобальные списки становятся доступными для всех процессов. Глобальные списки могут создаваться при входе в систему, если они указаны в списке запуска. Глобальные списки доступны при каждом обращении к объекту.

### Описания полей вкладки «Операции»

Вкладка «Операции» используется для определения ведения журнала обновлений (операций) для объекта.

Ниже приведены описания полей для вкладки «Операции».

Имя поля	Описание
Таблица журнала активности <i>activitylog.file.name</i>	Имя таблицы, в которую будут помещены записи журнала операций для объекта.
Переменная списка выбора <i>activity.selection.var</i>	Переменная, используемая в форме обновления для отображения типов операций, которые оператор может выбирать при обновлении записи для определенного объекта.
Копировать связь <i>activity.post.link</i>	Имя связи, которая используется для отправки информации.
Запрашивать обновление, если НЕ создано записи операции? <i>activity.mandatory</i>	Флажок для указания необходимости обновления операции, если установлен.
Обновление поля <i>update.field.var</i>	Поле или переменная, которая содержит обновление операции в форме. Это поле отображается только при наличии флажка <b>Требовать обновление, если запись операции НЕ создана?</b>
Отобразить сообщение <i>activity.mandatory.msg</i>	Сообщение, указывающее необходимость обновления операции. Это поле отображается только при наличии

Имя поля	Описание
	флажка <b>Требовать обновление, если запись операции НЕ создана?</b> .

## Описания полей вкладки «Предупреждения»

Вкладка «Предупреждения» определяет предупреждения и условия создания предупреждений. Эти предупреждения могут использоваться любым объектом с уникальным ключом.

Ниже приведены описания полей для вкладки «Предупреждения».

Имя поля	Описание
Размещение предупреждения <i>alert.location</i>	<p>Размещение предупреждения определяет, где находится или хранится имя определения предупреждения, которое должно быть выполнено. Укажите место хранения определения предупреждения.</p> <p>Запись. Предупреждения хранятся в самой записи.</p> <p>Категория. Предупреждения хранятся в файле категорий, определенном на вкладке «Информация объекта».</p> <p>Фаза. Предупреждения хранятся в записи фазы, определенной на вкладке «Информация объекта».</p> <p>Объект. Предупреждения хранятся в записи объекта. При выборе варианта будет показана таблица массива предупреждений, в которой можно указать используемые предупреждения.</p> <p>Примечание. Таблица объекта должна иметь уникальный ключ, который будет использован для связи с предупреждениями. Таблица с ключом по nulls вместо уникального ключа не может использовать предупреждения.</p>
Условие предупреждения <i>alert.condition</i>	Условие, определяющее необходимость обработки предупреждения. Пример: открыть в \$L.file~=false.
Имя поля предупреждения <i>alert.field.name</i>	Имя поля, которое содержит фактическое имя предупреждения, определенное размещением предупреждения.
Поле статуса предупреждения <i>alert.status.field</i>	Поле в текущей записи, в которое помещается статус предупреждения после обработки предупреждения.
Процесс обновления предупреждения	Имя записи процесса для дополнительных функций, которые выполняются системой после запуска предупреждения.

Имя поля	Описание
<i>alert.update.process</i>	
Регистрировать предупреждения? <i>log.alerts</i>	Если выбран этот вариант, предупреждения после обработки перемещаются в файл Alertlog для сохранения истории предупреждений.
Обрабатывать предупреждения в родительском процессе? <i>alerts.against.parent</i>	Если на вкладке «Блокировка/модификации» выбрано поле «Блокировка родительской записи» и установлен этот флажок, предупреждение также будет регистрироваться для родительской записи при активации.
Выполнить повторный расчет для предупреждений, если <i>alert.recalc</i>	Условия, которые определяют необходимость повторного расчета условий для существующих предупреждений.
Сбросить предупреждения, если <i>alert.reset</i>	Условия удаления существующих предупреждений и повторного расчета всех условий.

## Описания полей вкладки «Утверждения»

На вкладке «Утверждения» задаются параметры утверждения и соответствующие уведомления для объекта. Утверждения определены в файле ApprovalDef.

Ниже приведены описания полей для вкладки «Утверждения».

Имя поля	Описание
Условие утверждения <i>approval.condition</i>	Если условие утверждения имеет значение «истина», для записей объекта используются утверждения.
Размещение утверждения <i>approval.location</i>	Определяет место хранения информации утверждения: запись, фаза, объект или категория.
Имя поля утверждения <i>approval.field.name</i>	Имя поля, которое содержит фактическое имя утверждения, в таблице, определенной в параметре «Местоположение утверждения».
Поле статуса утверждения <i>approval.status.field</i>	Поле текущей записи, в котором хранится статус утверждения.
Группы утверждения <i>approval.groups</i>	Хранит переменную, в которой содержатся группы, членом которых должен быть текущий пользователь, чтобы принимать решения об утверждении для данного объекта.

Имя поля	Описание
Тип утверждения <i>appr.cond.type</i>	<p>Существует четыре predetermined типа утверждения.</p> <p>Должно быть утверждено всеми. Все группы или операторы, определенные в определении утверждения, должны принять решение об утверждении, чтобы статус записи в системе был изменен на «Утверждено». Если только один или некоторые (но не все) группы или операторы принимают решение об утверждении, устанавливается статус записи «В ожидании».</p> <p>Должно быть утверждено одним. Запись считается утвержденной при наличии утверждения любым членом группы утверждающих или оператором.</p> <p>Кворум. Запись утверждается, если решение об утверждении принято большинством группы утверждений.</p> <p>Должно быть утверждено всеми — немедленный запрет. Все группы и операторы должны утвердить запись. Первый отказ вызывает изменение статуса на «Отклонено». Участие всех остальных утверждающих не требуется.</p>
Уведомление об утверждении <i>single.notify.approval</i>	Выбор уведомления, которое будет выполнено при утверждении запроса.
Уведомление о запрете <i>single.notify.denial</i>	Выбор уведомления, которое будет выполнено при отклонении запроса утверждающим.
Уведомление об отмене <i>single.notify.retraction</i>	Выбор уведомления, которое будет запущено при отмене предыдущего действия.
Финальное уведомление об утверждении: <i>final.notify.approval</i>	Выбор уведомления, которое должно отправляться при получении окончательного утверждения.
Финальное уведомление об отказе: <i>final.notify.denial</i>	Выбор уведомления, которое будет отправлено при отклонении запроса.
Утверждение FC <i>appr.fc</i>	Имя записи управления форматами, которая должна выполняться при утверждении.
Процесс утверждения <i>approval.process</i>	Выбор процесса, который будет выполнен при утверждении записи.

Имя поля	Описание
Процесс запрета <i>denial.process</i>	Выбор процесса, который будет выполнен при отклонении записи.
Предварительно утверждать при открытии <i>preapprove.cond</i>	Определяет, должна ли запись утверждаться автоматически.  Если это условие имеет значение «истина», а пользователь является членом одной из групп утверждения, утверждение обрабатывается автоматически. Если пользователь не входит ни в одну из групп утверждения, утверждение не происходит автоматически, и требуется обычный процесс утверждения. По умолчанию имеет значение «истина».
Регистрировать утверждения? <i>log.approvals</i>	Установите этот флажок, чтобы регистрировать историю утверждений в таблице ApprovalLog.
Запрашивать комментарии утверждающего <i>approval.comments</i>	Если флажок установлен, утверждающему выдается запрос комментариев для утверждения.
Групповые утверждения? <i>aggregate.approvals</i>	Если флажок установлен, используются накопительные утверждения.
Выполнить повторный расчет для утверждений, если <i>approval.recalc</i>	Условия, определяющие необходимость повторного расчета условий для существующих утверждений.
Сбросить утверждения, если <i>approval.reset</i>	Условия удаления существующих утверждений и повторного расчета всех условий.

### Описания полей вкладки «Управление очередями»

Вкладка «Управление очередями» управляет отображением очередей и представлений, а также потоковой моделью. Эти же поля доступны в политике данных для файлов без связанной записи объекта; также поля могут быть виртуально встроены в запись datadict из записи объекта, если она существует.

Ниже приведены описания полей для вкладки «Управление очередями».

Имя поля	Описание
Условие управления	Условие, которое позволяет только определенным пользователям просматривать запросы, которые отображают

Имя поля	Описание
<i>scm.condition</i>	записи данного объекта. Пример: browse в \$G.pm.environment.
Управление форматом вывода <i>scm.manage.screen</i>	Выбор формата для отображения представления. В стандартной конфигурации Service Manager используется формат отображения по умолчанию sc.manage.generic, который применяется, если не выбрана какая-либо другая форма. Компания HP не рекомендует изменять формат sc.manage.generic.
Управление представлением по умолчанию <i>scm.inbox</i>	Выбор представления по умолчанию для данной очереди. Указание пользовательского представления для определенного пользователя позволяет настроить список представлений для управления очередями HP Service Manager. Если для пользователя не определено представление, используется представление пользователя по умолчанию.
Управление запросом по умолчанию <i>scm.query</i>	Запрос по умолчанию, который выполняется, если не выбрано представление по умолчанию.
Описание запроса по умолчанию <i>scm.query.name</i>	Имя предыдущего поля. С этим полем может быть связано сообщение. Пример: scmsg(491, "us").
Представление потока -> Поиск? <i>scm.thread.list.edit</i>	Значение «истина» или выражение со значением «истина» для создания нового потока при выполнении поиска.
Формат поиска (если необходимо) <i>scm.search.format</i>	Выбор формата поиска по умолчанию.
Поток поиск -> список? <i>scm.thread.search.list</i>	Значение «истина» или выражение с значением «истина» для нового потока, когда пользователь находит список записей для просмотра.
Список потока -> изменить? <i>scm.thread.list.edit</i>	Значение «истина» или выражение с значением «истина» для нового потока, когда пользователь выбирает запись для просмотра из списка записей.
Представление потока -> изменить? <i>scm.thread.inbox.edit</i>	Значение «истина» или выражение с значением «истина» для открытия нового потока, когда пользователь просматривает существующую запись из очереди.

Имя поля	Описание
Разрешить добавление условия <i>scm.add.condition</i>	Выражение, значение которого соответствует возможности оператора добавить запись.
Добавить/открыть приложение <i>scm.add.appl</i>	Имя приложения, которое вызывается при добавлении или открытии записи.
Имена параметров <i>scm.add.names</i>	Имена параметров, которые передаются приложению, указанному в поле «Добавить/открыть приложение».
Значения параметров <i>scm.add.values</i>	Значения параметров, которые передаются приложению, указанному в поле «Добавить/открыть приложение».

### Описания полей вкладки «Представления и шаблоны»

Вкладка «Представления и шаблоны» определяет возможность пользователя создавать глобальные и личные представления, а также использовать возможности шаблонов.

Ниже приведены описания полей для этой вкладки.

Имя поля	Описание
Можно создавать личные представления. <i>personal.inbox</i>	Условие, которое должно иметь значение true или false. Значение true разрешает пользователю создавать личные представления.
Можно создавать системные представления. <i>global.inbox</i>	Условие, которое имеет значение true или false и определяет возможность пользователя создавать глобальные представления.
Шаблон по умолчанию <i>default.template</i>	Имя шаблона по умолчанию для записей в данной таблице.
Поддерживает шаблоны? <i>supportTemplates</i>	Установите этот флажок, чтобы включить поддержку шаблонов для объекта.

### Описания полей вкладки «Уведомления»

Вкладка «Уведомления» определяет уведомления, отправляемые автоматически для операций добавления, обновления и удаления для объекта.

Ниже приведены описания полей вкладки «Уведомления».

Имя поля	Описание
Добавление <i>notification.add</i>	Выбор уведомления, которое автоматически отправляется при добавлении записи в таблицу.
Обновление <i>notification.update</i>	Выбор уведомления, которое автоматически отправляется при обновлении записи в таблице.
Удаление <i>notification.delete</i>	Выбор уведомления, которое автоматически отправляется при удалении записи из таблицы.

### Описания полей вкладки «Конфигурация поиска»

Вкладка «Конфигурация поиска» управляет доступными вариантами на вкладке «Дополнительные варианты» экрана поиска.

Ниже приведены описания полей для вкладки «Конфигурация поиска».

Имя поля	Описание
Имя таблицы <i>tablename</i>	Имя таблицы, в которой будет выполняться запрос.
Формат поиска <i>searchFormat</i>	Имя субформата, используемого для вкладки «Дополнительные варианты».
Процесс инициализации <i>init.process</i>	Имя процесса, который будет выполнен перед отображением формы поиска.
Разрешить расширенный поиск <i>allowAdvAccess</i>	Условия, которые определяют возможность использования расширенного поиска.

### Вкладка «Определенные запросы»

Вкладка «Определенные запросы» используется для определения инструкций и меток запросов, используемых на вкладке «Дополнительные варианты» экранов поиска (например, на вкладке «Дополнительные варианты» формы «Поиск инцидентов»). Поля определены в таблице SearchConfig.

Ниже приведены описания полей для вкладки «Определенные запросы».

Имя поля	Описание
Код <i>id</i>	Уникальный код (идентификатор) для запроса. Не может содержать специальные символы и пробелы.
Запрос <i>query</i>	Выражение запроса с использованием синтаксиса системного языка. Пример: <code>assignee.name=operator()</code> .
Описание <i>description</i>	Метка для флажка на вкладке «Дополнительные варианты».

### Вкладка «Диапазоны»

Вкладка «Диапазоны» позволяет легко настроить поиск для диапазонов начальной и конечной даты. Для этого щелкните ссылку **Изменить конфигурацию** и определите переменные для начала и окончания, которые будут использоваться в качестве входных значений в форме. Введите эту переменную в столбце «Переменная 1» и используйте столбцы «Поле» и «Оператор 1», чтобы определить выполняемый запрос. Функция «Изменить конфигурацию» также служит для внесения изменений на вкладке «Определенные запросы».

Ниже приведены описания полей для вкладки «Диапазоны».

Имя поля	Описание
Поле <i>fieldName</i>	Поле в таблице для использования в запросе.
Оператор 1 <i>operator1</i>	Оператор сравнения в запросе. (Пример: <code>&gt;=</code> )
Переменная 1 <i>variable1</i>	Переменная, используемая для ввода значения в форме.
Специальный тип <i>specialType</i>	В настоящее время не используется.



## Глава 3

---

### Состояния

Состояния вызываются объектами и определяются процессами. Запись состояния содержит сведения о том, как запись выглядит и работает в определенный период времени.

Определение записи состояния может включать следующие сведения.

- Имя состояния.
- Экран отображения для отображения записи или записей.
- Процесс инициализации, используемый при первом отображении списка.
- Формат, используемый для отображения данных.
- Режим ввода для указания возможности изменения записи пользователем.
- Процесс, запускаемый при включении пользователем определенного параметра отображения (действие по отображению).

Используемая запись состояния определяется тем, сколько записей просматривает пользователь. Существуют стандартные записи состояния для поиска записей, просмотра списка, просмотра одной записи и обзора записи.

### Поиск

При отсутствии записей в текущей переменной файла считается, что пользователь использует режим «поиска». Используется «Состояние поиска», определенное в записи объекта таблицы. По умолчанию используется состояние поиска «db.search».

### Списки записей

При просмотре списка записей используется «Состояние списка», определенное в записи объекта таблицы. По умолчанию используется состояние списка «db.list».

**Примечание.** В некоторых предыдущих версиях Service Manager списки записей назывались списками QBE.

### Просмотр одной записи

При просмотре одной записи эта запись сначала проверяется на наличие поля «Состояние» в dbdict. Если это поле существует и заполнено, содержимое этого поля используется в качестве текущего состояния записи. Если это поле не существует или имеет значение NULL, используется «Состояние по умолчанию», определенное в записи объекта таблицы. Значение по умолчанию для состояния по умолчанию — «db.view».

### Обзор записи

При обзоре записи в режиме «только для чтения» используется «Состояние обзора», определенное в записи объекта таблицы. Следует учитывать, что состояние обзора требуется только для файлов, использующих блокировку. Несмотря на то, что просмотр

записи без прав обновления похож на экран обзора, при этом используется состояние по умолчанию, а не состояние обзора. Не существует состояния обзора по умолчанию.

## Советы по интеграции для приложения отображения

Если экран отображения не имеет связанных событий отображения, система автоматически использует событие отображения `se.default`, которое выполняет `se.lock.object` в случае «OnFormModified». Если в записи параметров отображения установлен флажок «Изменение записи», запись блокируется при нажатии кнопки.

## Создание и обновление состояний

Чтобы создать новое состояние, выполните следующие действия.

1. Откройте средство Document Engine. См. раздел ["Доступ к средству Document Engine." \(on page 9\)](#).
2. В форме «Определение состояния» заполните поля, необходимые для создания состояния, которое будет выполнять требуемые функции. См. описания полей в разделе ["Описания полей определения состояния" \(on page 27\)](#).

Чтобы изменить существующее состояние, выполните следующие действия.

1. Откройте средство Document Engine. См. раздел ["Доступ к средству Document Engine." \(on page 9\)](#).
2. Введите имя состояния, которое требуется изменить, в поле «Состояние», или щелкните **Поиск**, чтобы выполнить поиск состояния.

## Описания полей определения состояния

Запись определения состояния определяет поведение и параметры отображения записи в определенный период времени.

В записи состояния могут быть определены не основные методы или может быть изменено поведение основных методов. Основные методы описаны в разделах «Основные функции в `se.search.engine`», «Основные функции в `se.view.engine`» и «Основные функции в `se.list.engine`». Они включают такие функции, как «Сохранить», «Найти», «Заполнить», «ОК», «Отмена» и «Поиск».

Ниже приведены описания полей определения состояния.

Имя поля	Описание
Состояние <i>state</i>	Имя состояния. (Обязательное поле.)
Экран отображения <i>display.screen</i>	Экран отображения, связанный с состоянием.

Имя поля	Описание
Процесс инициализации <i>init.process</i>	Имя процесса, который запускается перед переходом в данное состояние.
Формат <i>format.name</i>	Формат, в котором открывается запись состояния. Эта форма хранится в записи и может быть переменной или явно заданным значением. Имя формата может быть задано явно или может содержаться в переменной или извлекаться из записи при помощи выражения системного языка.
Режим ввода <i>input.condition</i>	Режим ввода используется для состояний просмотра и определяет для записи режим «только для чтения». Значение false соответствует режиму «только для чтения», а значение true включает возможность изменения. Также можно ввести выражение, которое принимает значения true или false. В противном случае введите значение true или выражение, которое принимает значения true или false.
<b>Не основные методы</b>	
Действие по отображению <i>process.label</i>	Параметр действия, который извлекается из действия отображения после ввода пользователя. Действие по отображению указано в поле действия в параметрах отображения кнопки или команды меню, выбранной пользователем.
Имя процесса <i>valid.process</i>	Имя процесса, который запускается по умолчанию в результате выполнения действия.
Условие <i>process.condition</i>	Выражение, которое принимает значение true для вызова процесса, указанного в поле «Имя процесса».
Сохранить сначала <i>run.save.before</i>	Выбор сохранения процесса перед запуском этого процесса. Введите true, чтобы сохранить процесс перед запуском, в противном случае введите false. Значение по умолчанию — false.



## Глава 4

---

### Процессы

Процессы представляют собой наименьшие возможные дискретные элементы работы, доступные в средстве Document Engine, и являются уровнем работы с данными. Пользователи могут создавать собственные процессы или использовать один из более чем 700 процессов, поставляемых с Service Manager. Определение процесса состоит из исходных выражений, выражений RAD и финальных выражений, которые вводятся на вкладках «Исходные выражения», «Исходный Javascript», «RAD», «Финальные выражения» и «Следующий процесс», и также включают необязательный вызов следующего процесса. Для создания выражений используются стандартные выражения RAD и (или) язык JavaScript.

### Создание и изменение процесса

Чтобы создать процесс, выполните следующие действия:

1. Откройте средство Document Engine. См. раздел ["Доступ к средству Document Engine." \(on page 9\)](#).
2. Используйте вкладки на панели «Процесс», чтобы заполнить поля, необходимые для создания процесса, который будет выполнять требуемые функции. Дополнительные сведения см. в разделе ["Форма «Определение процесса» и описание полей" \(on page 30\)](#).

Чтобы изменить процесс, выполните следующие действия:

1. Откройте средство Document Engine. См. раздел ["Доступ к средству Document Engine." \(on page 9\)](#).
2. Введите имя процесса, который требуется изменить, в поле «Имя процесса», или щелкните **Поиск**, чтобы выполнить поиск процесса.

### Форма «Определение процесса» и описание полей

Форма «Определение процесса» служит для определения новых процессов и изменения существующих процессов. Процессы выполняют код или выражения для выполнения действий, выбранных пользователем.

Ниже приведены описания полей формы определения процесса.

Имя поля	Описание
Имя процесса <i>process</i>	Имя процесса. (Обязательное поле.)

Имя поля	Описание
Сохранить положение курсора <i>save.cursor.position</i>	Установите этот флажок, чтобы восстанавливать положение курсора после выполнения действия (например, после заполнения).
Выполнить стандартный процесс после завершения? <i>run.standard</i>	Если выбран этот вариант, система выполняет стандартный процесс после завершения текущего действия или процесса. Пример: стандартный процесс «Сохранить». Если пользователем создан процесс сохранения и после его выполнения требуется выполнять процесс сохранения из комплекта средства Document Engine, установите этот флажок. Стандартный процесс определен в разделе настоящего документа, в котором описаны основные функции.
Выполнить в окне? <i>run.in.window</i>	Если выбран этот вариант, процесс выполняется в отдельном окне.
Заголовок окна <i>window.name</i>	Если установлен флажок <b>Выполнить в окне?</b> , укажите заголовок окна. Для локализованных заголовков окна могут использоваться выражения <code>scmsg</code> , например <code>scmsg(1980, "us")</code> .

Помимо описанных выше полей эта форма также содержит вкладки для дальнейшего определения процесса. Эти вкладки перечислены ниже.

- ["Вкладка «Исходные выражения»" \(on page 31\)](#)
- ["Вкладка «Исходный Javascript»" \(on page 32\)](#)
- ["Вкладка RAD" \(on page 32\)](#)
- ["Вкладка «Финальные выражения»" \(on page 32\)](#)
- ["Вкладка «Финальный Javascript»" \(on page 33\)](#)
- ["Вкладка «Следующий процесс»" \(on page 33\)](#)

## Вкладка «Исходные выражения»

На вкладке «Исходные выражения» определяются исходные выражения, которые выполняются перед выполнением исходного JavaScript и вызовов RAD, определенных на вкладке «RAD». Исходные выражения записываются с использованием стандартных выражений Service Manager.

## Вкладка «Исходный Javascript»

На вкладке «Исходный Javascript» определяются исходные выражения Javascript, которые выполняются до приложений RAD, определенных на вкладке «RAD».

## Вкладка RAD

Вкладка RAD служит для определения предварительных выражений RAD, вызовов RAD и заключительных выражений RAD, которые выполняются как часть процесса.

Ниже приведены описания полей для вкладки «RAD».

**Примечание.** Эти поля повторяются для каждого приложения RAD, определенного на этой вкладке.

Имя поля	Описание
Выражения, оцениваемые перед вызовом RAD <i>pre.rad.expressions</i>	Выражение, которое должно быть выполнено до выполнения приложения RAD, указанного в поле «Приложение RAD». Все значения параметров, передаваемые приложению RAD, должны быть переменными или выражениями. Переменным должны быть присвоены значения в выражениях, оцениваемых перед вызовом RAD. Пример: <code>\$L.value.name="Test"</code> .
Приложение RAD <i>application</i>	Имя приложения RAD, которое должно быть выполнено.
Условие <i>rad.condition</i>	Условия, при которых приложение RAD должно выполняться (условие имеет значение true) или не должно выполняться (условие имеет значение false).
Имена параметров <i>names</i>	Имена параметров, передаваемых приложению RAD.
Значения параметров <i>values</i>	Значения параметров, передаваемых приложению RAD. Эти значения должны представлять собой переменные или выражения. Строковые значения могут быть переданы в двойных кавычках ("Имя мастера").
Заключительные выражения RAD <i>post.rad.expressions</i>	Выражение RAD, которое должно выполняться после завершения работы приложения RAD.

## Вкладка «Финальные выражения»

На вкладке «Финальные выражения» определяются финальные выражения, которые выполняются после завершения обработки RAD. Финальные выражения записываются с использованием стандартных выражений Service Manager.

## Вкладка «Финальный Javascript»

Инструкции Javascript, определенные на этой вкладке, выполняются после финальных выражений и после приложений RAD на вкладке «RAD».

## Вкладка «Следующий процесс»

Эта вкладка указывает следующий процесс или процессы для выполнения после завершения текущего процесса.

Ниже приведены описания полей вкладки «Следующий процесс».

Имя поля	Описание
Следующий процесс <i>next.process</i>	Имя следующего выполняемого процесса.
Условие <i>process.condition</i>	Условие, связанное с процессом в поле «Следующий процесс», принимающее значение true или false. Пример: true для процесса sm.close.

## Глава 5

---

### Ресурсы средства Document Engine

Средство Document Engine включает локальные переменные и базовые функции, которые могут использоваться любым объектом. Базовые функции определены в стандартной системе по умолчанию для выполнения стандартных действий, доступных пользователю. Например, если пользователь нажимает кнопку «Сохранить» после изменения записи, средство Document Engine обрабатывает запрос и записывает изменения в базу данных. Базовая функция может быть перезаписана в записи состояния для выполнения другого процесса.

### Объект DEFAULT

Любой файл, доступ к которому осуществляется средствами диспетчера Database Manager, автоматически использует правила и процессы, определенные для соответствующего объекта. Если для этого файла не определен объект, используется объект DEFAULT. Объект DEFAULT дублирует функциональные возможности предыдущих версий диспетчера Database Manager.

Системный администратор также может осуществлять доступ к файлу с определенным объектом, используя объект DEFAULT, установив флажок «Режим администрирования» в диспетчере Database Manager. Этот флажок доступен только системным администраторам. Пользователь с мандатным словом «AlwaysAdmin» всегда будет использовать объект DEFAULT при доступе к информации при помощи Service Manager. Назначение мандатного слова пользователю не рекомендуется.

Средство Document Engine использует профили среды. В предыдущих версиях системы системному администратору были предоставлены все права (добавление, обновление, удаление) независимо от параметров управления форматами. При использовании средства Document Engine предоставленные права соответствуют правам, определенным в записи управления форматами. Администраторы, которые предпочитают старый способ предоставления прав, могут просто изменить значение параметра «Приложение профиля» в записи объекта DEFAULT с «db.environment» на «db.environment.sysadmin».

### Приложения RAD

В приложении на основе базы данных пользователь может работать с любым из следующих трех базовых наборов записей.

- Нет записей — при поиске информации.
- Много записей — при просмотре списка с информацией.
- Одна запись — при изменении отдельной записи.

С точки зрения пользователя при работе с данными возможны следующие ситуации.

- Просмотр пустой формы, в которой возможен ввод сведений для поиска (или создания новой записи).

- Работа с обновлением или просмотром отдельной записи.
- Просмотр нескольких записей в виде списка после выполнения поиска.

Средство Document Engine использует три основных приложения RAD в соответствии с описанной концепцией. Используемая подпрограмма RAD определяется числом отображаемых записей.

### **se.search.engine**

Поисковая система используется при отсутствии просматриваемых записей. Основное назначение этой подпрограммы — формирование запроса и выбор записей из требуемой таблицы. Эта подпрограмма также может использоваться для начального ввода информации в пустую запись для добавления новой записи в базу данных. По умолчанию используется состояние db.search.

### **se.list.engine**

Система списков позволяет отображать несколько записей. С помощью системы списков пользователь может выбрать определенную запись из списка или выполнить действие со всем списком записей. По умолчанию используется состояние db.list.

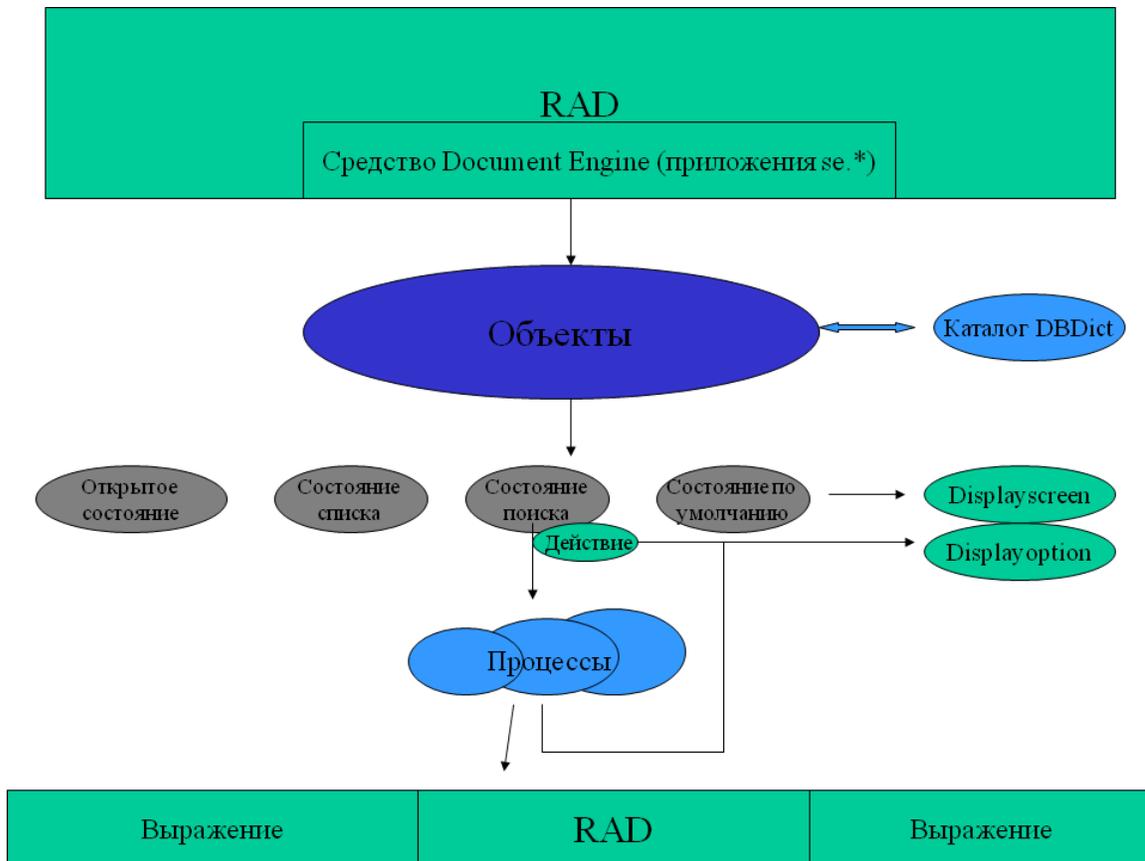
### **se.view.engine**

Система просмотра отображает отдельную запись. Это приложение используется для выполнения действий с отдельной записью, например для обновления или удаления. По умолчанию используется состояние db.view.

**Примечание.** При использовании функциональных возможностей списка записей Service Manager система просмотра используется как для списков, так и для отдельных записей.

## **Поток приложений RAD**

На следующей иллюстрации показан обзор взаимодействия других средств расширения функционала со средством Document Engine.



**Примечание.** Названия всех приложений RAD, относящихся к средству Document Engine, начинаются с символов «se», например se.view.engine. Механизм работы с документами считает каждую запись документом, предоставляемым пользователю.

При отображении файла с использованием средства Document Engine всегда используется одно из трех приложений, указанных выше в этом документе. Экран отображения, параметры отображения и формат определяются текущим состоянием записи. При включении параметра отображения после выполнения стандартных функций отображения средство проверяет «действие» этого параметра в отношении доступных процессов, определенных в текущей записи состояния. Если определен процесс, условие для которого имеет значение «истина», средство выполняет этот процесс для текущей записи (или набора записей). Если действие не определено в записи состояния, средство проверяет, определено ли это действие в качестве «базового процесса» в текущем приложении. Если это так, система выполняет этот базовый процесс для записи. Если нет, то никакие действия не выполняются.

### Советы по интеграции для приложения отображения

Если экран отображения не имеет связанных событий отображения, система автоматически использует событие отображения se.default, которое выполняет se.lock.object в случае «OnFormModified».

Если в записи параметров отображения установлен флажок «Изменение записи», запись блокируется при нажатии кнопки.

## Базовые функции в se.search.engine

Действие по отображению	Описание
back	Выход из записи.
find	Отображение подробных сведений из записи, связанной с полем.
fill	Внесение информации из связанной записи в целевую запись. Данные из поля источника помещаются в поле назначения.
advanced	Запуск процесса расширенного поиска, предоставляющего пользователю больше возможностей для поиска. Как правило эта функция доступна только системным администраторам, но доступ также может быть предоставлен и другим пользователям.
clear	Очистка текущего экрана.
openinbox / inbox	Предлагает пользователю открыть представление, связанное с текущим файлом. Это — предопределенный запрос, который показывает результаты в виде представления.
search	Выполняет стандартный запрос путем создания запроса поиска на основе указанных сведений, а затем показывает результаты в списке записей.
add	Выполняет «добавление» управления форматами и попытку добавления записи в базу данных.
restore	Восстановление содержимого экрана (после выполнения функции «clear»).
irquery	Выполнение запроса IR для текстового поиска.
validitylookup	Выполняет проверку достоверности Service Manager. Проверка достоверности позволяет проверить допустимость указанного значения для определенного поля с помощью записи в таблице достоверности. При помощи проверки достоверности система проверяет соответствие введенного пользователем значения правилам проверки.
expandarray	Функция развертывания массива предоставляет возможности изменения массивов, такие как вставка и удаление строк.
reset	Сброс текущего файла. Это действие удаляет <b>ВСЕ</b> записи в базе данных для таблицы.

Действие по отображению	Описание
regen	Повторное создание индексов текущего файла. Это действие применимо только к таблицам с ключом IR и выполняет только повторное создание IR.
export/unload	Запуск процесса экспорта или выгрузки «без записей». Используется только для выгрузки пустых определений DBDICT без данных.
views	При выборе система предоставляет пользователю список альтернативных форм для отображения текущей записи.
findrevision	Отображение модификаций записей в объекте.
initrevision	Создание модификации записи в объекте.
newsite	
newview	
newTable	
addFilter	Иницируется заказчиком, выполняющим поиск с расширенным фильтром на стандартном экране поиска.
editFilter	Иницируется заказчиком, выполняющим поиск с расширенным фильтром на стандартном экране поиска.
addCompound	Иницируется заказчиком, выполняющим поиск с расширенным фильтром на стандартном экране поиска.
removeSelection	Иницируется заказчиком, выполняющим поиск с расширенным фильтром на стандартном экране поиска.

## Базовые функции в se.view.engine

Действие по отображению	Описание
save	Выполняет обновление в управлении форматами и обновляет запись в базе данных, если обновление допустимо.
add	Выполняет добавление в управлении форматами и добавляет новую запись в базу данных, если новая запись допустима.

Действие по отображению	Описание
ok	Если запись изменена, выполняется сохранение, в противном случае — выход.
reselect	Повторный выбор текущей записи из базы данных (если запись была изменена).
fill	Внесение информации из связанной записи в целевую запись. Информация из поля источника помещается в поле назначения.
find	Отображение подробных сведений из записи, связанной с полем.
next	Перемещение к следующей записи в списке (после проверки на наличие изменений).
previous	Перемещение к предыдущей записи в списке (после проверки на наличие изменений).
back	Возврат в список или форму поиска.
menu	Выход в меню, из которого был осуществлен вызов.
delete	Выполняет удаление управления форматами и предпринимает попытку удаления записи из базы данных после проверки допустимости запроса.
views	При выборе система предоставляет пользователю список альтернативных форм для отображения текущей записи.
print	Печать текущей записи.
printlist*	Печать текущего списка записей.
validitylookup	Выполняет проверку достоверности Service Manager. Проверка достоверности позволяет проверить допустимость указанного значения для определенного поля с помощью записи в таблице достоверности. При помощи проверки достоверности система проверяет соответствие введенного пользователем значения правилам проверки.
export/unload	Стандартные функциональные возможности экспорта и выгрузки Service Manager (для нескольких записей).
massunload*	Функция выгрузки для списка записей.
massadd*	Добавление нового набора записей, который точно дублирует существующий набор записей, за исключением уникальных значений ключей, которые содержат новые данные. После обработки число записей в таблице увеличится в два раза.
massupdate*	Выполняет одинаковые обновления набора указанных полей

Действие по отображению	Описание
	в списке записей.
massdelete*	Удаление указанного набора записей из таблицы.
irquery	Выполнение запроса IR для текстового поиска.
expandarray	Функция развертывания массива предоставляет возможности изменения массивов, такие как вставка и удаление строк.
count*	Подсчет записей в списке записей и отображение итога.
audithistory	Вызов стандартной подпрограммы истории аудита. История аудита определяется полями для аудита, указанными в таблице auditdef.
inbox.save / inbox*	Сохранение текущего запроса в виде представления.
approval.log	Просмотр истории утверждений для текущей записи.
alert.log	Просмотр истории предупреждений для текущей записи.
alerts	Просмотр текущих и запланированных предупреждений для текущей записи.
pagelist / listpages	Список страниц показывает полную историю записи. Если включено использование страниц, при каждом обновлении копия записи записывается в файл страниц. Список страниц показывает полную историю записи по данным файла страниц.
clocks	Отображение записей счетчиков, связанных с текущей записью.
xmlfill	Обработка XML-полей, таких как параметры пользователя в каталоге услуг.
createTemplate	Создание шаблона записи из текущей записи.
applyTemplate	Применение существующего шаблона к текущей записи.

**Примечание.** \* Эти команды доступны только при использовании функций списка записей.

## Базовые функции в se.list.engine

Действие по отображению	Описание
exit (или back)	Возврат к форме поиска (или к вызвавшей форме).
inbox.save / inbox	Запрос пользователя о сохранении текущего запроса в виде нового представления.
count	Стандартные функции подсчета и отображение итогового числа записей.
refresh	Обновление списка с использованием текущего запроса.
big.green	Полное завершение работы текущего модуля. «Большая зеленая стрелка.»
print	Печать списка отображаемых записей.
views	При выборе система предоставляет пользователю список альтернативных форм для отображения текущей записи.
export/unload	Стандартные функциональные возможности экспорта и выгрузки Service Manager (для нескольких записей).
massadd	Добавление нового набора записей, который точно дублирует существующий набор записей, за исключением уникальных значений ключей, которые содержат новые данные. После обработки число записей в таблице увеличится в два раза.
massupdate	Выполняет одинаковые обновления набора указанных полей в списке записей.
massdelete	Удаление указанного набора записей из таблицы.

## Локальные переменные

Локальные переменные начинаются с символов \$L. и существуют только внутри выполняемого в данный момент приложения RAD. Сервер автоматически удаляет локальные переменные при выходе из приложения RAD.

Ниже приведен список стандартных переменных, используемых средством Document Engine.

\$L.action — значение действие отображения из параметра отображения.

\$L.bg — фоновый флаг.

\$L.category — запись категории (при наличии).

\$L.env — текущая запись среды.

\$L.exit — внутренний параметр выхода.

\$L.file — текущая переменная файла.

`$L.file.save` — копия записи в исходном состоянии.

`$L.format` — имя формата, используемого для отображения записи.

`$Irspread` — параметры IR Discovery: 0 — отдаленное сходство; 2 — глубокий поиск; 4 — полное совпадение.

`$L.mode` — режим, в котором находится просматриваемая запись, как правило «add» для создания новой записи, «update» для изменения существующей записи или «close» для завершения обработки существующей записи.

`$L.mult` — флаг, который имеет значение «истина», если в переменной файла `$L.file` несколько записей.

`$L.object` — запись объекта.

`$L.phase` — запись фазы (при наличии).

`$L.sql` или `$L.query` — текущий запрос.

`$L.sort` — текущий порядок сортировки.

`$L.state` — запись состояния, используемая системой (состояние, в котором находится запись).

Переменные, доступные в режиме просмотра (при просмотре отдельной записи).

`$L.fc` — копия подробной записи FormatControl.

`$L.fc.master` — копия основной записи FormatControl.



## Глава 6

---

### Обзор поиска и устранения неполадок

Чтобы успешно выполнить поиск и устранение неполадок в средстве Document Engine, необходимо собрать следующие сведения.

- Используемые dbdict и объект.
- Состояние, в котором находится запись.
- Вызываемый процесс.
- Шаги для воспроизведения неполадки (STR).

### Исследование пути приложения в Document Engine

При поиске и устранении неполадок средства Document Engine, как и при поиске и устранении неполадок всех остальных приложений Service Manager, введите **RTM:3** и **debugdbquery:999** в файле `sm.ini` Service Manager и откройте новое подключение клиента. Если этот пользовательский процесс не является первым процессом, запускающим отладку процессов Document Engine, выбор записей состояний и процессов может быть не отражен в файле `sm.log` в ходе этого отслеживания, однако будут получены полезные сведения в отношении того, какой процесс был запущен.

### Поиск используемого dbdict или объекта

Чтобы определить, какие записи dbdict или объект используются, выполните поиск в файле журнала.

**Пример журнала:**

```
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object get.object select CPU( 0 1411 )
1320 07/18/2006 11:00:36 (0x0129AC08) DBACCESS - Cache Find against file Object found 1
record, query: file.name="pcsoftware"
1320 07/18/2006 11:00:36 RADTRACE 20 [ 1] se.get.object set.access process CPU( 0 1411 )
```

### Поиск состояния записей

Следующие вопрос — это определение состояния записи. Чтобы найти эту информацию, выполните поиск следующих сведений в журнале `sm.log`.

**Пример журнала:**

```
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state select.state select CPU( 0 1491 )
1320 07/18/2006 11:00:48 (0x01292FB0) DBACCESS - Cache Find against file States found 1
record, query: state="pcs.list"
1320 07/18/2006 11:00:48 RADTRACE 10 [ 1] se.get.state exit.normal process CPU( 0 1491 )
```

## Поиск имени процесса

Имя процесса также можно найти путем поиска в журнале `sm.log` фрагмента, показанного в следующем примере.

### Пример журнала:

```
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process select.process select CPU( 0
1542 )
1320 07/18/2006 11:00:50 (0x00B56810) DBACCESS - Cache Find against file Process found 1
record, query: process="upgrade.pcs"
1320 07/18/2006 11:00:50 RADTRACE 20 [ 1] se.call.process run.pre.exp process CPU( 0 1542
)
```

## Исследование ошибок приложений

Процессы вызывают ряд приложений RAD и выполняют ряд выражений, после чего могут быть запущены дополнительные процессы. Если в каком-либо приложении или выражении произошла ошибка, связанная с неправильным символом или логической ошибкой, сведения об этом можно найти в файле `sm.log`.

### Пример журнала:

```
Панель обработки run.pre.exp в RAD se.call.process встретила ошибку в строке 1
(se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Неверный аргумент(2) оператора = (se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Неверный аргумент(2) оператора = (se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Неверный аргумент(2) оператора nullsub (se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Неверный аргумент(2) оператора в(se.call.process,run.pre.exp)
Не удалось оценить выражение (se.call.process,run.pre.exp)
Неустранимая ошибка в приложении: se.search.objects в панели call.list.engine
Неустранимая ошибка в приложении: se.list.engine в панели call.process.1
Неустранимая ошибка в приложении: se.call.process в панели run.pre.exp
```

В этом примере (не соответствующем стандартной настройке) произошла ошибка в процессе `se.call.process` с именем `run.pre.exp`, то есть при определении значений исходных выражений процесса. Чтобы определить, какой процесс вызвал неполадку, просмотрите приведенные выше шаги и определите имя процесса в строке

```
DBACCESS - Cache Find against file Process found 1 record, query: process="upgrade.pcs"
```

Перейдите к записи процесса с именем **upgrade.pcs** и проверьте инструкции на вкладке исходных выражений. В данном конкретном случае выражение содержит слово `nullsub`. Пример выражения для этого теста:

```
$L.icount=nullsub($L.icount, anynumberIwant)
```

Переменная `anynumberIwant` не является допустимым полем, литералом или переменной, поэтому ее необходимо заменить для устранения ошибки.

## Печать значений переменных или результатов выражений

В средстве Document Engine путь в рабочем процессе часто определяется значением, присвоенным полю или переменной. Чтобы определить значение, присвоенное полю или переменной, которое влияет на рабочий процесс, используйте функцию JavaScript `print()` или `log rtecall($L.void=rtecall("log", $L.rc, "message"))` в выражениях RAD. Сообщение может быть сцепленной строкой, например:

```
$L.message="Значение $L.test равно " + $L.test
```

где `$L.test` — переменная, которой присвоено символьное значение.



## Глава 7

---

### Обзор примера заказа на работу

Что такое заказ на работу? Заказ на работу — это определенная задача, назначенная отдельному инженеру, которая определяет некоторые действия, необходимые для разрешения инцидента или решения задачи в другом модуле, если система заказов на работы расширена с включением изменений, проблем или известных ошибок. Приведенный ниже пример создан для Service Manager версии 7.11 и позволяет пользователям создавать, обновлять и закрывать заказы на работу в модуле управления инцидентами. Этот пример легко изменить для работы с другим приложением Service Manager.

В примере создается система заказов на работы, демонстрирующая использование средства Document Engine. Эта система заказов на работы позволяет пользователям создавать заказы на работы для инцидентов и просматривать статус этих заказов на работы. Система заказов на работы также позволяет пользователям просматривать и обновлять заказы на работы из приложения управления инцидентами. Все заказы на работы связываются с определенными инцидентами в системе, при этом инцидент нельзя закрыть до тех пор, пока не будут закрыты все заказы на работы для этого инцидента.

Этот пример предназначен для пользователей, которые расширяют функциональные возможности системы или настраивают ее. Необходимо хорошо понимать работу следующих возможностей расширения функционала.

- Использование каталога Database Dictionary для создания новой таблицы.
- Использование конструктора Forms Designer для изменения готовых форм и создания новых.
- Инструмент создания мастера

В примере заказа на работу рассмотрены следующие этапы.

- Создание нового каталога базы данных (dbdict) при помощи средства dbdict.
- Определение ключевых полей таблицы.
- Создание формы EXWorkOrder для таблицы EXWorkOrder при помощи конструктора Forms Designer.
- Создание файла последовательных номеров.
- Изменение формы с добавлением раскрывающихся списков.
- Создание связи для формы EXWorkOrder.
- Создание формы, которая будет использоваться мастером для сбора информации о заказе на работу.
- Изменение форм закрытия и обновления инцидента.
- Создание псевдонимов для связи заказов на работы с инцидентами.
- Создание определений состояний.

- Создание определений экранов приложения отображения для операций открытия, закрытия и просмотра.
- Создание определений параметров приложения отображения «Добавить», «Отмена», «Заполнить» и «Найти» для открытия, закрытия и просмотра.
- Изменение записи определения процесса в `im.set.close`.
- Тестирование примера заказа на работу.

## Создание таблицы

Чтобы создать новую таблицу, используйте средство `dbdict`. В этом примере будет создана таблица с именем `EXWorkOrder`. Перед тем, как создать таблицу, необходимо знать, какие требуются поля и атрибуты этих полей. В этом примере в таблице хранятся данные заказов на работу, связанных с инцидентами.

Чтобы создать таблицу, выполните следующие действия:

1. В системном навигаторе выберите **Расширение функционала > Каталог Database Dictionary**.  
Откроется форма каталога Database Dictionary.
2. Введите `EXWorkOrder` в поле «Имя файла».
3. Нажмите **Создать**.
4. На вкладке «Поля» введите следующую информацию после нажатия **Создать поле/ключ**.

Поле	Описание
ID	Заполняется из файла номеров. Тип: символьный
RelatedID	Заполняется уникальным кодом связанной записи; пример: номер инцидента) Тип: символьный
status	Используется для хранения статуса заказа на работу. Тип: символьный
initiator	Оператор, который открыл заказ на работу. Тип: символьный
assignee.name	Оператор, которому назначен заказ на работу. Тип: символьный
description	Используется для хранения описания работы согласно заказу на работу. Тип: массив символов

category	Заполняется значением категории связанной записи. Тип: символьный
RelatedCIs	Используется для списка элементов конфигурации (ЭК), связанных с заказом на работу. Тип: массив символов
impact	Тип: символьный
urgency	Тип: символьный
priority	Тип: символьный
closure.code	Тип: символьный
deadline	Дата, к которой должен быть выполнен заказ на работу (заполняется при открытии). Тип: дата/время
est.finish	Оценка даты завершения назначенным оператором. Тип: дата/время
update.action	. Тип: массив символов
closure.comments	. Тип: массив символов

5. После добавления этих полей в таблицу необходимо добавить ключи. Не следует закрывать средство Database Dictionary до завершения этой операции для таблицы **EXWorkOrder**. Подробные сведения см. в разделе [Добавление ключей к таблице](#).

## Добавление ключевых полей к таблице

Чтобы добавить ключи к новой таблице, используйте средство dbdict. После создания таблицы EXWorkOrder необходимо добавить к ней ключи. Ключи используются для индексированного поиска, а также для обеспечения согласованности данных.

Чтобы добавить ключи к таблице, выполните следующие действия.

1. В каталоге Database Dictionary выберите вкладку **Ключи**.
2. Выберите первую доступную запись для нового ключевого поля.
3. Нажмите **Создать поле/ключ**.
4. Введите следующую информацию для каждого создаваемого или изменяемого ключа.

Поле	Описание
ID	Тип: уникальный

RelatedID	Тип: без значений null
RelatedCIs	Тип: нули и дубликаты
assignee.name	Тип: нули и дубликаты

5. Нажмите кнопку **ОК**.

## Создание формы

Перейдите к конструктору Forms Designer, чтобы создать форму для таблицы EXWorkOrder. Для создания формы с помощью мастера выберите «Сведения отдельной записи» и продолжите работу. В результате будет создана базовая форма со всеми полями, которые при необходимости можно изменить. Ниже приведен пример формы EXWorkOrder.

Чтобы создать форму, выполните следующие действия:

1. В системном навигаторе выберите **Расширение функционала > Конструктор Forms Designer**.  
Откроется форма конструктора Forms Designer.
2. Введите EXWorkOrder в поле «Имя формы».
3. Нажмите **Создать**.
4. Нажмите кнопку **Да** для использования мастера форм.
5. Введите EXWorkOrder в поле имени таблицы, для которой будет создана форма.
6. Выберите **Сведения отдельной записи** в качестве типа создаваемой формы.
7. Нажмите кнопку **ОК**.
8. Нажмите **Продолжить**, чтобы использовать в форме поля по умолчанию.
9. Используйте конструктор Forms Designer, чтобы изменить макет формы. Ниже приведен пример формы EXWorkOrder.
10. Для поля «Статус» используются следующие значения.
  - Новый (статус по умолчанию)
  - Открытый
  - Готовый
  - Закрытый
11. Для поля «Код закрытия» используются следующие значения.
  - Реализовано
  - Отменено
  - Откат

Ниже представлен пример формы EXWorkOrder.

WorkOrder

код  Статус

Связанный код

Категория

Общие | Информация о активе | обновление/Закрытие

инициатор  Влияние

Назначенный  Срочность

Срока  Приоритет

Планового окончания

Описание

## Создание копии формы

В конструкторе Forms Designer создайте форму `sc.manage.WorkOrder` путем копирования формы `sc.manage.problem`. Измените поля ввода столбцов таблицы, чтобы использовать поля таблицы `EXWorkOrder`.

Чтобы создать новую форму путем копирования формы, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Конструктор Forms Designer**.  
Откроется форма конструктора Forms Designer.
2. Введите `sc.manage.problem` в поле «Форма».
3. Нажмите кнопку **Поиск**.
4. Выберите `sc.manage.problem.g`.
5. В меню «Подробные параметры» выберите команду **Копировать/переименовать**.
6. Введите `sc.manage.WorkOrder` в поле «Новое имя».
7. Нажмите кнопку **ОК**.
8. В конструкторе Forms Designer обновите поля ввода для столбцов, используя таблицу `EXWorkOrder`.
9.
  - Код инцидента - ID
  - Категория - category
  - Связанный код - RelatedID
  - Статус - status
  - Назначенный - assignee.name

- Описание - description, 1
- Приоритет - priority
- Влияние - impact
- Срочность - urgency

10. Нажмите кнопку **ОК**.

## Создание связи для формы заказа на работу

Одним из преимуществ реляционных баз данных является исключение дублирующихся сведений. Для этого информация об определенном предмете хранится в одном месте или таблице, имеющей ссылки на другие предметы. Связи представляют собой совокупность данных и определений связей с наборов условий, описывающих взаимоотношение связанных элементов информации.

Чтобы создать связь для формы заказа на работу, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Связи**.  
Откроется форма «Файл связей».
2. Введите `EXWorkOrder` в поле «Имя».
3. Добавьте описание в поле «Описание».
4. Нажмите **Создать**.
5. Введите следующую информацию.

Имя исходного поля	Имя файла назначения	Имя поля назначения
initiator	operator	name
assignee.name	operator	name
RelatedCIs	device	logical.name

6. Выберите (выделите) строку инициатора, щелкните правой кнопкой мыши **initiator** и выберите команду **Выбрать строку**.
7. В форме `link.structure.g` введите следующие сведения об инициаторе. Затем повторите шаг 6 для **assignee.name** и **RelatedCIs**, используя приведенные ниже сведения для этих полей.

Поле источника (заполнить для/получить от)	Имя поля назначения (заполнить для/получить от)
initiator	name

assignee.name	name
RelatedCIs	logical.name

8. Нажмите кнопку **Сохранить**.
9. Нажмите кнопку **ОК**.

## Создание файла последовательных номеров

Создайте файл последовательных номеров для формирования последовательных номеров записей в таблице EXWorkOrder.

Чтобы создать файл последовательных номеров, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Последовательные номера**. Откроется форма «Файл последовательных номеров».
2. Введите `EXWorkOrder` в поле «Класс».
3. Введите `1` в поле «Последний номер».
4. Введите `Номер заказа на работу` в поле «Описание».
5. Введите `5` в поле «Длина».
6. Введите `wo` в поле «Префикс».
7. Нажмите кнопку **Добавить**.

## Создание определения объекта

Назначение этого объекта — определение характеристик и поведения объекта EXWorkOrder, который определяет, какие данные должны быть включены в запись заказа на работу, и как система будет обрабатывать заказы на работу.

**Примечание.** Описание полей определения объекта см. в разделе ["Форма и поля определения объекта" \(on page 11\)](#).

Чтобы создать определение объекта, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Объекты**. Откроется форма «Определение объекта».
2. Введите `EXWorkOrder` в поле «Имя файла».
3. Щелкните **Добавить**, чтобы создать запись объекта.
4. На вкладке «Информация объекта» введите следующие сведения.

Поле	Значение
------	----------

Уникальный ключ	Заполняется автоматически (ID)
Общее имя	Заполняется автоматически (EXWorkOrder)
Поле описания	Таблица для хранения заказов на работу, которая может быть связана с любым модулем.
Приложение профиля	db.environment
Переменная профиля	\$L.env
Имя записи с номером	EXWorkOrder
Имя таблицы категорий	category
Управление основными форматами	EXWorkOrder
Поле статуса	status
Поля назначенных	assignee.name
Открытое состояние	EXWorkOrder.open
Закрытое состояние	EXWorkOrder.close
Состояние по умолчанию	EXWorkOrder.view
Состояние возможности поиска	EXWorkOrder.search

5. Нажмите кнопку **Сохранить**.
6. Выберите вкладку **Управление очередями**.
7. На вкладке «Управление очередями» введите следующие сведения.

Поле	Значение
Условие управления	true
Управление форматом вывода	sc.manage.WorkOrder
Управление запросом по умолчанию	assignee.name=operator()
Разрешить добавление условия	false
Описание запроса по умолчанию	Мои заказы на работу

8. Нажмите кнопку **Сохранить**.

9. Нажмите кнопку **ОК**.

## Создание определения процесса инициализации

Эти определения процесса для примера заказа на работу указывают исходные выражения и приложения RAD, которые используются при открытии пользователем записи заказа на работу.

Чтобы создать определение процесса для инициализации, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Процессы**.  
Откроется форма «Определение процесса».
2. Введите `EXWorkOrder.open.initial` в поле «Имя процесса».
3. Нажмите **Добавить**.
4. Введите `$.format="EXWorkOrder"` на вкладке «Исходные выражения».
5. Нажмите кнопку **Сохранить**.
6. Введите следующую информацию на вкладке «RAD».

Поле	Значение
Выражения, оцениваемые перед вызовом RAD	<code>\$.number.record="EXWorkOrder";\$.number.type="string"</code>
Приложение RAD	<code>getnumb</code>
<b>Имена параметров</b>	<b>Значения параметров</b>
<code>name</code>	<code>\$.number.record</code>
<code>index</code>	ID in <code>\$.file</code>
<code>text</code>	<code>\$.number.type</code>

7. Нажмите кнопку **Сохранить**.

## Создание определения экрана приложения отображения

Определения экранов приложения отображения создаются для экранов открытия, закрытия и просмотра форм заказов на работу, чтобы пользователи могли открывать, закрывать и просматривать записи заказов на работу.

Чтобы создать определение экрана приложения отображения для открытия, закрытия и просмотра, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Экраны отображения**.  
Откроется форма определения экрана приложения отображения.
2. Введите `EXWorkOrder.open` в поле «Код экрана».
3. Введите следующую информацию.

Поле	Значение
Код экрана	EXWorkOrder.open
Название	Открытие нового заказа на работу
Формат	\$L.format
Ввод/вывод (если RIO)	true
При параметре 0:	обновить экран
Язык	RUS

4. Нажмите **Добавить**, а затем нажмите **ОК**.

В этом примере также потребуется создать определения экрана приложения отображения для экрана закрытия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Экраны отображения**.  
Откроется форма определения экрана приложения отображения.
2. Используйте следующую информацию.

Поле	Значение
Код экрана	EXWorkOrder.close
Название	Закрытие заказа на работу
Формат	\$L.format
Ввод/вывод (если RIO)	true
При параметре 0:	обновить экран
Язык	RUS

3. Нажмите **Добавить**, а затем нажмите **ОК**.

В этом примере также потребуется создать определения экрана приложения отображения для экрана просмотра.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Экраны отображения**.  
Откроется форма определения экрана приложения отображения.

2. Используйте следующую информацию.

Поле	Значение
Код экрана	EXWorkOrder.view
Название	Просмотр нового заказа на работу
Формат	\$L.format
Ввод/вывод (если RIO)	true
При параметре 0:	обновить экран
Язык	RUS

3. Нажмите **Добавить**, а затем нажмите **ОК**.

## Создание определений параметров приложения отображения

В этой процедуре приведены пошаговые инструкции по созданию определений параметров приложения отображения «добавить», «отмена», «заполнить» и «найти» для следующих определений наряда на работу.

- открыть
- закрыть
- просмотр

При этом одни и те же шаги повторяются четыре раза для каждого определения экрана наряда на работу (открыть, закрыть, просмотр). Однако каждый раз используются различные значения в форме «Определение параметров приложения отображения». В таблицах, приведенных после шагов, указаны значения, необходимые для настройки параметров приложения отображения.

Дополнительные сведения о параметрах приложения отображения см. в разделах справки приложения отображения на сервере справки HP Service Manager в сети.

После создания определений экранов приложения отображения для открытия, закрытия и просмотра необходимо создать параметры приложения отображения для экранов открытия, закрытия и просмотра.

Чтобы создать определение параметров приложения отображения, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Параметры отображения**.  
Откроется форма определения экрана приложения отображения.
2. Введите `EXWorkOrder.open` в поле «Код экрана».
3. Введите следующую информацию.

Поле	Значение
Уникальный код	Создается автоматически (EXWorkOrder.open_add)
Действие	добавление
Параметр GUI	4
Текстовый параметр	4
Банк	1
Условие	true
Метка по умолчанию	Добавление

4. Нажмите **Добавить**.
5. Нажмите кнопку **ОК**.
6. Повторите шаги 1 – 6 для каждого из указанных ниже определений параметров экрана, используя значения, приведенные в таблице.

**Примечание.** В некоторых случаях значение, необходимое для поля «Действие», отсутствует в раскрывающемся списке, но можно ввести нужное значение в поле.

#### Открыть - Отмена

Поле	Значение
Код экрана	EXWorkOrder.open
Уникальный код	Создается системой. (EXWorkOrder.open_cancel)
Действие	back
Параметр GUI	3
Текстовый параметр	3
Банк	1
Условие	true
Метка по умолчанию	Отмена

#### Открыть - Заполнить

Поле	Значение
Код экрана	EXWorkOrder.open

Уникальный код	Создается системой. (EXWorkOrder.open_fill)
Действие	fill
Параметр GUI	9
Текстовый параметр	9
Банк	1
Условие	true
Метка по умолчанию	Заполнить

#### Открыть - Найти

Поле	Значение
Код экрана	EXWorkOrder.open
Уникальный код	Создается системой. (EXWorkOrder.open_find)
Действие	find
Параметр GUI	8
Текстовый параметр	8
Банк	1
Условие	true
Метка по умолчанию	Найти

#### Закрыть - Отмена

Поле	Значение
Код экрана	EXWorkOrder.close
Уникальный код	Создается системой. (EXWorkOrder.close_cancel)
Действие	back
Параметр GUI	3
Текстовый параметр	3

Банк	1
Условие	true
Метка по умолчанию	Отмена

#### Закреть - Закреть

Поле	Значение
Код экрана	EXWorkOrder.close
Уникальный код	Создается системой. (EXWorkOrder.close_close)
Действие	close
Параметр GUI	5
Текстовый параметр	5
Банк	1
Условие	true
Метка по умолчанию	Закреть

#### Закреть - Заполнить

Поле	Значение
Код экрана	EXWorkOrder.close
Уникальный код	Создается системой. (EXWorkOrder.close_fill)
Действие	fill
Параметр GUI	9
Текстовый параметр	9
Банк	1
Условие	true
Метка по умолчанию	Заполнить

#### Закреть - Найти

Метка по умолчанию	Найти
--------------------	-------

Код экрана	EXWorkOrder.close
Уникальный код	Создается системой. (EXWorkOrder.close_find)
Действие	find
Параметр GUI	8
Текстовый параметр	8
Банк	1
Условие	true
Метка по умолчанию	Найти

#### Просмотр - Отмена

Поле	Значение
Код экрана	EXWorkOrder.view
Уникальный код	Создается системой. (EXWorkOrder.view_cancel)
Действие	back
Параметр GUI	3
Текстовый параметр	3
Банк	1
Условие	true
Метка по умолчанию	Отмена

#### Просмотр - Заполнить

Поле	Значение
Код экрана	EXWorkOrder.view
Уникальный код	Создается системой. (EXWorkOrder.view_fill)
Действие	fill
Параметр GUI	9

Текстовый параметр	9
Банк	1
Условие	true
Метка по умолчанию	Заполнить

#### Просмотр - Найти

Поле	Значение
Код экрана	EXWorkOrder.view
Уникальный код	Создается системой. (EXWorkOrder.view_find)
Действие	find
Параметр GUI	8
Текстовый параметр	8
Банк	1
Условие	true
Метка по умолчанию	Найти

#### Просмотр - Сохранить

Поле	Значение
Код экрана	EXWorkOrder.view
Уникальный код	Создается системой. (EXWorkOrder.view_save)
Действие	save
Параметр GUI	4
Текстовый параметр	4
Банк	1
Условие	true
Метка по умолчанию	Сохранить

## Создание определения состояния

Определения состояний для примера заказа на работу указывают используемые процессы и разрешенные действия, когда пользователь открывает, закрывает или просматривает запись заказа на работу.

**Примечание.** Описание полей определения состояния см. в разделе ["Описания полей определения состояния" \(on page 27\)](#).

Чтобы создать определение состояния для открытия, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Состояния**.  
Откроется форма «Определение состояния».
2. Введите следующую информацию.

Поле	Значение
Состояние	EXWorkOrder.open
Экран отображения	EXWorkOrder.open
Процесс инициализации	EXWorkOrder.open.initial
Формат	\$.format
Режим ввода	true

3. Нажмите **ДОБАВИТЬ**.

Чтобы создать определение состояния для закрытия, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Состояния**.  
Откроется форма «Определение состояния».
2. Введите следующую информацию.

Поле	Значение
Состояние	EXWorkOrder.close
Экран отображения	EXWorkOrder.close
Процесс инициализации	EXWorkOrder.close.initial
Формат	\$.format
Режим ввода	true

3. Нажмите **ДОБАВИТЬ**.

Чтобы создать определение состояния для просмотра, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Состояния**.  
Откроется форма «Определение состояния».
2. Введите следующую информацию.

Поле	Значение
Состояние	EXWorkOrder.view
Экран отображения	EXWorkOrder.view
Процесс инициализации	EXWorkOrder.view.initial
Формат	\$.format
Режим ввода	true

3. Нажмите **ДОБАВИТЬ**.

Чтобы создать определение состояния для поиска, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Состояния**.  
Откроется форма «Определение состояния».
2. Введите следующую информацию.

Поле	Значение
Состояние	EXWorkOrder.search
Экран отображения	db.search
Процесс инициализации	EXWorkOrder.view.search
Формат	\$.format
Режим ввода	true

3. Нажмите кнопку **ОК**.

## Добавление кнопки закрытия заказа на работу

Когда пользователь завершает задачу или задачи, необходимо закрыть заказ на работу. В этой процедуре описано добавление кнопки закрытия в форму EXWorkOrder и обновление состояния заказа на работу с установкой значения «Закрыто». В этой процедуре выполняется добавление определения параметра приложения отображения, добавление определения состояния для EXWorkOrder.view и добавление определения процесса для EXWorkOrder.close.

**Примечание.** В этой процедуре не рассматривается создание проверки для этой операции.  
Для добавления проверки используется управление форматами.

Чтобы создать определение параметра приложения отображения для EXWorkOrder.view, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Параметры отображения**.  
Откроется форма определения экрана приложения отображения.
2. Введите EXWorkOrder.view в поле «Код экрана».
3. Введите следующую информацию.

Поле	Значение
Уникальный код	Создается автоматически (EXWorkOrder.view_close)
Действие	close
Параметр GUI	5
Текстовый параметр	5
Банк	1
Условие	true
Метка по умолчанию	Закреть

4. Нажмите **Добавить**.
5. Нажмите кнопку **ОК**.

Чтобы обновить определение состояния EXWorkOrder.view, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Состояния**.  
Откроется форма «Определение состояния».
2. В поле «Состояние» введите **EXWorkOrder.view** и нажмите кнопку **Поиск**. Откроется форма определения состояния EXWorkOrder.view.
3. Добавьте следующие сведения, чтобы обновить форму определения состояния.

Поле	Значение
Действие по отображению	close
Имя процесса	EXWorkOrder.close
Условие	true

4. Нажмите **Сохранить** и затем **ОК**.

Чтобы добавить запись определения процесса для EXWorkOrder.close, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Процессы**.  
Откроется форма «Определение процесса».
2. Введите EXWorkOrder.close в поле «Имя процесса».
3. Нажмите **Добавить**.
4. На вкладке «Исходные выражения» введите следующее выражение:
  - status in \$L.file="Closed"
  - \$L.mode="closed"
5. Нажмите кнопку **Сохранить**.
6. На вкладке «RAD» введите следующие сведения.

Поле	Значение
Выражения, оцениваемые перед вызовом RAD	\$L.EXaction="update"
Приложение RAD	se.base.method
Условие	true
<b>Имена параметров</b>	<b>Значения параметров</b>
file	\$L.file
подсказка	\$L.EXaction
second file	\$L.file.save
record	\$L.fc
second.record	\$L.object
boolean1	false

7. Нажмите **Сохранить** и затем **ОК**.

## Создание мастера для заказа на работу

В этом примере для создания заказа на работу используется мастер, так как такой подход облегчает задачу пользователя, а функциональные возможности уже доступны в системе.

В этом примере используется средство создания мастеров, которое позволит пользователям создать заказ на работу в модуле управления инцидентами. Когда пользователь создает заказ на работу из инцидента, мастер запрашивает у пользователя необходимые сведения и автоматически заполняет некоторые поля заказа на работу.

Чтобы создать мастер для формы заказа на работу, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Мастера**. Откроется форма «Информация мастера».
2. Введите `Создание заказа на работу - 1` в поле «Имя мастера».
3. Щелкните **Добавить**, чтобы создать запись мастера.
4. На вкладке «Информация мастера» введите следующие сведения.

Поле	Значение
Краткое описание	Создание нового заказа на работу.
Заголовок окна	Создание заказа на работу
Название	Создание заказа на работу
Начальный узел	Установите флажок (значение true), чтобы указать, что мастер является первым в серии мастеров, если имеется серия записей мастеров.

5. Нажмите кнопку **Сохранить**.
6. На вкладке «Выбор файла» формы «Информация мастера» выберите вкладку **Выбор файла \$L.file**.
7. На вкладке «Выбор файла \$L.file» введите следующие сведения.

Поле	Значение
Создать запись	true
типа	EXWorkOrder

8. Нажмите кнопку **Сохранить**.
9. На вкладке «Использование» введите следующие сведения.

Поле	Значение
Использование мастера	Щелкните <b>Запрос ввода данных пользователем</b> в разделе «Использование мастера».
Субформат для просмотра	Введите <code>createWO.assigneeAndCIs</code>
Экран отображения	Введите <code>wizard.display</code>
Включить вариант «Готово»	Выберите (установите значение true), чтобы в форме мастера отображалась кнопка «Готово».

10. Нажмите кнопку **Сохранить**.

11. На вкладке «Действия» введите следующие сведения.

Поле	Значение
Выполнить действия в	Выберите «Текущий файл (\$L.file)».
Выполняемые действия > Выражения	initiator in \$L.file=operator() status in \$L.file="New" RelatedID in \$L.file=number in \$relatedRec category in \$L.file=category in \$relatedRec impact in \$L.file=initial.impact in \$relatedRec urgency in \$L.file=severity in \$relatedRec priority in \$L.file=priority.code in \$relatedRec
Отобразить запись (записи) при завершении?	Щелкните <b>Отобразить запись (записи) при завершении?</b>
Режим	Выберите «Добавление».

12. Нажмите кнопку **Сохранить**.

13. На вкладке «Выражения отмены» введите следующие сведения.

Поле	Значение
Выражения, выполняемые при отмене	cleanup(\$relatedRec)

14. Нажмите **Сохранить** и затем **ОК**.

Назад + Добавить Поиск Найти Заполнить

Обновлена запись в файле format.

### Информация мастера

◆ Информация мастера | ◆ Выбор файла | ◆ Использование | ◆ Действия | ◆ Обмен сообщениями | ◆ Переменные | ◆ Следующий мастер »2

Имя мастера:

Краткое описание:

Заголовок окна:

Название:

Приглашение:

Точечный рисунок:

Глобальные списки:


Запустить узел?

## Добавление записи определения процесса

Запись определения процесса указывает на то, как система реагирует на действие пользователя. Определение процесса использует выражения RAD, JavaScript и вызовы существующих приложений RAD для выполнения действий с текущей записью, в данном случае — с записью наряда на работу.

Чтобы добавить запись определения процесса, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Процессы**  
Откроется форма «Определение процесса».
2. Введите **create.WorkOrder** в поле «Имя процесса».
3. Нажмите **Добавить**.
4. На вкладке «Исходные выражения» введите следующее выражение:
  - `$L.void=fduplicate($relatedRec, $L.file)`
  - `$relatedCIs={}`
5. Нажмите кнопку **Сохранить**.
6. На вкладке «Исходный Javascript» введите следующее выражение.

```
sys-  
tem.vars.$relatedCIs=system.library.BSGFunctions.getMembers(system.vars.$L_  
file.affected_item, false, 3)
```

**Примечание.** Это выражение необходимо ввести на одной строке.

7. Нажмите кнопку **Сохранить**.

8. На вкладке «RAD» введите следующие сведения.

Поле	Значение
Выражения, оцениваемые перед вызовом RAD	Введите два следующих выражения: <code>\$L.wiz.name="Create Workorder-1"</code>  <code>if (not null(logical.name in \$L.file)) then (\$relatedCIs=insert (\$RelatedCIs, 1, 1, logical.name in \$L.file))</code>  <b>Примечание.</b> Вводите это выражение на одной строке. Также при вводе выражения следует убедиться, что после слова <code>insert</code> нет пробела. Например, указанное выше выражение продолжается после слова «insert» следующим образом: <code>insert(\$RelatedCIs, 1, 1, logical.name in \$L.file))</code>
Приложение RAD	wizard.run
Имена параметров	name
Значения параметров	\$L.wiz.name
Имена параметров	text
Значения параметров	\$L.exit
Условие	true

9. Нажмите кнопку **Сохранить**.

10. На вкладке «Финальные выражения» введите `cleanup($relatedRec)`.

11. Нажмите **Сохранить** и затем **ОК**.

## Создание формы ввода для мастера

Эта форма — первая форма, которая отображается в мастере заказа на работу. Пользователь вводит сведения, которые необходимы мастеру для создания записи заказа на работу. Для создания этой формы используется конструктор Forms Designer. В этом примере используется имя формы `createWO.assigneeAndCIs`.

Чтобы создать форму ввода для мастера, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Конструктор Forms Designer**.  
Откроется форма поиска/создания новой формы конструктора Forms Designer.
2. Введите `createWO.assigneeAndCIs` в поле «Имя формы».

3. Нажмите **Создать**. Для этой формы не требуется использование мастера конструктора Forms Designer.
4. Создайте в форме следующие поля ввода.
  - Описание (введите описание необходимой работы)
  - Срок (введите дату выполнения для этого заказа на работу)
  - Назначенный (введите назначенного сотрудника для этого заказа на работу)
  - ЭК (введите ЭК для этого заказа на работу)
5. Свойства этой формы должны включать следующие.
  - Ввод: **RelatedCIs**
  - Список значений: **\$relatedCIs**
  - Сортируемые: установлено.
6. Нажмите кнопку **Сохранить**.

На следующей иллюстрации показан пример формы createWO.assigneeAndCIs для формы ввода мастера EXWorkorder.

The screenshot shows a software window titled "Конструктор форм: createWO.assigneeAndCIs". The window contains a form with the following elements:

- A toolbar at the top with icons for grid, mouse, and other design tools.
- Four input fields with labels in Russian:
  - "введите описание необходимой работы:" followed by a large text area.
  - "введите дату выполнения для этого заказа на работу:" followed by a date picker.
  - "введите назначенного сотрудника для этого заказа на работу:" followed by a dropdown menu with a search icon.
  - "введите ЭК для этого заказа на работу:" followed by a large text area.
- Two arrows (right and left) positioned between the two large text areas.
- A scroll bar on the right side of the form area.
- A scroll bar at the bottom of the window.

## Изменение форм закрытия и обновления инцидента

Для примера заказа на работу необходимо изменить форму закрытия и обновления инцидента таким образом, чтобы можно было просматривать заказы на работу, назначенные инциденту, а затем открывать их двойным щелчком из инцидента для чтения или редактирования. Используйте конструктор Forms Designer, чтобы обновить формы *IM.update.incident* и *IM.close.incident*. Для этого необходимо добавить вкладку и форму на вкладке в формах *IM.update.incident* и *IM.close.incident*.

Чтобы изменить форму обновления инцидента, выполните следующие действия.

**Примечание.** Используйте эту же общую процедуру для обновления формы закрытия инцидента.

1. В системном навигаторе выберите **Расширение функционала > Конструктор Forms Designer**.  
Откроется форма поиска конструктора Forms Designer.
2. Введите **IM.update.incident** в поле «Имя формы».
3. Нажмите кнопку **Поиск**.
4. Нажмите кнопку **Проектирование**.
5. Добавьте новую вкладку к существующему блокноту.
6. Задайте заголовок вкладки «Заказ на работу».
7. Добавьте субформат со следующими свойствами.
  - Вывод на экран: установлено.
  - Формат: WorkOrder.vj.
  - Виртуальное соединение: установлено.
  - Отобр. пустого объекта: установлено.
  - Отобр. в виде таблицы: установлено.
  - Ввод: number.WO.vj
8. Нажмите кнопку **Сохранить**.
9. Используйте конструктор Forms Designer для создания формы WorkOrder.vj.
  - Код заказа на работу: ID
  - Срок заказа на работу: deadline
  - Статус заказа на работу: status
  - Назначенный оператор заказа на работу: assign.name
  - Описание заказа на работу: description
10. Повторите шаги 1 – 8 для **IM.close.incident**.

**Примечание.** Для виртуального соединения можно использовать форму WorkOrder.vj, созданную на шаге 9.

На следующих иллюстрациях показаны примеры добавляемых вкладок.

# Руководство Document Engine

## Глава 7: Обзор примера заказа на работу

The screenshot shows a web form titled "Конструктор форм: IM.update.incident". The form is divided into several sections for data entry:

- Сведения об инциденте:** Includes fields for "Код инцидента", "Статус", "Контакт", "Местоположение", "Затронутая услуга", "Затронутые ЭК" (with a checkbox for "ЭК функционирует (нет прог)"), "Начало простоя", "Окончание простоя", "Сервисный договор", and "Целевая дата SLA".
- Группа назначенных:** Includes "Назначенный", "Поставщик", and "Запись поставщика".
- Категория:** Includes "Категория", "Область", and "Подобласть".
- Воздействие:** Includes "Воздействие", "Срочность", and "Приоритет".
- Название и Описание:** A large text area for "Название" and "Описание".
- Код закрытия и Решение:** Fields for "Код закрытия" and "Решение".

On the right side, there is a navigation menu with options: "метрики KPI", "Предупреждения", and "Заказ на работу".

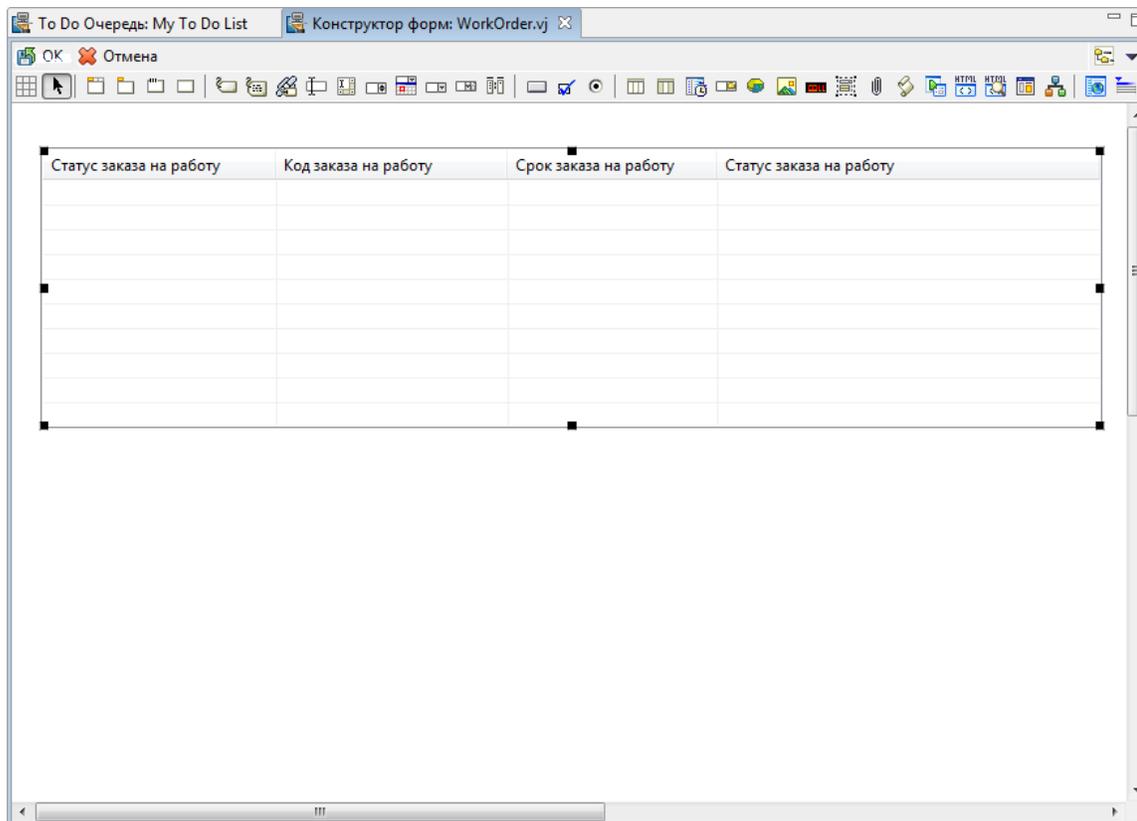
The screenshot shows a web form titled "Конструктор форм: WorkOrder.vj". The main content is a table with the following headers:

Статус заказа на работу	Код заказа на работу	Срок заказа на работу	Статус заказа на работу

Below the table, a "Modifica elenco" dialog box is open, showing a list of items to be modified:

- Имя вещи
- Вещи commenti
- Статус заказа на работу
- Код заказа на работу
- Срок заказа на работу
- Статус заказа на работу

The dialog box includes buttons for "Aggiungi", "Sottrai", "Elimina", "Sposta su", "Sposta giù", "OK", and "Annulla".



## Создание псевдонима в таблице probsummary для связи

В примере заказа на работу пользователям требуется доступ к заказу на работу из записи инцидента, если с инцидентом связан заказ на работу. Для этого необходимо создать псевдоним в таблице probsummary и затем использовать этот псевдоним для создания связи между таблицей probsummary и таблицей EXWorkOrder.

Чтобы добавить псевдоним к таблице probsummary, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Каталог Database Dictionary**.  
Откроется форма каталога Database Dictionary.
2. Введите probsummary в поле «Имя файла».
3. Нажмите кнопку **Поиск**.
4. На вкладке «Поля» выберите поле номера и щелкните **Изменить поле/ключ**.
5. Щелкните **Создать псевдоним** и введите number.WO.vj в качестве имени; укажите символьный тип.
6. Нажмите кнопку **ОК**.

Чтобы связать таблицу EXWorkOrder с таблицей probsummary, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средства расширения функционала > Связи**.  
Откроется форма «Файл связей».
2. Введите `probsummary` в поле «Имя».
3. Нажмите кнопку **Поиск**.
4. Щелкните после последней записи, чтобы создать пустую строку для новой записи.
5. В поле «Источник» введите `number.WO.vj`.
6. Выберите всю новую строку и щелкните **Выбрать строку**.
7. Введите следующую информацию.

Поле	Значение
Поле (из/источник)	<code>number.WO.vj</code>
Файл (к/назначение)	<code>EXWorkOrder</code>
Поле (к/назначение)	<code>RelatedID</code>
Запрос	<code>\$query</code>
Выражения	<code>\$query="RelatedID=\'"+number in \$File+\'"</code>

8. Нажмите **Сохранить** и затем **Назад**.
9. Повторите шаги для создания строки связи для поля кода (ID).
10. Введите следующую информацию.

Поле	Значение
Поле (из/источник)	<code>ID</code>
Файл (к/назначение)	<code>EXWorkOrder</code>
Поле (к/назначение)	<code>ID</code>
Запрос	<code>\$query</code>
Выражение	<code>\$query="ID=\'"+nullsub(cursor.field.contents(), "xxx")+\'"</code>

11. Нажмите **Сохранить** и затем **Назад**.

## Изменение определения процесса `im.set.close`

Этот процесс необходимо изменить, чтобы пользователь не мог закрыть инцидент при наличии открытых заказов на работу для этого инцидента.

Чтобы изменить запись определения процесса, выполните следующие действия.

1. В системном навигаторе выберите **Расширение функционала > Средство Document Engine > Процессы**.

Откроется форма «Определение процесса».

2. Введите **im.set.close** в поле «Имя процесса».
3. Нажмите кнопку **Поиск**.

4. На вкладке «Исходный Javascript» введите следующий код JavaScript.

```
var WO=new SCFile ("EXWorkOrder")

var FoundOpenWO=WO.doSelect ("RelatedID=\""+system.vars.$L_
file.number + "\""+ " and status ~=\"\" + "Closed" + "\"")

if (FoundOpenWO == RC_SUCCESS)
{
system.vars.$openWO=true;
}
else
{
system .vars.$openWO=false;
}
}
```

5. Нажмите кнопку **Сохранить**.
6. На вкладке «RAD» введите следующие сведения.

Поле	Значение
Примечание. В некоторых полях вкладки «RAD» указаны предопределенные значения, изменять которые не нужно.	
Приложение RAD. Выберите пустой раздел ниже раздела с вызовом приложения RAD us.consume.wrapper и введите следующие сведения.	
Выражения, оцениваемые перед вызовом RAD	\$L.text="Имеются открытые заказы на работу. Инцидент нельзя закрыть в настоящее время."
Приложение RAD	apm.mb.ok
Условие	\$openWO=true
Имена параметров	text
Значения параметров	\$L.text

7. Нажмите кнопку **Сохранить**.
8. На вкладке «Финальные выражения» введите следующие сведения.

- if (\$openWO=true) then (\$L.exit="badval")
- \$L.exit="closestate"

9. Нажмите **Сохранить** и затем **ОК**.

## Тестирование примера заказа на работу

После завершения создания системы заказов на работу необходимо проверить правильность ее работы. Чтобы проверить основные функции примера заказа на работу, выполните следующие действия.

- В управлении инцидентами найдите или создайте инцидент.
- Используйте меню «Параметры» или кнопку на панели инструментов, чтобы создать заказ на работу. В меню «Параметры» должна быть показана команда «Создать заказ на работу».
- В мастере заказа на работу введите данные для создания заказа на работу для инцидента.
- Добавьте заказ на работу к инциденту и сохраните изменения инцидента.
- Откройте инцидент для изменения и на вкладке «Заказ на работу» измените заказ на работу.
- Сохраните изменения.
- Снова откройте инцидент, измените заказ на работу и закройте его.
- Теперь должна появиться возможность закрытия инцидента.
- Повторите эти шаги, создавая два заказа на работу для инцидента.
- Закройте только один заказ на работу и попробуйте закрыть инцидент. Система должна выдать сообщение об ошибке, указывающее на наличие открытых заказов на работу для инцидента и невозможность его закрытия.
- Закройте все заказы на работу для инцидента. Теперь должна появиться возможность закрытия инцидента.

