



**Система обновлений терминального ПО.
Программное обеспечение
«Процессинговый центр Pay-logic»
Руководство пользователя**

АННОТАЦИЯ

Описывает технологию обновления ресурсов и компонентов терминального ПО. Содержит инструкции по эффективному использованию системы обновлений программного обеспечения «Процессинговый центр Pay-logic»

Версия руководства: 3.3

Руководство актуально для ТПО версий 5.1xx, 7.1xx, РМА 6.3x

2008–2023 ООО «Софт-Лоджик», г. Барнаул, Россия

Данный документ входит в комплект поставки программных продуктов.

Права использования данного документа предусмотрены соответствующим лицензионным договором.

ООО «Софт-Лоджик»

656006, г. Барнаул, Малахова ул., дом 146в

Тел: (3852) 72-27-27

© *Soft-logic*

Web: <https://soft-logic.ru/>

Mail: info@soft-logic.ru

ОГЛАВЛЕНИЕ

ИСТОРИЯ ИЗМЕНЕНИЙ.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.9.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.0.....	5
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.1.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.2.....	6
ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.3.....	6
1 ОСНОВНЫЕ ТЕРМИНЫ.....	8
2 ОСНОВНЫЕ СОКРАЩЕНИЯ.....	10
3 ВВЕДЕНИЕ.....	11
4 СИСТЕМА ОБНОВЛЕНИЙ.....	13
4.1 НАЗНАЧЕНИЕ И АРХИТЕКТУРА СИСТЕМЫ ОБНОВЛЕНИЙ.....	13
4.2 АРХИТЕКТУРА СИСТЕМЫ ОБНОВЛЕНИЙ.....	15
4.3 СТРУКТУРА КАТАЛОГОВ СИСТЕМЫ ОБНОВЛЕНИЙ.....	17
5 РАБОТА С СИСТЕМОЙ ОБНОВЛЕНИЙ.....	35
5.1 НАСТРОЙКА СИСТЕМЫ ОБНОВЛЕНИЯ В КАБИНЕТЕ.....	35
5.2 АЛГОРИТМ РАБОТЫ СИСТЕМЫ ОБНОВЛЕНИЙ.....	37
5.3 ПОДГОТОВКА ОБНОВЛЕНИЯ НА СЕРВЕРЕ.....	43
5.4 НАСТРОЙКА АВТОМАТИЧЕСКОЙ ЗАГРУЗКИ ФОРМ И СЦЕНАРИЕВ В КАБИНЕТ.....	44
6 РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ СИСТЕМЫ ОБНОВЛЕНИЙ.....	46
6.1 ОБЩИЕ ПОЛОЖЕНИЯ.....	46
6.2 РЕКОМЕНДУЕМЫЙ ПОРЯДОК ОБНОВЛЕНИЯ ТОЧЕК.....	47
6.3 АВТОМАТИЗАЦИЯ ОБНОВЛЕНИЯ ТОЧЕК.....	48
6.4 ПЕРЕДАЧА ПРОИЗВОЛЬНЫХ ФАЙЛОВ НА ТЕРМИНАЛ.....	51
6.5 СКАЧИВАНИЕ ФАЙЛОВ БЕЗ ПЕРЕЗАГРУЗКИ.....	53
6.6 УДАЛЕНИЕ ФАЙЛОВ С ТЕРМИНАЛА.....	54
7 ОСОБЕННОСТИ ОБНОВЛЕНИЯ ТОЧЕК ДРУГИХ ТИПОВ.....	56
7.1 ОСОБЕННОСТИ ОБНОВЛЕНИЯ ТОЧЕК ТИПА «РМА».....	56

ПРИЛОЖЕНИЕ А. СКРИПТ ПОДГОТОВКИ ФАЙЛОВ ДЛЯ ОБНОВЛЕНИЯ.....	58
--	----

ИСТОРИЯ ИЗМЕНЕНИЙ

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 2.9

Дата публикации: 08.01.2019.

Изменение	Раздел
Улучшения в версии 4.7.5:	
Для обновления РМА версии 6 через систему обновлений используется каталог <корневой каталог СО>/rma6	7.1

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.0

Дата публикации: 22.02.2019.

Изменение	Раздел
Дополнения в документации к ранее имевшемуся функционалу:	
Добавлен раздел по архитектуре систем обновлений	4.2

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.1

Дата публикации: 19.08.2022.

Изменение	Раздел
Улучшения в версии 5.123:	
Добавлена возможность разнесения СО по версиям ОС	4.2, 4.3

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.2

Дата публикации: 28.12.2022.

Изменение	Раздел
Улучшения в версии 7.129:	
Добавлена распаковка архивов после загрузки обновлений	6.1

ИЗМЕНЕНИЯ В ВЕРСИИ ДОКУМЕНТА 3.3

Дата публикации: 15.11.2023.

Изменение	Раздел
Улучшения в документации:	
Актуализирована информация по каталогу с обновлениями ТПО 5 версии с	4.3

Изменение	Раздел
подкаталогами для ОС Linux и Windows	

1 ОСНОВНЫЕ ТЕРМИНЫ

Агент — юридическое лицо, владеющее точками приема платежей или действующее как дистрибьютор, не имея собственных точек приема платежей и выполняющее функции по приему платежей или предоставлению возможности другим лицам проводить платежи через себя.

Карта сдачи — представляет собой виртуальные денежные средства, которые сохраняются на сервере, при внесении пользователем денежных средств в сумме большей, чем требуется для проведения платежа. Один из возможных вариантов реализации механизма сдачи.

Личный кабинет агента — специализированный веб-сайт, который предназначен для управления сетью точек приема платежей и просмотра финансовой статистики.

Номерная емкость — файл с указанием привязки номера к оператору связи региона.

Объект процессинга — модель данных процессинга содержит в себе описание структур объектов и их отношений. Каждый объект обладает набором определенных свойств и для него могут быть указаны значения этих свойств.

Платёж — расчёт за полученную услугу. Платёж, осуществляемый абонентом (клиентом) при помощи наличных денежных средств или банковской карты.

Платежная система (Поставщик) — юридическое лицо, самостоятельно предоставляющее возможность оплачивать свои услуги, либо юридическое лицо, предоставляющее возможность оплачивать услуги других компаний.

Провайдер платёжных сервисов (от англ. payment service provider) — компания, которая предоставляет онлайн-сервисы по осуществлению электронных платежей различными способами, включая смарт-карты, банковские платежи, такие как банковские операции и другие. Некоторые провайдеры платёжных сервисов предоставляют различные инновационные сервисы: платёжные системы, включая платежи наличными, электронные кошельки, предоплаченные карты или ваучеры и т. д.

Произвольное свойство объекта — определяется объекту дополнительно независимо от заданной структуры и взаимосвязей в модели данных. Недоступно в фильтрах, выводится в отчетах по SQL-запросу к БД процессинга.

Сервис — услуга, по которой принимаются платежи в платёжной системе.

Сервис провайдера — сервис в процессинге, жестко привязанный к поставщику услуги по идентификационному коду в системе поставщика.

Собственный поставщик — поставщик, заведенный на уровне текущего агента, имеющий договор с этим агентом.

Субагент — дочерний агент, находящийся в агентской сети на один уровень ниже по отношению к текущему агенту.

Точка — программно-аппаратное устройство, с помощью которого (через которую) осуществляется прием платежа.

DEF-code — негеографические телефонные коды мобильных операторов.

Fraud-контроль — система контроля безопасности, используемая процессингом для отслеживания и блокировки мошеннических операций, несанкционированных действий. Fraud-мониторинг осуществляется платежной системой и позволяет по заданным критериям отслеживать и анализировать подозрительные операции.

2 ОСНОВНЫЕ СОКРАЩЕНИЯ

АБС — автоматизированная банковская система.

БД — база (базы) данных.

ЛК — личный кабинет.

ОС — операционная система.

ПО — программное обеспечение.

ПС — платежная система.

РМА — рабочее место кассира.

ТПО — терминальное программное обеспечение.

ТПП — точка приема платежей.

СУБД — система управления базами данных.

URL (англ. Uniform Resource Locator) — стандартизированный способ записи адреса ресурса в сети Интернет.

3 ВВЕДЕНИЕ

Руководство предназначено для администраторов и пользователей ПО «Процессинговый центр Pay-logic», имеющих опыт развертывания и эксплуатации процессинга, знакомых с принципами обслуживания ТПО и знающих основы работы в ОС Linux и Windows.

Система обновления предназначена для удаленного обновления ПО на терминальных точках, РМА, мобильных точках. Руководство рассматривает принципы работы системы обновлений, возможности системы и типовые решаемые с её помощью задачи.

Описание настройки разделов кабинета агента приведено в документах:

1. [«Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#) — описывает настройку и администрирование ПО «Процессинговый центр Pay-logic».
2. [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) — описывает использование кабинета агента ПО «Процессинговый центр Pay-logic» в части управления агентами и пользователями, ТПП и отслеживания состояния оборудования.
3. [«Провайдеры. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) — описывает назначение и процесс настройки раздела «Провайдеры» кабинета агента. Функционал позволяет осуществлять управление поставщиками услуг ПС, а также просматривать информацию об их финансовом состоянии, например, сведения об изменении остатков на счетах провайдеров, или прогнозировать сумму необходимых оборотных средств. Кроме того, раздел позволяет осуществлять управление универсальным БД-шлюзом, который предназначен для снижения количества вводимых пользователям реквизитов платежа.
4. [«Диспетчерская. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) — описывает назначение и процесс настройки раздела «Диспетчерская» кабинета агента. Раздел содержит полную информацию по

каждому платежу. Подсистема поиска позволяет проводить анализ по суммам платежей в различных разрезах, строить графики, формировать отчёты в Excel.

5. [«Реклама. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) — описывает назначение и процесс настройки раздела «Реклама» кабинета агента, с помощью которого возможно использовать различные виды рекламных услуг на экранах терминалов, проводить рекламные кампании и промоакции.

6. [«Интеграция с автоматизированной банковской системой. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) — описывает интеграцию ПО «Процессинговый центр Pay-logic» с АБС.

Описание настройки ТПО приведено в документах:

1. [«Терминальное ПО 5 версии. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).
2. [«Терминальное ПО 7 версии. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

Описание работы РМА приведено в документе [«РМА приема платежей с помощью ПК. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

Описание шаблонов чека платежа приведено в документе [«Шаблон чека. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

Процедуры создания сервисов описаны в документах:

1. [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#).
2. [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

4 СИСТЕМА ОБНОВЛЕНИЙ

4.1 НАЗНАЧЕНИЕ И АРХИТЕКТУРА СИСТЕМЫ ОБНОВЛЕНИЙ

Система обновлений предназначена для централизованной и дистанционной передачи на ТПП обновленных и актуальных ресурсов, конфигурационных файлов, рекламных роликов и баннеров, исполняемых файлов, а также служебных файлов и данных.

Процессинговый центр предполагает ведение широко разветвленной агентской сети, при этом терминалы могут принадлежать как одному, так и разным юридическим лицам. Может возникнуть необходимость в том, чтобы ресурсы на терминалах различных агентов отличались полностью или частично, например исполняемые модули — должны быть одинаковыми, или наоборот — версии исполняемых модулей должны быть разные на разных терминалах. Система обновлений позволяет наряду с централизацией ведения обеспечить и необходимое разделение (разграничение) обновлений по агентам, и тем самым обеспечить полную или частичную независимость при обновлении клиентского ПО разных агентских сетей.

Система обновлений позволяет разграничить обновления по ВОС, версии клиентского ПО.

Со стороны сервера технически система обновлений представлена сервером, отдающим файлы обновлений на терминальные точки, и настройками обновления терминальных точек, которые задают параметры системы обновлений для агента в кабинете процессингового центра.

С клиентской стороны система представлена модулем обновления и сценарием обновления, встроенными в терминальное клиентское ПО.

Модуль обновления разработан для надежного функционирования в условиях нестабильной связи и малой пропускной способности каналов. Клиентское ПО при выполнении обновления скачивает только необходимые файлы, то есть только те файлы, которые с момента последнего обновления точки изменились. Поддерживается «докачка» файлов при разрыве соединения. Таким образом, если из-за нестабильности

сетевого соединения файл скачался наполовину, при восстановлении связи весь файл не будет скачиваться заново.

Система обновлений также может быть настроена для обновления парка ТПП по частям в соответствии с расписанием (актуально при большом числе точек).

4.2 АРХИТЕКТУРА СИСТЕМЫ ОБНОВЛЕНИЙ

Сервер системы обновлений может быть настроен таким образом, что настроена одна основная система обновлений и множество других систем обновлений, в которых часть каталогов ссылаются на основную систему обновления, а часть — отличается.

Пример такой архитектуры приведен на рисунке 4.2.1.

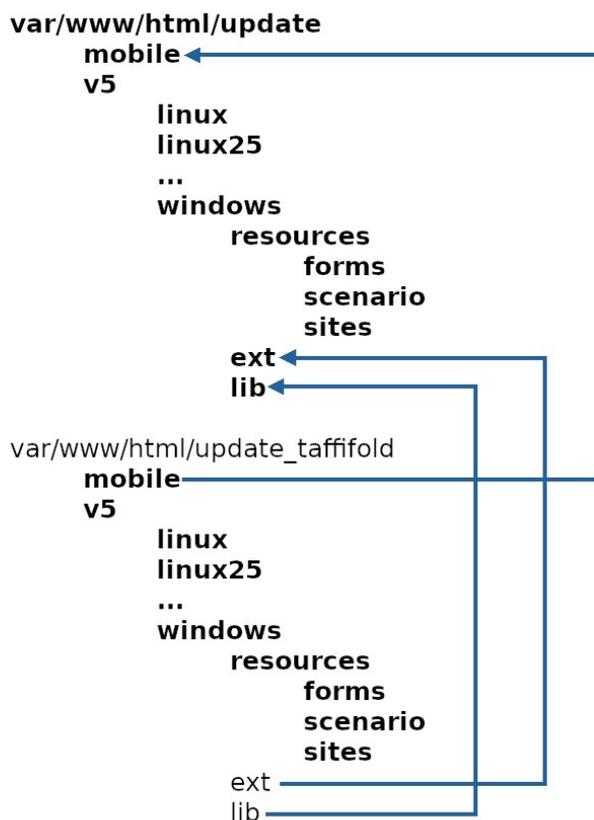


Рисунок 4.2.1 — Архитектура системы обновлений со ссылкой на основную систему обновлений

В такой архитектуре одна из систем является базовой, а другие — содержат символические ссылки на каталоги основной системы обновления (если содержимое этих каталогов совпадает, например, каталоги `./ext`, `./lib` и т. д.)

Чтобы создать символическую ссылку в ОС Linux необходимо ввести команду в формате:

```
ln -s /path_to_folder /path_from_folder_link
```

где `/path_to_folder` — каталог, на который создается ссылка

`/path_from_folder_link` — каталог, в котором создается ссылка

После того, как файлы будут выложены в основную систему обновлений, необходимо запустить генератор обновлений (утилиту `FileListGenerator.jar`) для каждой системы обновлений.

Система, из которой обновляются терминалы, указывается для каждой группы терминалов в кабинете процессинга в разделе «Точки — Параметры точек».

4.3 СТРУКТУРА КАТАЛОГОВ СИСТЕМЫ ОБНОВЛЕНИЙ

Сервер системы обновлений представляет собой web-сервер Apache. Сама система обновлений представляет собой один из каталогов в корневой директории веб-сервера. Одна система обновлений может содержать:

1. Каталоги с обновлениями ТПО для каждой базовой версии (с подкаталогами по операционным системам).
2. Каталог с обновлениями для Andorid-приложений.
3. Каталог с системами обновлений рабочего места кассира (РМА).
4. Каталог с содержимым подсистемы рекламы (рекламных баннеров).
5. Каталог с содержимым рекламного контента (роликов) для второго монитора.

```
update
  ad
  android
  android_im7
  banners
  mobile
    CalcBannerCRC.jar
    crc
    crc1
    crc7
    defectors.csv
    defectors.csv.gz
    patch7
    patch7.gz
  rma
    img
    lib
      AbsoluteLayout.jar
      ...
      mail.jar
      ...
    resources
      ...
      rma.jar
      list.crc
      list.xml
      FileListGenerator.jar
      update.build
  rma6
    ...
```

```
v5
  linux
    ext
    img
    resources
    ...
    list.crc
    list.xml
    FileListGenerator.jar
    update.build
  linux25
  linux32
  ...
  windows
    ext
    img
    resources
    ...
    list.crc
    list.xml
    FileListGenerator.jar
    update.build
  windows7
  windows10
  ...
v7
  linux
    ext
    gui
    java.properties
    lib
    res
    run.sh
    scheme
    list.crc
    list.xml
    FileListGenerator.jar
    update.build
    update.sh
  windows
    ext
    gui
    java.properties
    lib
    res
    run.sh
    scheme
    list.crc
    list.xml
    FileListGenerator.jar
    update.build
    update.js
    update.sh
```

Каталог на сервере, в котором находится одна система обновлений, называется корневым каталогом системы обновлений. Здесь:

1. Каталог **«ad»** — каталог с роликами.
2. Каталог **«android»** — каталог с обновлениями для клиентов, использующий Android-приложения.
3. Каталог **«android»** — каталог с обновлениями для клиентов, использующий Android-приложения с 7 input-моделью.
4. Каталог **«banners»** — каталог с файлами (картинками) рекламных баннеров.
5. Каталог **«mobile»** — каталог с файлами для работы БДПН (подробнее работа БДПН описана в документе [«Провайдеры. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#)). Для 7 версии ТПО файлы для работы БДПН должны размещаться в каталоге *<корень системы обновлений>/v7/каталог ОС/res/module/input/mobile/<ISO-код страны>/*.
6. Каталог **«rma»** — каталог с обновлениями рабочего места кассира версии 5.
7. Каталог **«rmaб»** — каталог с обновлениями рабочего места кассира версии 6.
8. Каталог **«v5»** — каталог с обновлениями ТПО версии 5 с подкаталогами для ОС Linux и Windows. Для версий ТПО 5.123 и выше возможно разносить файлы по версиям ОС. Например,

```
v5
  linux
  linux30
  linux29
  windows
  windows10.0
  window5.1
  windows8.1
```

В первую очередь при обновлении проверяется каталог, соответствующий версии ОС, используемой на терминале. Если в нем файлы не найдены, то они будут взяты из общего каталога ОС: **windows** или **linux** в зависимости от ОС терминала.

9. Каталог «v7» — каталог с обновлениями ТПО версии 7 с подкаталогами для ОС Linux и Windows.

Из структуры системы обновлений видно, что внутри корневого каталога находятся каталоги с обновлениями для соответствующих видов терминальных точек (РМА, терминала), причем обновления для РМА не отличаются для различных ОС, а обновления для ТПО — разделяются, поскольку часть файлов (скрипты, отвечающие за запуск, перезапуск и обновление ПО, а также некоторые другие файлы) для ТПО в зависимости от типа ОС терминала — различаются.

Если в парке терминалов присутствуют разные базовые версии ТПО (например, 5 и 7 версии), то для терминалов с разной базовой версией ПО должен быть создан отдельный подкаталог на одном уровне с v5.

Каталог с обновлением ТПО (внутри версии и ОС) и каталог с обновлением РМА внутри имеют схожую структуру и одинаковые принципы устройства. Внутри каталога обновления находятся:

1. Каталоги с ресурсами (формами, сценариями).
2. Каталоги с изображениями (картинками для кнопок, иконками).
3. Каталоги с библиотеками ТПО.
4. Отдельные файлы ТПО.
5. Иные файлы, которые необходимо обновлять на терминале.
6. Файлы, обеспечивающие работу системы обновлений.

На рисунке 4.3.1 приведена структура каталогов типичной системы обновлений для ТПО версии 5, ОС Linux.

Состав файлов и каталогов копирует структуру домашнего каталога ТПО, с той лишь разницей, что в каталоге обновления присутствуют только те файлы, которые необходимо в ходе обновления зачислять на терминал. То есть файлы терминала, которые не требуют централизованного обновления — не помещаются в каталог системы обновлений.

Index of /update/v5/linux

Name	Last modified	Size	Description
 Parent Directory		-	
 FileListGenerator.jar	22-Feb-2013 12:56	8.7K	
 ext/	23-Jan-2014 17:38	-	
 lib/	23-Jan-2014 17:42	-	
 list.crc	01-Jun-2015 15:43	29	
 list.xml	01-Jun-2015 15:43	2.2K	
 run.sh	23-Jan-2014 16:54	1.7K	
 update.build	01-Jun-2015 15:43	2	

Apache/2.2.15 (CentOS) Server at ng.pay-logic.ru Port 80

Рисунок 4.3.1 — Файлы системы обновлений v5 для Linux

На рисунке 4.3.2 приведена структура каталогов типичной системы обновлений для ТПО версии 5, ОС Windows.

Index of /update/v5/windows

Name	Last modified	Size	Description
 Parent Directory		-	
 FileListGenerator.jar	22-Feb-2013 12:56	8.7K	
 ext/	06-Feb-2014 11:42	-	
 list.crc	06-Feb-2014 11:48	29	
 list.xml	06-Feb-2014 11:48	1.9K	
 update.build	06-Feb-2014 11:48	1	

Apache/2.2.15 (CentOS) Server at ng.pay-logic.ru Port 80

Рисунок 4.3.2 — Файлы системы обновлений v5 для Windows

Структура каталогов ТПО версии 5 выглядит следующим образом:

```
ad
banners
bin
data
ext
fonts
img
  smoke
    buttons
    elements
    icon
    messages
    params.properties
...
java_dump
jni
  32
  64
...
key
lib
log
  add
  card
  cashin.log
  dbg.log
  pay.log
resources
  dispensers
  dmi
  forms
  i18n
  info
  keyboards
  mobile
  scenario
  utility
  validators
  verify-data
  key-server.properties
  mail-server.properties
  modules.properties
  params.properties
sequence
templates
  default
    collection
  logo
trademark
update
adplay.sh
capacity.xml
configure.sh
configuration.xml
```

```
java.properties  
maintenance.rules  
offer.html  
run.sh  
scan.sh  
serialnumbers.store  
stop.sh
```

1. Каталог **ad** содержит рекламные ролики. Например, *playlist*; *test-0000.mpeg*; ...
2. Каталог **banners** содержит рекламные баннеры. Например, *1046342357.png*; ...; *704954814_2.png*.
3. Каталог **bin** содержит скрипты, используемые при работе ТПО: *calib*; *chmtu*; *gencalib*; *g-maintatance*; *g-ppp*; *gprsconfig*; *gprsconfig_alternate*; *ifacelist*; *mountflash*; *processlist*; *script.sh*; *unmountflash*; *ziplog*.
4. Каталог **data** содержит данные, необходимые для работы ТПО: справочники, очереди платежей и сообщений и др.: *cash-in* (*bvr.obj*, *bvr.obj.tmp*, *cash-acceptor-stat.obj*, *cvr.obj*, *cvr.obj.tmp*); **dispenser (bill)**; **network**; *commands.obj*; *promo.obj*; *queue.obj*; *reference.obj*.
5. Каталог **ext** содержит библиотеки «Soft-logic».
6. Каталог **fonts** содержит шрифты. Например, Century\ Gothic\ Bold\ Italic.ttf, Ubuntu-R.ttf, ...
7. Каталог **img** содержит каталоги с ресурсами конкретного интерфейса. Например, каталог **smoke** содержит ресурсы интерфейса Smoke — различные изображения, файлы с текстовками и др. Внутри каталога для каждого интерфейса содержится каталог **icon** с изображениями. Для интерфейса Smoke в каталоге **<корень ТПО>/img/smoke/icon/service/** содержатся изображения размером «124x80» для кнопок по сервису. Как правило, сами кнопки круглые. В каталоге **<корень ТПО>/img/smoke/icon/group/** содержатся изображения размером «100x100» для кнопок по группе сервисов. Как правило, сами кнопки прямоугольные. Желательно, что фон изображения был прозрачным. Кроме того, в каталоге каждого интерфейса размещен файл *params.properties*, в котором задаются цветовые схемы для интерфейса.
8. Каталог **java_dump** предназначен для снятия дампа java-машины в случае нештатной ситуации, настраивается в *java.properties*.

9. Каталог *jni* содержит нативные библиотеки. Библиотеки для 32-разрядной системы содержатся в каталоге **32**, для 64-разрядной — **64**. Библиотеки, не отличающиеся в зависимости от разрядности системы находятся непосредственно в каталоге *jni*.

10. Каталог *key* содержит ключи ТПП.

11. Каталог *lib* содержит сторонние библиотеки.

12. Каталог *log* содержит журналы работы ТПО.

13. Каталог *resources* содержит ресурсы ТПО:

1) каталог *dispensers* содержит ресурсы для диспенсеров;

а) каталог *bill* — каталог содержит профили для диспенсеров купюр:

- каталог **2** содержит профили для двухбуксовых диспенсеров купюр, например, *default_usd, ...*;
- каталог **4** содержит профили для четырехбуксовых диспенсеров купюр, например, *default, ...*;

2) каталог *dmi* содержит информацию об оборудовании, полученную командой *dmidecode*. Например, в файле *barcode* содержится информация о сканере штрих-кодов;

3) каталог *forms* содержит формы для сервисов, работающих на универсальном обработчике;

4) каталог *i18n* содержит текстовки, используемые для интернационализации форм/сценариев. Подробнее об интернационализации форм и сценариев в документах [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#), [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) соответственно: *input_en.properties, ..., input_uk.properties*;

5) каталог *keyboards* для каждого интерфейса содержит служебные файлы для описания клавиатуры, отображаемой клиенту;

- 6) каталог **mobile** содержит ресурсы для работы мобильного обработчика;
 - 7) каталог **scenario** содержит сценарии для сервисов, использующих усовершенствованный провайдер;
 - 8) каталог **utility** содержит служебные формы: **checkcopy.xml** — форма копии чека, **console.xml** — форма входа в сервисное меню, **paystatus.xml** — форма статуса платежа, **srchprv.xml** — форма поиска сервиса;
 - 9) каталог **validators** содержит файлы с внешними валидаторами;
 - 10) каталог **verify-data** содержит файлы для использования комплекс-валидатора;
 - 11) файл **key-server.properties** — содержит адрес веб-сервера для генерации ключей на точке, **mail-server.properties** — содержит параметры почтового сервера, **modules.properties** — содержит указание на классы, необходимые для работы подключаемых устройств, **params.properties** определяет параметры java-машины, подробнее в документе [«Терминальное ПО 5 версии. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).
14. Каталог **sequence** содержит файлы для работы генератора уникальных последовательностей.
 15. Каталог **templates** содержит шаблоны чеков.
 - 1) каталог **default** — шаблоны чеков по умолчанию;
 - а) каталог **collection** — шаблон чека инкассации;
 - 2) каталог **logo** содержит логотипы, которые могут быть распечатаны на чеке.
 16. Каталог **trademark** содержит иконки для configurатора ТПО.
 17. Каталог **update** содержит обновления, доступные для ТПО.
 18. Файл **adplay.sh** представляет собой скрипт запуска плеера рекламы.
 19. В файле **capacity.xml** находятся номерные емкости.
 20. Файл **configure.sh** представляет собой скрипт запуска configurатора.
-

21. Файл `configuration.xml` представляет собой конфигурационный файл терминала. Содержит настройки соединения, оборудования и т. д. Генерируется конфигуратором.
22. Файл `java.properties` содержит параметры java-машины. Подробнее в разделе 8.2.
23. Файл `maintenance.rules` задает правила на удаление файлов.
24. Файл `offer.html` представляет собой шаблон оферты с клиентом системы.
25. Файл `run.sh` представляет собой скрипт запуска терминала.
26. Файл `scan.sh` представляет собой скрипт для поиска подключенных к терминалу устройств. Находит только купюроприемник, пин-пад, кард-ридер.
27. Файл `serialnumbers.store` содержит серийные номера устройств, которые задаются в конфигураторе на вкладке «Серийные номера».

При запуске ТПО5 поддерживается автоматическая распаковка архивов в форматах `.bz2`, `.gz`, `.tar` и `.tar.gz` при помощи разархиваторов `VzipDecompressor`, `GzipDecompressor` и `TarDecompressor` соответственно.

В версии 5.99 добавлен разархиватор `ZipDecompressor`, который позволяет автоматически распаковывать `zip`-архивы.

При этом архивы в форматах `.bz2` и `.gz` после обработки будут удалены. Архивы в форматах `.zip`, `.tar` и `.tar.gz` будут находиться в каталогах и после разархивирования.

На рисунке 4.3.3 приведена структура каталогов типичной системы обновлений для ТПО версии 7, ОС Linux, на рисунке 4.3.4 — для ТПО версии 7, ОС Windows.

Index of /update/v7/linux

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 ext/	20-Apr-2015 09:52	-	
 gui/	24-Jun-2014 13:51	-	
 java.properties	11-Jul-2014 10:22	13	
 lib/	08-Apr-2015 16:22	-	
 list.crc	20-Apr-2015 10:44	30	
 list.xml	20-Apr-2015 10:44	18K	
 res/	14-Dec-2014 11:14	-	
 run.sh	13-Apr-2015 11:24	2.6K	
 scheme/	10-Apr-2015 16:51	-	
 update.build	20-Apr-2015 10:44	2	
 update.sh	10-Apr-2015 16:45	343	

Apache/2.2.15 (CentOS) Server at ng.pay-logic.ru Port 80

Рисунок 4.3.3 — Файлы системы обновлений v7 для Linux

Index of /update/v7/windows

<u>Name</u>	<u>Last modified</u>	<u>Size</u>	<u>Description</u>
 Parent Directory		-	
 ext/	20-Apr-2015 09:52	-	
 gui/	24-Jun-2014 13:51	-	
 java.properties	11-Jul-2014 10:22	13	
 lib/	08-Apr-2015 16:29	-	
 list.crc	20-Apr-2015 10:44	29	
 list.xml	20-Apr-2015 10:44	18K	
 res/	16-Apr-2015 09:42	-	
 scheme/	10-Apr-2015 16:51	-	
 update.build	20-Apr-2015 10:44	2	
 update.js	14-Apr-2015 11:59	2.2K	
 update.sh	10-Apr-2015 16:45	343	

Apache/2.2.15 (CentOS) Server at ng.pay-logic.ru Port 80

Рисунок 4.3.4 — Файлы системы обновлений v7 для Windows

Структура каталогов ТПО версии 7 выглядит следующим образом:

```
bin
content
  ad
  banners
data
  cash-in
  disp (bill)
  flags
  network
  save
  screenshot
  sequence
  keeper.obj
  networkstat.obj
  promo.obj
  queue-d.pst
```

```
queue-d.tmp
queue-f.pst
queue-f.tmp
references.pst
ext
  drivers
  keeper
  player
  advanced-hardware.jar
  commlogic.jar
  ...
  utils.jar
  version.jar
gui
  fonts
  skin
    blues
    ...
    skin.properties
jni
  32
  64
key
lib
log
  add
    adv.log
    ...
    utility.log
  cashin.log
  dbg.log
  pay.log
res
  dev
    disp
      bill
        2
        4
    dmi
  module
    i18n
    input
      advanced
      card
      connector
      stub
      sequence.bin
      forms
      i18n
      res
    ds
    exchange
    mobile
      205
      ...
```

```
401
...
universal
keeper
maps
paytpl
refs
keeper.properties
menu
utility
system
templates
trademark
maintenance.rules
update
adplay.sh
configure.sh
conf.xml
javapath
java.properties
run.sh
```

1. Каталог **bin** содержит скрипты, используемые при работе ТПО. Например, *calib*; *chmtu*; *gencalib*; *g-maintatance*; *g-ppp*; *gprsconfig*; *gprsconfig_alternate*; *ifacelist*; *mountflash*; *processlist*; *script.sh*; *unmountflash*; *ziplog*.

2. Каталог **content** содержит рекламный контент:

- 1) каталог **ad** содержит рекламные ролики. Например, *test-0000.mpeg*; ... ;
- 2) каталог **banners** содержит рекламные баннеры. Например, *1046342357.png*;; *704954814_2.png*.

3. Каталог **data** содержит данные, необходимые для работы ТПО: справочники, очереди платежей и сообщений и др.:

- 1) каталог **cash-in** содержит файлы, необходимые для работы купюроприемника;
- 2) каталог **disp** содержит файлы, необходимые для работы диспенсера;
- 3) каталог **network** содержит файлы, необходимые для работы сетевого устройства;
- 4)

4. Каталог **ext** содержит библиотеки «Soft-logic»:

- 1) каталог **drivers** содержит библиотеки по работе с оборудованием, драйвера устройств. Например, *device-manual-driver.jar*, *softlogic-hardware-core.jar*, *softlogic-hardware.jar*, *softlogic-hardware-scanner.jar*;
 - 2) каталог **keeper** содержит библиотеки, имеющие отношение к работе кошелька. Например, *keeper-impl.jar*, *keeper.jar*;
 - 3) каталог **player** содержит библиотеки, имеющие отношение к работе плеера рекламы;
 - 4) ...
5. Каталог **gui** содержит ресурсы интерфейса:
- 1) каталог **fonts** содержит шрифты. Например, Century\ Gothic\ Bold\ Italic.ttf, Ubuntu-R.ttf;
 - 2) каталог **skin** содержит ресурсы конкретных интерфейсов. Например, каталог **blues** содержит ресурсы интерфейса Bluesphere2 — различные изображения, файлы с текстовками и др. Кроме того, в каталоге каждого интерфейса размещен файл *skin.properties*, в котором указано название текущего интерфейса
6. Каталог **jni** содержит нативные библиотеки. Библиотеки для 32-разрядной системы содержатся в каталоге **32**, для 64-разрядной — **64**. Библиотеки, не отличающиеся в зависимости от разрядности системы находятся непосредственно в каталоге **jni**.
7. Каталог **key** содержит ключи ТПП.
8. Каталог **lib** содержит сторонние библиотеки. Например, *bcprov-jdk15on-152.jar*, *cgi-0.6.jar*, *commons-beanutils-1.9.2.jar*, ..., *zxing-core-3.2.0.jar*.
9. Каталог **log** содержит журналы работы ТПО.
10. Каталог **res** — каталог содержит ресурсы ТПО:
- 1) каталог **dev** содержит ресурсы, необходимые для работы оборудования:
 - а) каталог **disp** содержит ресурсы для диспенсеров:
 - каталог **bill** содержит профили для диспенсеров купюр:

- каталог **2** содержит профили для двухбуксовых диспенсеров купюр: `default_usd, ...`;
- каталог **4** содержит профили для четырехбуксовых диспенсеров купюр: `default, ...`;

2) каталог **dmi** содержит информацию об оборудовании, полученную командой `dmidecode`. Например, в файле `barcode` содержится информация о сканере штрих-кодов;

3) каталог **module** содержит ресурсы, необходимые для работы модулей;

а) каталог **i18n** содержит текстовки, используемые для интернационализации форм/сценариев. Подробнее об интернационализации форм и сценариев в документах [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#), [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#) соответственно: `input_en.properties, ..., input_uk.properties`;

б) каталог **input** содержит ресурсы модулей ввода:

- каталог **advanced** содержит сценарии для сервисов, использующих усовершенствованный провайдер;
- каталог **mobile** содержит ресурсы для работы мобильного обработчика. Например, каталог **250** содержит `capacity.csv.gz` — файл емкостей для определения оператора, `defectors.csv.gz` — файл со списком перебежчиков, `form.xml` — форма, которая будет использована при оплате, `params.properties` — файл содержит параметры для работы с файлами емкостей и перебежчиков. Число является тсс кодом (250 — Россия).
- каталог **card** содержит ресурсы карточного модуля;
- каталог **ds** содержит ресурсы для autocomplete-полей. Например, для сервиса с номер «192» в каталоге **192** могут содержаться файлы: `stations.dat, ..., vozi.dat` — список файлов, по этому имени будет производиться обращение к файлу из форм/сценариев;

-
- каталог **exchange** содержит ресурсы модуля обмена валюты:
 - *form.xml* — форма, в которой задается сумма для обмена;
 - каталог **keeper** содержит ресурсы кошелька:
 - каталог **maps** содержит изображения карт городов и файлы свойств, используемые для привязки изображений к геоданным (широта, долгота);
 - каталог **paytpl** содержит формы для модуля шаблонизации;
 - каталог **refs** содержит справочники кошелька;
 - *keeper.properties* — файл с настройками кошелька;
 - каталог **menu** содержит ресурсы меню. Например, *help.html*, *help_mk.html*, *help_uk.html* — локализованные текстовки экрана помощи; *offer.html*, *offer_uk.html* — локализованные текстовки экрана оферты;
 - каталог **universal** содержит формы для сервисов, работающих на универсальном обработчике;
 - каталог **utility** содержит служебные формы: *checkcopy.xml* — форма копии чека, *console.xml* — форма входа в сервисное меню, *paystatus.xml* — форма статуса платежа, *srchprv.xml* — форма поиска сервиса;
- 4) каталог **system** — *key-server.properties*, *mail-server.properties*;
- 5) каталог **templates** содержит шаблоны чеков;
- а) каталог **default** — шаблоны чеков по умолчанию;
 - б) каталог **logo** содержит логотипы, которые могут быть распечатаны на чеке.
- 6) каталог **trademark** содержит логотипы;
- 7) *maintenance.rules* — файл содержит список правил на удаление файлов из текущей сборки. Подробнее в разделе [6.6](#);
- 8) *params.properties* — файл с настройками ТПО. [«Терминальное ПО 7 версии. Программное обеспечение «Процессинговый центр Pay-logic». Руководство](#)
-

[пользователя»;](#)

11. Каталог **update** содержит обновления, доступные для ТПО.
12. Файл *adplay.sh* представляет собой скрипт запуска плеера рекламы.
13. Файл *configure.sh* представляет собой скрипт запуска конфигулятора.
14. Файл *conf.xml* — конфигурационный файл терминала. Содержит настройки соединения, оборудования и т.д. Генерируется конфигуратором.
15. Файл *javapath* содержит путь до используемой JVM.
16. Файл *java.properties* содержит параметры java-машины.
17. Файл *run.sh* представляет собой скрипт запуска терминала.

Каких-либо ограничений на количество каталогов, файлов или их наименования и/или типы нет в системе обновлений. Иными словами, в систему обновления можно помещать любые файлы и система обновления будет закачивать эти файлы в домашний каталог ТПО.

Служебные файлы системы обновления удалять или модифицировать вручную нельзя. Это приведет к неработоспособности системы обновления.

5 РАБОТА С СИСТЕМОЙ ОБНОВЛЕНИЙ

5.1 НАСТРОЙКА СИСТЕМЫ ОБНОВЛЕНИЯ В КАБИНЕТЕ

Параметры системы обновлений терминальные точки получают от сервера в виде справочников. Информацию о том, где располагается система обновлений, терминал получит только после того, как установит связь с сервером процессинга.

Адрес системы обновлений определяется в разделе «Справочники — Система — Профили адресов сервера» кабинета агента.

Затем возможно выбрать созданный профиль адресов сервера в разделе:

1. «Справочники — Системные параметры» на вкладке «Системные параметры».
2. «Агент — Агенты» на вкладке «Основное» в свойствах агента.
3. «Точки — Параметры точки», а затем назначить группу параметров для определенной точки.

Выбор профиля адресов сервера для каждой точки осуществляется по следующему алгоритму: если профиль адресов сервера задан в свойствах агента, то будет использоваться он, другие игнорируются. При этом, если в настройках вышестоящего агента профиль тоже указан и выставлен флаг **«Распространять на субагентов»**, то используется адрес сервера вышестоящего агента. Если в свойствах агента профиль адресов сервера не задан, то анализируется, настроен ли профиль в параметрах точек. Если в параметрах точек профиль также не задан, тогда используется адрес сервера, который указан в системных параметрах («Справочники — Системные параметры»).

Различные системы обновлений могут находиться как на одном сервере, но иметь разные корневые каталоги, так и физически находиться на разных серверах.

Управление обновлением терминальных точек выполняется в разделе «Мониторинг - Состояние оборудования». В этом разделе выставляются или снимаются флажки обновления ПО для каждой точки (индивидуально или массово). Если флажки обновлений сняты, соответствующие точки не скачивают обновление и не

обновляются. Также в разделе мониторинга отображается текущее состояние модулей ПО точки: версия и номер билда — сборки обновления.

Обновление точек до последней актуальной версии также может выполняться автоматизированно по расписанию. Данный функционал позволяет обновить парк терминалов, единовременно обновляя лишь некоторое количество терминальных точек, что исключает перегрузку сервера обновлениями запросами.

Параметры для автоматизированного обновления также задаются в кабинете в разделе «Справочники — Системные параметры — Системные параметры» с помощью параметра **Максимальное количество точек для обновления по заданиям** — определяет сколько точек могут одновременно получать обновления при задействовании функционала обновления по расписанию.

Функционал обновления по расписанию и его возможности подробно рассмотрены в разделе 6.3 настоящего руководства.

5.2 АЛГОРИТМ РАБОТЫ СИСТЕМЫ ОБНОВЛЕНИЙ

Для обеспечения корректной работы автоматического обновления применяются служебные файлы. Эти служебные файлы содержат информацию о том, какие файлы находятся на сервере, версии файлов, а также версии (билд) всех файлов в целом.

Сборка — совокупность всех файлов (за исключением служебных), находящихся в системе обновлений и подлежащих скачиванию терминалами, объединенная общим перечнем (списком) файлов.

Билд (build) — порядковый номер сборки системы обновлений. Монотонно увеличивающееся число от 0 до 99. При каждой подготовке сборки обновления билд увеличивается на 1, по достижении 99 его значение снова сбрасывается в 0.

Процедура выкладывания обновления заключается в размещении в каталоге обновления новых и измененных файлов и запуске служебного ПО — генератора билда, подготавливающего список файлов в сборке и изменяющего (поднимающего) билд текущей сборки.

Процедура обновления файлов ТПО происходит по следующему алгоритму:

1. Определяются флаги обновления программ из справочников. Если флаги обновлений выставлены, начинается процесс обновления ПО, в противном случае обновление не происходит.
2. Определяются адреса системы обновлений для точки, значения берутся из профиля адресов в параметрах точки
3. Сравнивается текущий номер сборки системы обновлений и сборки ТПО. Если ТПО ранее не скачивало обновления, то номер сборки терминальной точки не определен. Если текущий номер сборки не определен или не равен номеру сборки системы обновлений — терминальная точка считается требующей обновления.
4. Осуществляется получение списка файлов в сборке системы обновлений (скачивание файла `list.xml`).
5. Определяется перечень файлов, которые требуется скачать. Для построения такого перечня строится список файлов, находящихся в каталоге `/update`

домашнего каталога ТПО и сравнивается с полученным списком файлов. Сравнение выполняется по дате последней модификации, размеру и контрольной сумме файла. Если локальный файл отличается по этим параметрам от файла из системы обновлений, то принимается решение о необходимости скачивания файла. Таким образом с сервера скачиваются только новые и измененные файлы.

6. Осуществляется скачивание необходимых файлов. Закачка файлов производится последовательно. По завершении закачки его размер и контрольная сумма сравниваются со значениями в файле *list.xml*.

7. По завершении скачивания всех файлов, в случае, если ошибок при скачивании не выявлено — ТПО выставляет специальные флаги обновления для последующей перезагрузки и замены рабочих файлов в домашнем каталоге скачанными. При наличии ошибок скачивания флаги успешности обновления не выставляются и терминал по истечении определенного таймаута (5 минут) будет пытаться скачать обновление заново.

8. Осуществляется перезагрузка и обновление рабочих файлов ТПО. Выставленные флаги обновления отслеживаются ТПО, и при их наличии ТПО по завершении платежа (выход в главное меню) инициирует процедуру обновления рабочих файлов. Поскольку рабочие файлы при работе ПО могут быть открыты, выполняются следующие шаги:

- 1) завершение работы ТПО;
- 2) замена файлов. На этом шаге служебный сценарий, обеспечивающий повторный запуск ПО после его завершения, выполняет копирование всех файлов из каталога */update* в домашний каталог ТПО с заменой существующих файлов. Копируется содержимое каталога */update* полностью;
- 3) запуск ТПО, сброс флагов обновления и генерация событий об успешном обновлении на сервер.

Изложенный алгоритм распространяется на программные файлы и на файлы ресурсов, то есть только на те файлы, которые находятся внутри каталога обновления ТПО. Для скачивания рекламных роликов и рекламных баннеров применяется иной механизм. Различие обусловлено тем, что скачивание рекламных роликов и баннеров менее критичный процесс и не требует перезапуска ТПО.

Описание служебных файлов, находящихся в системе обновлений:

1. **list.xml** — перечень файлов, входящих в текущую сборку. Представляет собой XML документ, содержащий перечень файлов для скачивания. Для каждого файла указывается путь до него (относительно каталога обновления), его размер (в байтах), его контрольная сумма CRC32, а также время последней модификации.

2. **list.crc** — файл, содержащий значение контрольной суммы, размера (в байтах) и даты изменения самого файла *list.xml*.

3. **scheme** — служебный каталог с файлами необходимыми для работы FileListGenerator'a:

1) **libs** — каталог содержит библиотеки FileListGenerator'a:

a) *commons-logging-1.2.jar*, *java-cup-11b-runtime.jar*,
org.eclipse.wst.xml.xpath2.processor_1.1.0.jar, *resolver.jar*,
xercesImpl.jar, *xml-apis.jar* — библиотеки для работы xsd1.1 валидации, при их отсутствии проверка производится не будет;

b) *schemas.jar* — библиотека с xsd-валидатором и схемами для него;

c) **spelling** — библиотеки для работы валидатора орфографии: *anguagetool-2.0.1.en.ru.ua.jar* (урезанная версия библиотеки *languagetool-2.0.1* (~45Мб) из которой удалены нативные библиотеки для windows и оставлены словари для русского, украинского и английского языков), *morfologik-fsa-1.10.0.jar*, *morfologik-speller-1.10.0.jar*, *morfologik-stemming-1.10.0.jar*, *segment-1.4.2.jar*;

2) *options.properties* — файл содержит настройки валидации.

4. **start.sh (update.sh)** — скрипт запуска утилиты.

5. **FileListGenerator.jar** — генератор обновления. Исполняемый файл, обеспечивающий генерацию сборки.

6. **update.build** — файл с номером билда, хранит числовое значение текущего билда.

При построении сборки в перечень файлов включаются все файлы, находящиеся в каталоге обновления и его подкаталогах, за исключением файлов *list.xml*, *list.crc*, *FileListGenerator.jar*. Файл *update.build* включается в обновление.

FileListGenerator для 7 версии ТПО позволяет осуществлять проверку форм и сценариев на соответствие xsd-схеме, а так же проверку орфографии. Синтаксис форм и сценариев ТПО 7 версии отличается от ТПО 5 версии. Подробно синтаксис и отличия в версиях описаны в документах [«Формы оплаты для универсального модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство администратора»](#), [«Сценарии оплаты для усовершенствованного модуля ввода данных. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#). Параметры запуска FileListGenerator определяются настройками файла *options.properties*:

1. ***form.check=[*enable|validate-only|disable]*** — определяет проводить ли xsd-валидацию для форм:
 - 1) ***enable*** — проверка включена, при отрицательном результате проверки обновление сгенерировано НЕ будет. Значение по умолчанию;
 - 2) ***validate-only*** — проверка включена, при отрицательном результате проверки обновление будет сгенерировано;
 - 1) ***disable*** — проверка выключена.
2. ***form.namespace=[*optional|required]*** — выполнять подстановку пространства имен для форм, если xsd-валидация включена:
 - 1) ***optional*** — подстановка пространства имен в документ производится не будет;
 - 2) ***required*** — обязательная подстановка пространства в каждый документ.
3. ***form.spelling=[enable|*validate-only|disable]*** — выполнять проверку орфографии для форм, ***validate-only*** — значение по умолчанию.
4. ***scenario.check=[*enable|validate-only|disable]*** — проводить ли xsd-валидацию для сценариев, ***enable*** — значение по умолчанию.
5. ***scenario.namespace=[*optional|required]*** — выполнять подстановку пространства имен для сценариев, если xsd-валидация включена, ***optional*** —

значение по умолчанию.

6. **`scenario.spelling=[enable|*validate-only|disable]`** — выполнять проверку орфографии для сценариев, **`validate-only`** — значение по умолчанию.

7. **`paths.check.exclude=gui/fonts,res/module/keeper.jre`** — список каталогов, которые будут проигнорированы при xsd-валидации. Каталоги **`gui/fonts,res/module/keeper,jre`** исключены по умолчанию.

8. **`paths.generate.exclude=`** — список каталогов, которые будут игнорироваться при генерации обновления.

9. **`spelling.lang=ru,en,ua`** — список языков для проверки орфографии (пока доступно только 3 языка: ru,en,ua). Первый язык основной, на нем будет выведен список ошибок. По умолчанию список пустой, проверка выполняться не будет.

10. **`maintenance.check=[*enable|validate-only|disable]`** — параметр доступен с версии FileListGenerator 1.3. Определяет, выполнять ли поиск коллизий между файлами обновления и содержимым `maintenance.rules`. Значение по умолчанию **`enable`**.

11. **`maintenance.path=path/maintenance.rules`** — параметр доступен с версии FileListGenerator 1.3. Путь до файла `maintenance.rules`, по умолчанию файл ищется в корневой папке.

В случае, если в результате проверки были обнаружены ошибки, то генерируется уведомление об ошибках, адрес доставки указывается в скрипте запуска обновления. В зависимости от значений параметров, обновление может быть сгенерировано или не сгенерировано.

Скрипт запуска `start.sh`:

```
#!/bin/bash
java -cp ./scheme/libs/*:./scheme/libs/spelling/*:FileListGenerator.jar
ru.softlogic.generator.FileListGenerator $1
```

Параметры:

1. **val** — выполнить только валидацию контента. Настройки валидации берутся из `options.properties`.
2. **gen** — выполнить генерацию обновления.
3. При запуске без параметров будет выполнена валидация контента согласно `options.properties`, если валидация прошла успешно, то сгенерируется обновление.

Команды запуска скрипта для каждого из случаев будут выглядеть следующим образом:

```
sh start.sh val
```

```
sh start.sh gen
```

```
sh start.sh
```

Рекламные баннеры (содержимое каталога **banners**) скачивает отдельный поток, не связанный с потоком, скачивающим основное обновление. Поток, скачивающий ролики, также вычисляет при этом контрольную сумму для проверки корректности скачивания ролика.

Рекламные ролики (содержимое каталога **ad**) также скачиваются отдельным потоком. При этом проверка корректности скачивания выполняется при помощи сверки с контрольной суммой, которая указывается для каждого ролика в кабинете. Ошибки при скачивании роликов не влияют на работоспособность ТПО.

Журнал (лог) обновления системы находится в файле `/log/update.log` домашнего каталога ТПО. В этот журнал пишется вся информация о ходе обновления согласно вышеизложенному алгоритму, в журнал также фиксируются все ошибки, возникшие при скачивании.

5.3 ПОДГОТОВКА ОБНОВЛЕНИЯ НА СЕРВЕРЕ

Для того, чтобы терминальное ПО могло начать скачивание обновлений, необходимо подготовить обновление на сервере. Подготовка заключается в выкладывании обновленных файлов в каталог обновления (подготовка файлов) и построении служебных файлов (подготовка сборки).



Внимание!

В системе обновлений должна быть размещена только одна версия библиотек ТПО. То есть, недопустимо размещать в системе обновлений для одной версии ТПО, например, библиотеку *vlcj-2.3.1.jar* и *vlcj-2.4.1.jar*.

Подготовка файлов выполняется любыми доступными средствами ОС на сервере обновлений (ftp, ssh, и т.д.). Как правило, используются скрипты для подготовки файлов и запуска утилиты *FileListGenerator.jar* (пример скрипта приведен в приложении А). Запуск осуществляется командой:

```
java -jar FileListGenerator.jar
```

В процессе работы *FileListGenerator.jar* сканирует каталог обновления со всеми обновлениями, вычисляет для каждого файла величину контрольной суммы и строит список файлов — *list.xml*. По завершении сканирования каталогов и построения файла *list.xml* утилита рассчитывает размер и контрольную сумму самого файла *list.xml* и увеличивает номер сборки в файле *update.build*.

5.4 НАСТРОЙКА АВТОМАТИЧЕСКОЙ ЗАГРУЗКИ ФОРМ И СЦЕНАРИЕВ В КАБИНЕТ

При развертывании системы обновлений создаются скрипты, реализующие различные задачи, в том числе настройку автоматической загрузки форм и сценариев.

Как правило, загрузка файлов форм и сценариев осуществляется первоначально на FTP-сервер. После чего осуществляется их загрузка в систему обновлений и в кабинет. Подготовка сборки осуществляется с помощью утилиты *FileListGenerator.jar* (раздел 5.3). Перед запуском *FileListGenerator.jar* работу осуществляют скрипты, которые сканируют, есть ли текущие подключения к серверу — в таком случае осуществляется завершение работы скрипта, пуст ли каталог — в таком случае осуществляется завершение работы скрипта, были ли выложены новые файлы в течение последний 5 минут — в таком случае так же осуществляется завершение работы скрипта. Иначе осуществляется подготовка файлов, копирование, подготовка профиля обновлений и запуск утилиты *FileListGenerator.jar*.

В скрипте так же указываются адрес директории с формами, сценариями в кабинете. Например:

```
GFFORMSDIR="/srv/glassfish3/forms/"
```

Затем осуществляется копирование файлов в систему обновлений и в кабинет:

```
for file in ` $FIND $DATADIR -type f `;do
  log "Перемещаем $file"
  file2=`echo $file | sed "s\/\var\/www\/ftpsupport\/v5\/\/\/"`
  FILESIZE=`du -k $file | cut -f1`
  echo "$file2 ($FILESIZE kB)" >> /tmp/ftpfilelistv5
  if echo "$file2" | grep -i "/forms/" >/dev/null; then
    log "Дополнительно копируем $file2"
    cp -f $file2 $GFFORMSDIR
  fi
  fullpath="$HTTPODIR/linux/$file2"
  filepath="{fullpath%/*}"
  mkdir -p $filepath > /dev/null 2>&1
  cp -f $file2 $fullpath
  fullpath="$HTTPODIR/windows/$file2"
  filepath="{fullpath%/*}"
  mkdir -p $filepath > /dev/null 2>&1
  mv -f $file2 $fullpath
done
```

Копирование в кабинет осуществляется в следующей части скрипта:

```
if echo "$file2" | grep -i "/forms/" >/dev/null; then
  log "Дополнительно копируем $file2"
  cp -f $file2 $GFFORMSDIR
fi
```

Пример скрипта целиком приведен в приложении А.

6 РЕКОМЕНДАЦИИ ПО ИСПОЛЬЗОВАНИЮ СИСТЕМЫ ОБНОВЛЕНИЙ

6.1 ОБЩИЕ ПОЛОЖЕНИЯ

Система обновления (СО) является универсальным и надежным механизмом для дистанционного обновления парка терминалов. Прямое назначение системы — обновление ПО, файлов форм, сценариев, текстовок локализации, файлов емкостей и иных файлов ТПО. Система также может использоваться для доставки любых файлов на терминал, даже если эти файлы напрямую не относятся к ТПО.

В совокупности с возможностью выполнения произвольных команд система обновлений позволяет при необходимости доставить на терминал и удаленно запустить скрипты и/или сторонние программы, не входящие в ТПО, но требующиеся по каким-либо причинам на терминале. Возможности действий, которые можно выполнить с помощью этого механизма, ограничиваются только привилегиями пользователя под учетной записью которого функционирует ТПО.

В версии 7.129 на ТПО реализован механизм работы с архивами, аналогичный механизму для ТПО 5 версии. Архив (формата tar.gz и bz2) скачивается из СО и разархивируется. Во избежание цикличного обновления сами архивы не удаляются и копируются в папку с ТПО вместе с папкой, получившейся в итоге разархивации.

6.2 РЕКОМЕНДУЕМЫЙ ПОРЯДОК ОБНОВЛЕНИЯ ТОЧЕК

Обновление терминальных точек — стандартный этап рабочего процесса при эксплуатации ТПО. Тем не менее для обеспечения максимально надежной работы системы обновлений следует придерживаться определенного порядка выполнения обновления терминальных точек:

1. Снять флаги обновления со всех точек и убедиться, что точки получили новые версии справочников с отключенным флагом обновления (в разделе «Мониторинг — Состояние оборудования»). Это действие крайне желательно делать всегда до того, как выполняются любые действия в системе обновлений.
2. Выложить новые/измененные файлы в систему обновления.
3. Сгенерировать новый билд (раздел 5.3) и убедиться, что в ходе генерации не возникло ошибок. Утилита FileListGenerator проверяет валидность xml-файлов.
4. Установить флаг обновления на небольшую группу точек (одна — три точки), дождаться получения обновления точками и убедиться, что они, получив обновления, работают корректно.
5. Установить флаги обновления для оставшейся части парка терминальных точек.

6.3 АВТОМАТИЗАЦИЯ ОБНОВЛЕНИЯ ТОЧЕК

Настройка системы обновлений на стороне сервера позволяет автоматизировать процесс обновления сколь угодно большого парка терминалов. Функционал обновления по расписанию позволяет достичь нескольких целей:

1. Обновить последовательно все точки, применяя обновление небольшими группами.
2. Исключить перегрузку сервера обновлений при большом парке терминалов.
3. Синхронизировать изменение настроек меню и изменение форм.

Для осуществления обновления точек по расписанию предназначен раздел «Мониторинг — Обновление терминалов». Раздел позволяет просматривать уже созданные и выполненные задания на обновление так и добавить новое задание на обновление терминалов. Подробно добавление задания на обновление описано в документе [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

Функционал заданий на обновление работает следующим образом: выбираются терминальные точки, требующие обновления — версия ПО и билд ниже, чем заданные в поле «Версия». Из общего числа точек, требующих обновления, выбираются первые N, где N — это максимальное количество точек для обновления по заданиям из настроек системных параметров. Для этих точек устанавливаются флаги обновления программ и ресурсов, для всех остальных точек эти флаги снимаются. Таким образом, одновременно к серверу обновлений будут обращаться не более N терминалов.

По мере завершения обновления этих терминалов (когда версия ПО и билд по данным мониторинга станет равным указанному в задании) с обновленных терминалов флаги обновления будут сняты, а для следующих нескольких терминалов из списка требующих обновления, но которые ещё не обновлены — флаги будут выставлены. Таким образом, циклично переставляя флаги обновления с одних терминалов на другие, сервер автоматически будет постепенно обновлять весь парк (или его часть, если ограничен город).

Использование триггера для обновления версий меню	<input checked="" type="checkbox"/>
ID сервиса карт сдачи	210
ID сервиса сохранения средств	549
Максимальное количество точек для обновления по заданиям	60

Рисунок 6.3.1 — Параметры обновления «Справочники — Системные параметры»

Флаг «Увеличить версию справочников» в параметрах задания обновления используется в том случае, когда нужно одновременно с обновлением принудительно обновить и справочники (в том числе, меню). Как правило, должен применяться одновременно со снятием флага «Использование триггера для обновления версий меню».

Флаг «Использование триггера для обновления версий меню» (рисунок 6.3.1) отвечает за поведение системы при внесении изменений, которые должны быть переданы на терминалы. Если этот флаг установлен, то при внесении изменений в справочники, профили меню, соответствующие версии справочников, изменяются (увеличиваются) и терминальные точки благодаря этому обнаруживают изменения и запрашивают новые справочники с сервера.

Если флаг снят, то при изменении меню и справочников изменение (увеличение) версии справочников не происходит и терминалы не получают обновленные справочники и меню до тех пор, пока версии справочников не будут подняты вручную.

Типичным примером когда данное действие необходимо, является массовая перенастройка действующих меню: добавление в него новых сервисов. Логичным является отключить обновление справочников на терминалах до того времени, пока не будут внесены все изменения или группа изменений, тем более, что одновременно с изменением меню, как правило, меняются и формы. В этом случае, для синхронного обновления меню и форм выполняются следующие действия:

1. Снимается отметка «Использование триггера для обновления версий меню».
2. Снимаются флаги обновления со всех точек.

3. Вносятся необходимые изменения в настройки профиля меню, свойства сервисов.
4. Подготавливаются измененные/новые формы.
5. Новые и измененные формы выкладываются в систему обновления, генерируется очередной билд.
6. Создается задание на обновление по расписанию, с установленной отметкой «Увеличить версию справочников».

Обработчик заданий автоматического обновления в этом случае, выставив флаг обновления очередной точке одновременно с этим поднимает и версию справочников для неё, что и к скачиванию обновления, и к получению последних актуальных справочников и меню.

6.4 ПЕРЕДАЧА ПРОИЗВОЛЬНЫХ ФАЙЛОВ НА ТЕРМИНАЛ

Как было отмечено выше, система обновления позволяет передавать на терминал любые файлы. Поэтому при необходимости в целях обслуживания парка передать на терминалы некоторую утилиту, архив или скрипт операционной системы, это можно сделать через систему обновления. Для этого желательно в каталоге обновления создать отдельный каталог, в который и поместить требуемые к передаче файлы.

Следует помнить о том, что ТПО скачивает файлы в каталог **update** домашнего каталога ТПО, а после завершения скачивания перезапуска копирует содержимое каталога **update** в домашний каталог ТПО. Поэтому путь, по которому будут располагаться переданные на терминал файлы, будет по окончании обновления таким же, каким он был в системе обновления, но относительно домашнего каталога ТПО. Это существенно при необходимости в дальнейшем выполнить сценарий ОС: при написании сценариев нужно использовать относительные пути либо определять абсолютные пути до текущего каталога сценария в самой сценарии доступными средствами ОС терминала.

Для того, чтобы использовать эту возможность в целях запуска скриптов ОС и/или стороннего ПО на терминале, необходимо:

1. Подготовить нужные файлы и/или скрипты в систему обновления.
2. Выполнить процесс обновления ТПО, учитывая рекомендации раздела 6.2.
3. Убедиться что терминальные точки скачали обновление с требуемыми файлами и обновились. После этого файлы и скрипты оказываются в домашнем каталоге ТПО по тому же относительному пути, по которому они находились в системе обновления.
4. Запустить требуемые исполняемые программы и/или скрипты при помощи функционала отправки произвольной команды терминалам.

Отправка команд терминалам подробно рассмотрена в руководстве [«Кабинет агента. Программное обеспечение «Процессинговый центр Pay-logic». Руководство](#)

[пользователя»](#), здесь рассматриваются особенности выполнения произвольных команд.

Произвольная команда — это команда ТПО выполнить команду ОС. Произвольная команда может содержать как выполнение команды ОС или утилиты ОС, так и выполнение команды по запуску скрипта или приложения. Для ОС семейства Linux для написания скриптов применяется язык shell (файлы ***.sh**), для ОС Windows – применяются либо bat-файлы (файлы ***.bat**), либо JavaScript-сценарии (файлы ***.js**). JavaScript предоставляет несравнимо большую гибкость и возможности чем язык bat-файлов, поэтому является предпочтительным.

При выполнении произвольных команд для запуска произвольной программы, текст команды имеет вид:

```
./path_to_file/program_file
```

Например, если необходимо запустить скрипт *maintenance.sh*, который требуется поместить в домашний каталог в подкаталог **custom_script**, следует:

1. Поместить скрипт в систему обновления по такому же относительному пути, т.е в системе обновления скрипт будет располагаться в каталоге:

```
/update/v5/linux/custom_script/maintenance.sh
```

2. Обновить терминалы, по завершении обновления скрипт будет находиться в домашнем каталоге ТПО с относительным путем:

```
custom_script/maintenance.sh
```

3. Отправить терминалу произвольную команду:

```
./custom_script/maintenance.sh
```

Команды на выполнение сценариев shell, bat — выполняются непосредственно. Команды на выполнение сценариев JavaScript (***.js**) выполняются при помощи утилиты Windows — wscript:

```
wscript.exe ./custom_script/maintenance.js
```

 **Внимание!** Возможности, которые можно таким образом задействовать, ограничиваются только привилегиями пользователя, под учетной записью которого функционирует ТПО.

6.5 СКАЧИВАНИЕ ФАЙЛОВ БЕЗ ПЕРЕЗАГРУЗКИ

Обновление ТПО приводит к последующему его перезапуску. В определенных случаях требуется доставить файл на терминал, но перезапуска ТПО необходимо избежать. Данное действие возможно сделать, сформировав обновление по следующему алгоритму:

1. Подготовить файлы обновления, которые необходимо передать на терминал.
2. Снять флаги обновления с точек.
3. Поместить требуемые файлы на сервер обновления в каталог обновления.
4. Добавить в каталог обновления (например, `/update/v5/linux/`) файл с именем **z** и без расширения (либо с расширением `*.txt`) и содержащий любой текст.
5. Построить новый билд системы обновления.
6. Незамедлительно удалить файл **z** («`z.txt`»).

Данная последовательность действий приведет к тому, что в перечне файлов (`list.xml`) последним в списке окажется файл **z**, который по причине его удаления с сервера не сможет скачать ни один терминал. Результатом станет то, что терминалы, скачав обновления в каталог `./update` не выставят флаг успешного обновления и не будут перезапускаться и заменять исполняемые и иные файлы в домашнем каталоге на скачанные. Тем не менее, поскольку нужные файлы будут физически скачаны и будут находиться по тем же относительным путям, но относительно каталога `/update`, а не домашнего каталога ТПО — к данным файлам можно будет обратиться при помощи произвольной команды терминалам, нужно лишь учесть относительный путь – добавить в этот путь каталог `update`.

Примеры:

```
./update/custom_script/maintenance.sh
```

```
wscript.exe ./update/custom_script/maintenance.js
```

6.6 УДАЛЕНИЕ ФАЙЛОВ С ТЕРМИНАЛА

Для удаления файлов с терминала применяются правила прописанные в файле *maintenance.rules*, который располагается в системе обновлений в каталогах ОС (windows/linux) и содержит в себе пути до файла(-ов), которые необходимо удалить, относительно папки ТПО. Пример содержимого файла:

```
rmi resources/forms/943.xml
rmi resources/forms/944.xml
rmi ext/gui-bluesphere2.jar
rmi lib/commons-codec-1.3.jar
rmi lib/commons-compress.jar
rmi lib/commons-io.jar
rmi lib/httpclient-4.1.2.jar
rmi lib/httpcore-4.1.2.jar
rmi lib/log4j-1.2.15.jar
rmi lib/mail.jar
rmi lib/velocity-dep.jar
rmi forms.xml
```

Команды:

1. `rmi` — удалить файл из папки ПО и из папки обновлений ПО.
2. `rm` — удалить файл из папки ПО.



Внимание! Нельзя указывать правила на удаление файлов, которые присутствуют в системе обновления. В таком случае можно получить бесконечную перезагрузку ТПО, так как файл будет скачиваться и тут же удаляться при перезагрузке ТПО. В ОС семейства Linux эти файлы необходимо удалить из системы обновлений, в ОС семейства Windows — загрузить файл размером 0 кб с таким же именем.

Доступна возможность работы с масками и регулярными выражениями.

Работа с масками поддерживается с помощью конструкции `wildcard`. Для замены любой строки символов используется звёздочка (*), для замены любого одиночного символа — знак вопроса (?).

Пример:

```
rmu wildcard log/*.log
```

В примере из каталога `log/` ТПО и из папки обновлений будут удалены все файлы с расширением `.log`.

Работа с регулярными выражениями поддерживается с помощью конструкции `regex`. Для указания правил используется язык регулярных выражений.

Пример:

```
rmu regex log/(.*d.*)
```

В примере из каталога `log/` ТПО и из папки обновлений будут удалены все файлы, в названии которых есть хотя бы один цифровой символ `.`

7 ОСОБЕННОСТИ ОБНОВЛЕНИЯ ТОЧЕК ДРУГИХ ТИПОВ

7.1 ОСОБЕННОСТИ ОБНОВЛЕНИЯ ТОЧЕК ТИПА «РМА»

Для обновления РМА через систему обновлений используется каталог **<корневой каталог СО>/rma6**.

Процедура обновления файлов приложения РМА происходит по следующему алгоритму:

1. Определяется флаг обновления приложения из раздела «Мониторинг — Мониторинг РМА и Моб.». Если флаг обновлений установлен, то начинается процесс обновления ПО, в противном случае обновление не происходит.
2. Определяются адреса системы обновлений для точки, значения берутся из профиля адресов сервера в параметрах точки.
3. Сравнивается текущий номер сборки системы обновлений и сборки приложения РМА. Если приложение РМА ранее не скачивало обновления, то номер сборки точки не определён. Если текущий номер сборки не определён или не равен номеру сборки системы обновлений — точка считается требующей обновления.
4. Осуществляется получение списка файлов в сборке системы обновлений (скачивание файла *list.xml*).
5. Определяется перечень файлов, которые требуется скачать. Для построения такого перечня строится список файлов, находящихся в каталоге */update* домашнего каталога приложения «РМА» и сравнивается с полученным списком файлов. Сравнение выполняется по дате последней модификации, размеру и контрольной сумме файла. Если локальный файл отличается по этим параметрам от файла из системы обновлений, то принимается решение о необходимости скачивания файла. Таким образом, с сервера скачиваются только новые и измененные файлы.
6. Осуществляется скачивание необходимых файлов. Закачка файлов производится последовательно. При завершении закачки его размер и контрольная сумма сравниваются со значениями в файле *list.xml*.

7. После скачивания файлов обновления возможны два варианта:

1) *Обновление с согласия кассира* — при завершении скачивания обновлений кассир получает уведомление о том, что обновление ожидает установки. Обновление применяется при перезапуске РМА, но кассир может продолжить работу без применения обновления на кассе.

2) *Принудительное обновление* — обновление считается принудительным, если в системе обновлений присутствует файл *force.update*, в котором содержится один символ Y. При получении такого обновления касса при завершении скачивания обновлений блокирует работу, кассир получает уведомление о том, что обновление ожидает установки и необходим перезапуск. В случае принудительного обновления, если кассир принимает платеж клиента, то он может завершить текущий платеж и завершить расчет, но не может создать новый платеж. После обновления файл *force.update* требуется удалить из системы обновлений или изменить его содержимое.

Обновление происходит при перезапуске приложения.

В главном меню приложения в разделе «Обновление» доступны функции:

1. **Проверить наличия обновлений** — доступна всегда, при нажатии проверяется наличие обновления и при наличии инициируется скачивание.
2. **Обновить** — выполняется завершение сеанса кассира и закрытие РМА. Кнопка доступна только при наличии скачанного и готового к применению обновления.

Структура каталогов приложения РМА приведена в документе [«РМА приема платежей с помощью ПК. Программное обеспечение «Процессинговый центр Pay-logic». Руководство пользователя»](#).

ПРИЛОЖЕНИЕ А. СКРИПТ ПОДГОТОВКИ ФАЙЛОВ ДЛЯ ОБНОВЛЕНИЯ

```
#!/bin/bash
# AQ
# 20150121

sleep 2
. /root/bin/ftpupdate_params

DATADIR="/var/www/ftpsupport/v5"
HTTPDIR="/var/www/html/update/v5"
FIND=`which find`
LOG="/var/log/ftpupdate.log"
GENERATOR="$HTTPDIR/windows/FileListGenerator.jar"
BACKUPDIR="/srv/backup/update"
GFFORMSDIR="/srv/glassfish3/forms/"

function log() {
    dst=`date +"%Y.%m.%d %T"`
    echo "$dst support_v5 $@" >> $LOG
}

NETSTATE=`netstat -antp | grep -e "$SERVER_IP:21 " | grep ESTABLISHED`
if [ "x$NETSTATE" != "x" ];then
    log "Установлено сетевое соединение. Выход."
    exit 0
fi

FILELIST=`$FIND $DATADIR -type f`
if [ "x$FILELIST" == "x" ];then
    log "Каталог пуст. Выход."
    exit 1
fi

NEWFILELIST=`$FIND $DATADIR -type f -mmin -5`
if [ "x$NEWFILELIST" != "x" ];then
    log "Есть файлы созданные менее 5 минут назад. Выход."
    exit 0
fi
```

```
cd $DATADIR
SIZE=`find ./ -type f -print0 | du -sck --files0-from=- | tail -n 1 | cut
-fl`

rm -rf $DATADIR/ad > /dev/null 2>&1
rm -rf $DATADIR/banners > /dev/null 2>&1
rm -rf $DATADIR/bin > /dev/null 2>&1
rm -rf $DATADIR/data > /dev/null 2>&1
rm -rf $DATADIR/ext > /dev/null 2>&1
rm -rf $DATADIR/fonts > /dev/null 2>&1
rm -rf $DATADIR/jni > /dev/null 2>&1
rm -rf $DATADIR/key > /dev/null 2>&1
rm -rf $DATADIR/lib > /dev/null 2>&1
rm -rf $DATADIR/log > /dev/null 2>&1
rm -rf $DATADIR/sequence > /dev/null 2>&1
rm -rf $DATADIR/update > /dev/null 2>&1

log "Валидация"
java -jar $GENERATOR > /dev/null 2>&1
BUILDCODE=$?
rm -rf $DATADIR/list.xml
rm -rf $DATADIR/list.crc
rm -rf $DATADIR/update.build
if [ $BUILDCODE -ne 0 ]; then
    log "Ошибка при валидации: $BUILDCODE"
    BUILDERR=`java -jar $GENERATOR 2>&1 | grep Error`
    rm -rf $DATADIR/list.xml
    rm -rf $DATADIR/list.crc
    rm -rf $DATADIR/update.build
    log "$BUILDERR"
    # Отправка события.
    /root/bin/ftpupdate_sendmail.pl "$SERVER: Ошибка обновления v5"
    "$BUILDCODE: $BUILDERR"
    exit 1
fi

# сохранение текущего состояния
if [ ! -d $BACKUPDIR ] ;then
    mkdir $BACKUPDIR
fi
tar -cjf $BACKUPDIR/update_v5-`date +%Y%m%d-%H%M`.tar.bz2 $HTTPTDIR >
/dev/null 2>&1
```

```
OLD_IFS=$IFS
IFS=$'\n'

for file in ` $FIND $DATADIR -type f `;do
    log "Перемещаем $file"
    file2=`echo $file | sed "s/\var/www/ftpsupport/v5///"`
    FILESIZE=`du -k $file | cut -f1`
    echo "$file2 ($FILESIZE kB)" >> /tmp/ftpfilelistv5

    if echo "$file2" | grep -i "/forms/" >/dev/null; then
        log "Дополнительно копируем $file2"
        cp -f $file2 $GFFORMSDIR
    fi

    fullpath="$HTTTPDIR/linux/$file2"
    filepath="{fullpath%/*}"
    mkdir -p $filepath > /dev/null 2>&1
    cp -f $file2 $fullpath
    fullpath="$HTTTPDIR/windows/$file2"
    filepath="{fullpath%/*}"
    mkdir -p $filepath > /dev/null 2>&1
    mv -f $file2 $fullpath
done

IFS=$OLD_IFS

log "Генерируем профиль"
echo >> /tmp/ftpfilelistv5
echo "Total: $SIZE kB" >> /tmp/ftpfilelistv5

if [ -d $HTTTPDIR/linux ]; then
    cd $HTTTPDIR/linux/
    LINGENTEXT=`java -jar FileListGenerator.jar 2>&1`
fi
if [ -d $HTTTPDIR/windows ]; then
    cd $HTTTPDIR/windows/
    WINGENTEXT=`java -jar FileListGenerator.jar 2>&1`
fi

chown -R 0.0 $HTTTPDIR/./
find $HTTTPDIR/./ -perm 600 | xargs chmod 644
```

```
/root/bin/ftpupdate_sendmail.pl "$SERVER: Обновление v5 сгенерировано"  
"Windows: $WINGENTEXT, Linux: $LINGENTEXT" "/tmp/ftpfilelistv5"  
rm -rf /tmp/ftpfilelistv5
```