

Спецификация ярлыка рабочего стола

Version 1.1

Preston Brown <pbrown@kde.org>
Jonathan Blandford <jrb@redhat.com>
Owen Taylor <otaylor@gtk.org>
Vincent Untz <vuntz@gnome.org>
Waldo Bastian <waldo.bastian@intel.com>
Ryan Lortie <desrt@desrt.ca>

1 апреля 2014

Содержание

1	Наименование файлов	2
1.1	ID файлов ярлыков	3
2	Основной формат файла	3
2.1	Комментарии	3
2.2	Заголовки групп	3
2.3	Строки файла	4
3	Возможные типы значений	4
4	Локализованные значения для ключей	5
5	Распознанные ключи ввода на рабочем столе	6
6	Ключ Exec	8
7	Активация D-Bus	10
8	Интерфейсы	11
9	Регистрация типов MIME	12
10	Дополнительные действия приложений	12

10.1 Идентификатор действия	12
10.2 Ключи действий	12
10.3 Замечания по реализации	12
11 Расширение формата	13
12 Приложения	14
12.1 А. Пример рабочего стола	14
12.2 В. В настоящее время зарезервировано для использования в KDE	14
12.3 С. Устаревшие элементы	15
12.4 D. Кодирование Legacy-Mixed (устарело)	15

Аннотация

И в среде рабочего стола KDE, и в GNOME принят аналогичный формат для «записей рабочего стола» или файлов конфигурации, описывающих, как должна запускаться конкретная программа, как она отображается в меню и т.д. Преимуществом единого стандарта является его открытость для более широкого сообщества. Он может быть согласованным всеми сторонами, так что взаимодействие между этими двумя средами и, фактически, любыми дополнительными средами, которые реализуют спецификацию, становится проще.

1 Наименование файлов

Файлы ярлыков рабочего стола должны иметь расширение *.desktop*, за исключением файлов типа Directory, которые должны иметь расширение *.directory*. Определение типа файла на основе расширения делает это определение очень простым и быстрым. Если расширение файла отсутствует, десктоп система должна вернуться к распознаванию с помощью «обнаружения магии».

Для приложений часть имени файла рабочего стола перед расширением *.desktop* должна быть допустимым общеизвестным именем D-Bus. Это означает, что это последовательность непустых элементов, разделенных точками (U + 002E FULL STOP), ни один из которых не начинается с цифры, и каждый из которых содержит только символы из набора [A-Za-z0-9- _]: Буквы ASCII, цифры, тире (U + 002D HYPHEN-MINUS) и подчеркивание (U + 005F LOW_LINE).

Имя ярлыка рабочего стола должно следовать соглашению об «обратном DNS»: оно должно начинаться с обратного доменного имени DNS, контролируемого автором приложения, в нижнем регистре. За доменным именем должно следовать имя приложения, которое обычно пишется слитными словами и начальными заглавными буквами (CamelCase). Например, если владелец *example.org* пишет «Foo Viewer», он может выбрать имя *org.example.FooViewer*, что приведет к созданию файла с именем *org.example.FooViewer.desktop*.

Хорошо известные имена, содержащие тире, разрешены, но не рекомендуются, потому что дефис не разрешен в некоторых связанных случаях использования обращенных имен DNS, таких как пути к объектам D-Bus и имена интерфейсов, а также идентификаторы приложений Flatpak. Если доменное имя автора содержит тире, рекомендуется заменить его подчеркиванием: это не может привести к неоднозначности, так как подчеркивание недопустимо в доменных именах DNS.

Если доменное имя автора содержит метку, начинающуюся с цифры (что недопустимо в известных

именах D-Bus), рекомендуется добавить подчеркивание к этому элементу имени записи рабочего стола. Например, 7-zip.org может выпустить приложение с именем *org._7_zip.Archiver*.

1.1 ID файлов ярлыков

Каждый ярлык рабочего стола, представляющий приложение, идентифицируется его идентификатором файла рабочего стола, который основан на его имени файла.

Чтобы определить идентификатор файла рабочего стола, укажите его полный путь относительно компонента *\$XDG_DATA_DIRS*, в котором установлен файл рабочего стола, удалите префикс «*apps/*» и замените «*/*» на «*-*».

Например, */usr/share/applications/foo/bar.desktop* имеет идентификатор файла рабочего стола *foo-bar.desktop*.

Если несколько файлов имеют одинаковый идентификатор файла рабочего стола, используется первый файл в порядке приоритета *\$XDG_DATA_DIRS*.

Например, если *\$XDG_DATA_DIRS* содержит пути по умолчанию */usr/local/share:/usr/share*, то */usr/local/share/applications/org.foo.bar.desktop* и */usr/share/applications/org.foo.bar.desktop* имеют одинаковый идентификатор файла рабочего стола *org.foo.bar.desktop*, но будет использоваться только первый.

Если существуют и *foo-bar.desktop*, и *foo/bar.desktop*, то не определено, какой из них будет выбран.

Если файл рабочего стола не установлен в подкаталоге приложений одного из компонентов *\$XDG_DATA_DIRS* у него нет идентификатора.

2 Основной формат файла

Входные файлы рабочего стола закодированы в UTF-8. Файл интерпретируется как последовательность строк, разделенных символами перевода строки. Регистр значим везде в файле.

Совместимые реализации **НЕ ДОЛЖНЫ** удалять поля из файла, даже если они их не поддерживают. Такие поля должны храниться где-то в списке, и если файл «переписывается», то они будут включены. Это гарантирует, что любые специфичные для рабочего стола расширения будут сохранены, даже если другая система обращается к файлу и изменяет его.

2.1 Комментарии

Строки, начинающиеся с *#* и пустые строки, считаются комментариями и будут игнорироваться, однако они должны сохраняться при чтении и записи файла записи рабочего стола.

Строки комментариев не интерпретируются и могут содержать любой символ (кроме LF). Однако рекомендуется использовать UTF-8 для строк комментариев, которые содержат символы, отсутствующие в ASCII.

2.2 Заголовки групп

Заголовок группы с именем *groupname* представляет собой строку в формате:

```
[groupname]
```

Имена групп могут содержать все символы ASCII, кроме '[' и ']' и управляющих символов.

Несколько групп не могут иметь одно и то же имя.

Все пары {ключ, значение}, следующие за заголовком группы, пока не появится новый заголовок группы, будут принадлежать текущей группе.

Базовый формат файла ярлыков рабочего стола требует наличия заголовка группы с именем Desktop Entry. В файле могут присутствовать другие группы, но это самая важная группа, которую необходимо явно поддерживать. Эта группа также должна использоваться в качестве «магического ключа» для автоматического определения типа MIME. В файле ярлыков рабочего стола не должно быть ничего, предшествующего этой группе, кроме одного или нескольких комментариев.

2.3 Строки файла

Строки в файле представляют собой пары ключ, значение в формате:

```
Key=Value
```

Пробел до и после знака равенства следует игнорировать; знак = является фактическим разделителем.

В именах ключей могут использоваться только символы A-Za-z0-9-.

Поскольку регистр имеет значение, ключи *Name* и *NAME* не эквивалентны.

Несколько ключей в одной группе не могут иметь одно и то же имя. Ключи в разных группах могут иметь одинаковые имена.

3 Возможные типы значений

Возможные типы значений: *string*, *localestring*, *boolean* и *numeric*.

Значения типа *string* могут содержать все символы ASCII, кроме управляющих символов.

Значения типа *localestring* отображаются пользователем и кодируются в UTF-8.

Значения типа *boolean* должны быть либо строкой *true*, либо *false*.

Значения типа *numeric* должны быть действительными числами с плавающей запятой, которые распознаются спецификатором %f для scanf в локали C.

Экранирующие последовательности \s, \n, \t, \r и \\ поддерживаются для значений типа *string* и *localestring*, для обозначения пробела ASCII, новой строки, табуляции, возврата каретки и обратной косой черты, соответственно.

Некоторые ключи могут иметь несколько значений. В таком случае значение ключа указывается во множественном числе: например, *string(s)*. Несколько значений должны быть разделены точкой с запятой, а значение ключа может быть необязательно завершено точкой с запятой. Завершающие пустые строки всегда должны заканчиваться точкой с запятой. Точки с запятой в этих значениях

Таблица 1: Возможные совпадения различных значений LC_MESSAGES

Значение LC_MESSAGES	Возможные ключи в порядке сопоставления
lang_COUNTRY@MODIFIER	lang_COUNTRY@MODIFIER, lang_COUNTRY, lang@MODIFIER, lang, default value
lang_COUNTRY	lang_COUNTRY, lang, default value
lang@MODIFIER	lang@MODIFIER, lang, default value
lang	lang, default value

должны быть экранированы с помощью \;

4 Локализованные значения для ключей

Ключи с типом *localestring* могут иметь постфикс *[LOCALE]*, где *LOCALE* - тип локали записи. *LOCALE* должен иметь форму *lang_COUNTRY.ENCODING@MODIFIER*, где *_COUNTRY*, *.ENCODING* и *@MODIFIER* могут быть опущены. Если встречается ключ с постфиксом, тот же ключ должен присутствовать и без постфикса.

При чтении в файле ярлыка рабочего стола значение ключа выбирается путем сопоставления текущей локали POSIX для категории *LC_MESSAGES* с постфиксами *LOCALE* всех вхождений ключа с удаленной частью *.ENCODING*.

Сопоставление производится следующим образом. Если *LC_MESSAGES* имеет форму *lang_COUNTRY*, то ключ будет соответствовать ключу в форме *lang_COUNTRY@MODIFIER*. Если такой ключ не существует, то пытаемся сопоставить *lang_COUNTRY* с последующим *lang@MODIFIER*. Затем будет предпринята попытка сопоставления с языком. Наконец, если соответствующий ключ не найден, используется требуемый ключ без указанной локали. Кодировка из значения *LC_MESSAGES* игнорируется при сопоставлении.

Если *LC_MESSAGES* не имеет поля *MODIFIER*, то ни один ключ с модификатором не будет сопоставлен. Точно так же, если *LC_MESSAGES* не имеет поля *COUNTRY*, то ни один ключ с указанной страной не будет сопоставлен. Если у *LC_MESSAGES* просто есть поле *lang*, то оно выполнит прямое совпадение с ключом с аналогичным значением. В следующей таблице 1 перечислены возможные совпадения различных значений LC_MESSAGES в порядке их проверки. Обратите внимание, что поле *ENCODING* не отображается.

Например, если текущим значением категории *LC_MESSAGES* является *sr_YU@Latn* и файл рабочего стола содержит:

```
Name=Foo
Name[sr_YU]=...
Name[sr@Latn]=...
Name[sr]=...
```

тогда используется значение имени с ключом *sr_YU*.

Хотя имена значков типа *iconstring* являются локализуемыми, они не являются удобочитаемыми строками, поэтому обычно не должны обрабатываться инструментами перевода. Не ожидается, что большинство приложений локализуют свои значки; Исключения могут включать значки, содержащие текст или символику культуры.

5 Распознанные ключи ввода на рабочем столе

Ключи либо **НЕОБЯЗАТЕЛЬНЫ**, либо **ТРЕБУЮТСЯ**. Если ключ не является обязательным, он может присутствовать или не присутствовать в файле. Однако, если это не так, реализация стандарта не должна взорваться, она должна обеспечить некоторые нормальные значения по умолчанию.

Некоторые ключи имеют смысл только в контексте, когда другой конкретный ключ также присутствует и имеет определенное значение. Эти ключи не должны использоваться, если конкретный ключ отсутствует или не установлен на определенное значение. Например, ключ *Terminal* можно использовать только в том случае, если значение ключа типа равно «*Application*».

Если **ТРЕБУЕМЫЙ** ключ действителен только в контексте другого ключа, для которого задано определенное значение, он должен присутствовать только в том случае, если для другого ключа задано определенное значение. Например, ключ URL должен присутствовать тогда и только тогда, когда значение ключа типа равно «*Link*».

Некоторые примеры ключей в таблице 2:

Таблица 2: Примеры ключей

Ключ	Описание	Тип значения	REQ?
Type	Эта спецификация определяет 3 типа записей рабочего стола: Приложение (тип 1), Ссылка (тип 2) и Каталог (тип 3). Чтобы разрешить добавление новых типов в будущем, реализации должны игнорировать записи рабочего стола с неизвестным типом.	String	Yes
Version	Версия спецификации ярлыка рабочего стола, которой соответствует этот ярлык. Записи, которые требуют эту версию спецификации, должны использовать 1.1. Обратите внимание, что поле версии не обязательно должно присутствовать.	String	No
Name	Конкретное название приложения, например «Mozilla».	localestring	Yes
GenericName	Общее название приложения, например «Веб-браузер».	localestring	NO
NoDisplay	NoDisplay означает «это приложение существует, но не отображать его в меню». Такое может быть полезно, например, для того, чтобы связать это приложение с типами MIME, чтобы оно запускалось из файлового менеджера (или других приложений), не имея для него пункта меню (для этого есть множество веских причин, например, netscape -remote или kfmclient openURL и другие вещи).	boolean	NO
CHiddenomment	Подсказка для ярлыка, например «Просмотр сайтов в интернете». Значение не должно быть избыточным по отношению к Name и GenericName.	localestring	NO
Icon	Значок для отображения в файловом менеджере, меню и т.д. Если имя является абсолютным путем, будет использован данный файл. Если имя не является абсолютным путем, для определения значка будет использоваться алгоритм, описанный в спецификации темы значков.	iconstring	NO

Hidden	Hidden должно было бы называться Deleted. Это означает, что пользователь удалил (на своем уровне) что-то, что присутствовало (на верхнем уровне, например, в системных каталогах). Это строго эквивалентно тому, что файл .desktop вообще не существует, в отношении этого пользователя. Этот метод также можно использовать для «удаления» существующих файлов (например, из-за переименования) - разрешив make install установить этот файл с Hidden = true в нем.	boolean	NO
OnlyShowIn, NotShowIn	<p>Список строк, идентифицирующих среды рабочего стола, которые должны отображать/не отображать данную запись рабочего стола.</p> <p>По умолчанию файл рабочего стола должен отображаться, если только не присутствует ключ OnlyShowIn, в этом случае по умолчанию файл не отображается.</p> <p>Если установлено значение \$XDG_CURRENT_DESKTOP, то оно содержит список строк, разделенных двоеточиями. По порядку считается каждая строка из этого списка. Если соответствующая запись найдена в OnlyShowIn, файл рабочего стола отображается. Если запись найдена в NotShowIn, то файл рабочего стола не отображается. Если ни одна из строк не совпадает, то выполняется действие по умолчанию (как указано выше).</p> <p>Менеджер входа должен был установить \$XDG_CURRENT_DESKTOP в соответствии со значением DesktopNames, найденным в файле сеанса. Запись в файле сеанса имеет несколько значений, разделенных обычным способом: точкой с запятой.</p> <p>Одно и то же имя рабочего стола не может отображаться и в OnlyShowIn группе, и в NotShowIn.</p>	string(s)	NO
DBusActivatable	Логическое значение, указывающее, поддерживается ли активация D-Bus для этого приложения. Если этот ключ отсутствует, значением по умолчанию является false. Если значение равно true, то реализации должны игнорировать ключ Exec и отправлять сообщение D-Bus для запуска приложения. Смотрите https://specifications.freedesktop.org/desktop-entry-spec/latest/ar01s08.html Активацию D-Bus для получения дополнительной информации о том, как это работает. Приложения должны по-прежнему включать строки Exec= в свои файлы рабочего стола для совместимости с реализациями, которые не понимают ключ DBusActivatable.boolean	NO	
TryExec	Путь к исполняемому файлу на диске, который используется для определения, установлена ли программа на самом деле. Если путь не является абсолютным, файл ищется в переменной среды \$PATH. Если файл отсутствует или не является исполняемым, запись можно игнорировать (например, не использовать в меню).	string	NO
Exec	Программа для выполнения, возможно, с аргументами. См. Ключ Exec для деталей о том, как работает этот ключ. Ключ Exec необходим, если для DBusActivatable не установлено значение true. Даже если DBusActivatable имеет значение true, необходимо указать Exec для совместимости с реализациями, которые не понимают DBusActivatable.	string	NO
Path	Если запись имеет тип Application, рабочий каталог для запуска программы.	string	NO
Terminal	Работает ли программа в окне терминала.	boolean	NO

Actions	Идентификаторы для действий приложения. Их можно использовать, чтобы указать приложению выполнить определенное действие, отличное от поведения по умолчанию. Раздел Действия приложения описывает действия.	string(s)	NO
MimeType	Тип(ы) MIME, поддерживаемые этим приложением.	string(s)	NO
Categories	Категории, в которых запись должна отображаться в меню (возможные значения см. В Спецификации меню рабочего стола).	string(s)	NO
Implements	Список интерфейсов, которые реализует это приложение. По умолчанию ярлык рабочего стола не реализует интерфейсы. Посмотрите Интерфейсы для получения дополнительной информации о том, как это работает.	string(s)	NO
Keywords	Список строк, которые могут использоваться в дополнение к другим метаданным для описания этой записи. Это может быть полезно, например, для облегчения поиска по записям. Значения не предназначены для отображения и не должны быть избыточными по отношению к значениями Name или GenericName.	ocalestring(s)	NO
StartupNotify	Если true, то ИЗВЕСТНО, что приложение отправит сообщение «удалить» при запуске с установленной переменной среды DESKTOP_STARTUP_ID. Если значение равно false, то ИЗВЕСТНО, что приложение вообще не работает с уведомлением о запуске (не показывается ни одно окно, прерывается даже при использовании StartupWMClass и т.д.). Если отсутствует, разумная обработка до реализаций (предполагая false, используя StartupWMClass и т. Д.). (Подробнее см. В спецификации протокола уведомлений при запуске).	boolean	NO
StartupWMClass	Если задано, известно, что приложение отобразит по крайней мере одно окно с данной строкой в качестве своего класса WM или подсказки имени WM (см. Спецификацию протокола уведомления при запуске для более подробной информации).	string	NO
URL	Если ярлык имеет тип ссылки, URL для доступа.	string	YES
PrefersNonDefaultGPU	Если true, то приложение предпочитает запускаться на более графическом процессоре, если таковой имеется, который мы определяем как «графический процессор, отличный от стандартного» в этой спецификации, чтобы избежать необходимости определять, что такое графический процессор и в каких случаях он может считаться более мощным, чем стандартный графический процессор. Этот ключ является лишь подсказкой, и его поддержка может отсутствовать в зависимости от реализации.	boolean	NO

6 Ключ Exec

Ключ *Exec* должен содержать командную строку. Командная строка состоит из имени исполняемой программы, за которой может следовать один или несколько аргументов. Исполняемая программа может быть указана либо с полным путем, либо с именем только исполняемого файла. Если полный путь не указан, выполняется поиск исполняемого файла в переменной среды *\$PATH*, используемой в среде рабочего стола. Имя или путь исполняемой программы не могут содержать знак равенства (=). Аргументы разделяются пробелом.

Аргументы могут быть закавычены как целое. Если аргумент содержит зарезервированный символ, аргумент должен быть заключен в кавычки. Правила закавычивания аргументов также примени-

мы к имени исполняемого файла или пути исполняемой программы, как указано.

Заключение в кавычки должно выполняться путём заключения аргумента между двойными кавычками и экранировании символа двойной кавычки, символа обратной кавычки (‘), знака доллара (\$) и символа обратной косой черты (\), предшествующего дополнительному символу обратной косой черты. Реализации должны удалить кавычки перед расширением кодов полей и перед передачей аргумента в исполняемую программу. Зарезервированные символы: пробел (" "), табуляция (\t), новая строка (\n), двойная кавычка ("), одинарная кавычка (‘), символ обратной косой черты (\), знак "больше"(>), знак "меньше"(<), тильда (~), вертикальная черта (|), амперсанд (&), точка с запятой (;), знак доллара (\$), звездочка (*), знак вопроса (?), знак хеша (#), скобки ((и)) и символ обратного апострофа (‘).

Обратите внимание, что общее правило экранирования для значений типа строки гласит, что символ обратной косой черты также может быть экранирован как (\\) и что это правило экранирования применяется перед правилом закавычивания. Таким образом, для однозначного представления буквального символа обратной косой черты в аргументе в кавычках в файле ярлыка рабочего стола необходимо использовать четыре последовательных символа обратной косой черты (\\\\\\). Аналогично, буквальный знак доллара в кавычках в файле ярлыка рабочего стола однозначно обозначается ("\\\$").

Был определен ряд специальных кодов полей, которые будут интерпретироваться файловым менеджером или программой запуска программ при их обнаружении в командной строке. Коды полей состоят из символа процента («%»), за которым следует буквенный символ. Сам символ процента должен быть экранирован как \%. Устаревшие коды полей должны быть удалены из командной строки и проигнорированы. Коды полей раскрываются только один раз, строка, используемая для замены кода поля, не должна проверяться на наличие кодов полей.

Командные строки, которые содержат код поля, который не указан в этой спецификации, являются недопустимыми и не должны обрабатываться; в некоторых реализациях может отсутствовать поддержка кодов полей, не указанных в этой спецификации. Расширения, если таковые имеются, следует вводить с помощью нового ключа.

Реализации должны заботиться о том, чтобы не расширять коды полей в несколько аргументов, если это явно не указано в данной спецификации. Это означает, что поля имени, имена файлов и другие замены, которые могут содержать пробелы, должны быть переданы как один аргумент исполняемой программе после расширения.

Хотя ключ Ehes определен так, чтобы иметь значение типа строка, которая ограничена символами ASCII, расширение кода поля может вводить символы не-ASCII в аргументах. Реализации должны позаботиться о том, чтобы все символы в аргументах, передаваемых исполняемой программе, были правильно закодированы в соответствии с применимой настройкой локали.

Распознаются следующие коды полей 3:

Командная строка может содержать не более одного кода поля %f, %u, %F или %U. Если приложение не должно открывать какой-либо файл, коды полей %f, %u, %F и %U должны быть удалены из командной строки и проигнорированы.

Коды полей не должны использоваться внутри аргумента в кавычках, результат раскрытия кода поля внутри аргумента в кавычках не определен. Коды полей %F и %U могут использоваться только в качестве аргумента самостоятельно.

Таблица 3: Коды полей

Код	Описание
%f	Одно имя файла (включая путь), даже если выбрано несколько файлов. Система, считывающая ярлык рабочего стола, должна распознать, что рассматриваемая программа не может обрабатывать несколько аргументов файла, и она должна, вероятно, создавать и выполнять несколько копий программы для каждого выбранного файла, если программа не может обрабатывать дополнительные аргументы типа файла. Если файлы не находятся в локальной файловой системе (т.е. находятся на HTTP или FTP серверах), файлы будут скопированы в локальную файловую систему, и %f будет преобразован так, чтобы указывать на временный файл. Используется для программ, которые не понимают синтаксис URL.
%F	Список файлов. Используйте для приложений, которые могут открывать несколько локальных файлов одновременно. Каждый файл передается как отдельный аргумент исполняемой программе.
%u	Один URL. Локальные файлы могут передаваться как file:URL или как путь к файлу.
%U	Список URL. Каждый URL передается как отдельный аргумент исполняемой программе. Локальные файлы могут передаваться как file:URL или как путь к файлу.
%d	Устаревший
%D	Устаревший
%n	Устаревший
%N	Устаревший.
%i	Ключ иконки ярлыка на рабочем столе расширяется в виде двух элементов: сначала <code>-icon</code> , а затем значение ключа <code>icon</code> . Не должен расширяться до каких-либо аргументов, если ключ <code>Icon</code> пуст или отсутствует.
%c	Переведенное имя приложения, указанное в соответствующем ключе имени в ярлыке рабочего стола.
%k	Местоположение файла рабочего стола в виде URI (если он, например, получен из системы <code>vfolder</code>) или локального имени файла или пустое, если местоположение не известно.
%V	Устаревший.
%M	Устаревший.

7 Активация D-Bus

Приложения, поддерживающие запуск D-Bus, должны реализовывать следующий интерфейс (приведенный в формате XML для внутреннего анализа D-Bus):

Листинг 1: Интерфейс D-bus

```

1 <interface name='org.freedesktop.Application'>
2   <method name='Activate'>
3     <arg type='a{sv}' name='platform_data' direction='in' />
4   </method>
5   <method name='Open'>
6     <arg type='as' name='uris' direction='in' />
7     <arg type='a{sv}' name='platform_data' direction='in' />
8   </method>
9   <method name='ActivateAction'>
10    <arg type='s' name='action_name' direction='in' />
11    <arg type='av' name='parameter' direction='in' />

```

```
12         <arg type='a{sv}' name='platform_data' direction='in' />
13     </method>
14 </interface>
```

Приложение должно назвать свой файл ярлыка рабочего стола в соответствии с рекомендациями по присвоению имен во введении (например, имя файла должно быть как *org.example.FooViewer.desktop*). Приложение должно иметь активируемую службу D-Bus под известным именем, равным имени файла ярлыка рабочего стола с удаленной частью *.desktop* (для нашего примера, *org.example.FooViewer*). Приведенный выше интерфейс должен быть реализован по пути объекта, определяемому следующим образом: начиная с известного имени приложения D-Bus, измените все точки на косую черту и добавьте в начало косую черту. Если найден дефис ('-'), преобразуйте его в подчеркивание ('_'). Для нашего примера это */org/example/FooViewer*.

Метод *Activate* вызывается, когда приложение запускается без открытия файлов.

Метод *Open* вызывается, когда приложение запускается с файлами. Массив строк представляет собой массив URI в UTF-8.

Метод *ActivateAction* вызывается при активации действий на рабочем столе. Параметр *action-name* - это имя действия.

Все методы принимают аргумент *platform-data*, который используется аналогично тому, как могут использоваться переменные среды. В настоящее время только одно поле определяется спецификацией: *desktop-startup-id*. Это должна быть строка с тем же значением, которое будет храниться в переменной среды *DESKTOP_STARTUP_ID*, как указано в спецификации протокола уведомления при запуске.

8 Интерфейсы

Ключ *Implements* можно использовать для объявления одного или нескольких интерфейсов, которые реализует файл рабочего стола.

Каждое имя интерфейса должно соответствовать правилам, используемым для имен интерфейсов D-Bus, но кроме этого они не имеют особого значения. Например, перечисление интерфейса здесь не обязательно означает, что это приложение реализует этот интерфейс D-Bus или даже то, что такой интерфейс D-Bus существует. Это полностью зависит от объекта, который определил конкретный интерфейс, чтобы понять, что оно означает для его реализации.

Хотя разработчик интерфейса должен решить, что означает данное имя интерфейса, вот некоторые рекомендуемые «лучшие практики»:

- интерфейсы должны требовать, чтобы приложение было *DBusActivatable*, включая требование, чтобы файл рабочего стола приложения был назван с использованием соглашения об обратном DNS D-Bus
- имя интерфейса должно соответствовать интерфейсу D-Bus, который приложение экспортирует по тому же пути объекта, что и интерфейс *org.freedesktop.Application*.
- если интерфейс желает предоставить подробную информацию о реализации, он должен сделать это, указав, что разработчики добавляют группу в свой файл рабочего стола с тем же именем, что и интерфейс (например: «*/org.freedesktop.ImageAcquire*»)

Несмотря на рекомендации, интерфейсы могут указывать практически любые мыслимые требования, включая такие (смешные) вещи, как «при запуске через строку *Exec*, ожидается, что приложение

представит окно с набором свойств `_FOO_IDENTIFIER`, после чего сообщение X-клиента будет отправлено в это окно". Другой пример: «все реализации этого интерфейса должны быть помечены как *NoDisplay* и при запуске не будут отображать окна и будут удалять все файлы пользователя без подтверждения».

Те, кто определяет интерфейсы должны заботиться о том, чтобы учитывать вопросы обратной и прямой совместимости при разработке своих интерфейсов.

9 Регистрация типов MIME

Ключ *MimeType* используется для указания типов MIME, с которыми приложение знает, как обращаться. Ожидается, что для некоторых приложений этот список может стать длинным. Ожидается, что приложение сможет разумно открывать файлы этих типов с помощью команды, указанной в ключе *Exec*.

В этом поле не должно быть приоритета для типов MIME или любой другой формы приоритета в файле рабочего стола. Приоритет для приложений обрабатывается вне файлов *.desktop*.

10 Дополнительные действия приложений

Ярлыки рабочего стола типа *Application* могут включать одно или несколько действий. Действие представляет собой дополнительный способ вызова приложения. Средства запуска приложений должны предоставлять их пользователю (например, в виде подменю) в контексте приложения. Это используется для создания так называемых «Quicklists» или «Jumplists».

10.1 Идентификатор действия

Каждое действие идентифицируется строкой в том же формате, что и имена ключей (см. Раздел «Строки файла» 2.3). Каждый идентификатор связан с группой действий, которая должна присутствовать в файле *.desktop*. Группа действий - это группа с именем *Desktop Action %s*, где %s - идентификатор действия.

Недопустимо иметь группу действий для идентификатора действия, не упомянутого в ключе действий. Такая группа действий должна игнорироваться разработчиками.

10.2 Ключи действий

Следующие ключи поддерживаются в каждой группе действий. Если ключ **REQUIRED** отсутствует в группе действий, разработчик должен игнорировать это действие.

10.3 Замечания по реализации

Действия приложения должны поддерживаться разработчиками. Однако, если они не поддерживаются, разработчики могут просто игнорировать ключ *Actions* и соответствующие группы действий *Desktop* и продолжать использовать группу *Desktop Entry*: основной способ описать и вызвать приложение - через ключи *Name*, *Icon* и *Exec* из группы *Desktop Entry*.

Таблица 4: Ключи действий

Ключ	Описание	Тип значения	REQ?
Name	Метка, которая будет показана пользователю. Поскольку действия всегда отображаются в контексте определенного приложения (то есть в виде подменю средства запуска), это должно быть однозначно только в одном приложении и не должно включать имя приложения.	localestring	ДА
Icon	Значок, отображаемый вместе с действием. Если имя является абсолютным путем, будет использован данный файл. Если имя не является абсолютным путем, для определения значка будет использоваться алгоритм, описанный в спецификации темы значков. Реализации могут игнорировать этот параметр.	iconstring	НЕТ
Exec	Программа для выполнения этого действия, возможно, с аргументами. См. Ключ Exec для деталей о том, как работает этот ключ. Ключ Exec необходим, если для DBusActivatable не установлено значение true в основной группе ярлыка рабочего стола. Даже если DBusActivatable имеет значение true, необходимо указать Exec для совместимости с реализациями, которые не понимают DBusActivatable.	String	НЕТ

Не ожидается, что другие компоненты рабочего стола, показывающие списки приложений (например, установщики программного обеспечения), будут предоставлять какой-либо пользовательский интерфейс для этих действий. Поэтому приложения должны включать в себя только те действия, которые имеют смысл как общие средства запуска.

11 Расширение формата

Если стандарт должен быть дополнен новой парой ключ, значение, которая должна быть применима ко всем сторонам поддержки, должно состояться групповое обсуждение. Это предпочтительный метод для внесения изменений. Если одна конкретная сторона желает добавить поле для личного использования, ей следует добавить к ключу строку со строкой *X-PRODUCT*, например, *X-NewDesktop-Foo*, следуя прецеденту, установленному другими стандартами IETF и RFC.

Альтернативно, поля могут быть помещены в их собственную группу, где они могут иметь произвольные имена ключей. Если это так, то группа должна следовать схеме, описанной выше, то есть *[X-PRODUCT GROUPNAME]* или что-то подобное. Эти шаги позволят избежать столкновений пространства имен между различными, но похожими средами.

12 Приложения

12.1 А. Пример рабочего стола

```
[Desktop Entry]
Version=1.0
Type=Application
Name=Foo Viewer
Comment=The best viewer for Foo objects available!
TryExec=fooview
Exec=fooview %F
Icon=fooview
MimeType=image/x-foo;
Actions=Gallery;Create;

[Desktop Action Gallery]
Exec=fooview --gallery
Name=Browse Gallery

[Desktop Action Create]
Exec=fooview --create-new
Name=Create a new Foo!
Icon=fooview-new
```

12.2 В. В настоящее время зарезервировано для использования в KDE

По историческим причинам KDE использует некоторые специфичные для KDE расширения, которые в настоящее время не имеют префикса X-KDE-префикса.

- Специальные ключи KDE: ServiceTypes, DocPath, InitialPreference
- Специальные типы KDE: ServiceType, Service и FSDevice

KDE также использовал ключ Keywords до его стандартизации, используя запятые вместо точек с запятой в качестве разделителей. Во время стандартизации поле было префиксом с префиксом X-KDE, но вилка Trinity все еще использовала вариант без префикса.

KDE использует следующие дополнительные ключи в ярлыках рабочего стола типа FSDevice 5:

Таблица 5: Специальные ключи FSDevice

Ключ	Описание	Тип значения
Dev	Устройство для монтирования.	строка
FSType	Тип файловой системы для монтирования.	строка
MountPoint	Точка монтирования рассматриваемого устройства.	строка
ReadOnly	Указывает, доступно ли устройство только для чтения.	логический
UnmountIcon	Значок для отображения, когда устройство не подключено. Установленные устройства отображают значок с помощью Icon.	iconstring

12.3 С. Устаревшие элементы

Поскольку этот стандарт довольно старый, есть некоторые устаревшие элементы, которые могут или не могут использоваться несколькими реализациями.

- *Type = MimeType* устарел, так как теперь существует новый стандарт для этого, см. Спецификацию базы данных Shared MIME-info для получения дополнительной информации. В результате шаблоны ключей (различные расширения имен файлов, связанные с типом MIME) и *DefaultApp* (приложение по умолчанию, связанное с этим типом MIME) также устарели.
- Использование *.kdelnk* вместо *.desktop* в качестве расширения файла не рекомендуется.
- Использование *[KDE Desktop Entry]* вместо *[Desktop Entry]* в качестве заголовка не рекомендуется.
- Ключ *Encoding* устарел. Он использовался для указания того, были ли ключи типа *localestring* закодированы в UTF-8 или в указанной локали. Возможные значения: UTF-8 и Legacy-Mixed. См. Приложение D, Унаследованное смешанное кодирование (устарело) для получения более подробной информации.
- Устаревшие коды полей *Exec: %m* (мини-значок, связанный с записью на рабочем столе, должен быть расширен в виде двух аргументов, *-miniicon* и содержимого ключа *MiniIcon*, его также можно игнорировать, подставив его при отсутствии аргументов), *%v* (устройство, указанное в разделе *Dev* в файле рабочего стола), *%d* (каталог файла), *%D* (каталоги файлов), *%p* (базовое имя файла) и *%N* (базовые имена файлов).
- Устаревшие ключи: *MiniIcon* (маленькая иконка для меню и т.д.), *TerminalOptions* (если программа запускается в терминале, любые параметры, которые следует передать эмулятору терминала перед фактическим выполнением программы), *Protocols*, *Extensions*, *BinaryPattern*, *MapNotify*.
- Ключи *SwallowTitle* и *SwallowExec* устарели. Ключ *SwallowTitle* имеет тип *localestring* и указывает заголовок окна, если он размещён на панели. Ключ *SwallowExec* имеет тип *string* и указывает программу для выполнения при нажатии приложения, размещённого на панели.
- Ключ *SortOrder* устарел. Он имеет тип *string(s)* и может использоваться для указания порядка отображения файлов. Спецификация меню рабочего стола определяет другой механизм для определения порядка пунктов меню.
- Ключ *FilePattern* устарел. Значение представляет собой список регулярных выражений, с которыми нужно сопоставить файловый менеджер, чтобы определить, должен ли отображаться значок этой записи. Обычно просто имя основного исполняемого файла и друзей.
- Исторически некоторые логические значения были представлены числовыми записями 0 или 1. С этой версией стандарта они теперь должны быть представлены в виде логической строки. Однако, если реализация читает запись рабочего стола до 1.0, она должна интерпретировать 0 и 1 как *false* и *true* соответственно.
- Исторически списки были разделены запятыми. Это не согласуется с другими списками, разделенными точкой с запятой. При чтении десктоп ярлыков версии до 1.0, списки, разделенные запятыми, должны по-прежнему поддерживаться.

12.4 D. Кодирование Legacy-Mixed (устарело)

Кодировка Legacy-Mixed соответствует традиционной кодировке файлов рабочего стола в старых версиях файлов рабочего стола GNOME и KDE. В этой кодировке кодирование локали каждого ключа

строки определяется тегом локали для этого ключа, если он есть, вместо UTF-8. Для ключей без языкового тега значение должно содержать только символы ASCII.

Если в файле указана неподдерживаемая кодировка, реализация должна либо игнорировать файл, либо, если пользователь запросил прямую операцию над файлом (например, открыть его для редактирования), отобразить для пользователя соответствующую индикацию ошибки.

В отсутствие ключа кодировки реализация может выбрать автоматическое определение кодировки файла с использованием таких факторов, как:

- Расположение файла в файловой системе
- Является ли содержимое файла действительным UTF-8

Если реализация не выполняет такое автоматическое обнаружение, она должна обрабатывать файл без ключа кодирования так же, как файл с неподдерживаемым ключом кодирования.

Если тег *locale* включает в себя часть *.ENCODING*, то это определяет кодировку для строки. В противном случае кодировка определяется языком или парой *lang_COUNTRY* из тега языкового стандарта в соответствии со следующей таблицей 6.

Таблица 6: Кодировки

Encoding	Aliases	Tags
ARMSCII-8 (*)		hy
BIG5		zh_TW
CP1251		be bg
EUC-CN	GB2312	zh_CN
EUC-JP		ja
EUC-KR		ko
GEORGIAN-ACADEMY (*)		
GEORGIAN-PS (*)		ka
ISO-8859-1		br ca da de en es eu fi fr gl it nl no pt sv wa
ISO-8859-2		cs hr hu pl ro sk sl sq sr
ISO-8859-3		eo
ISO-8859-5		mk sp
ISO-8859-7		el
ISO-8859-9		tr
ISO-8859-13		lt lv mi
ISO-8859-14		cy ga
ISO-8859-15		et
KOI8-R		ru
KOI8-U		uk
TCVN-5712 (*)	TCVN	vi
TIS-620		th
VISCII		

Кодирование Перечисленные здесь имена кодировок, обычно является каноническим именем для кодировки в средстве *iconv* библиотеки GNU C. Кодировки, отмеченные (*), в настоящее время не поддерживаются библиотекой GNU C; по этой причине реализации могут предпочесть игнорировать строки в файлах рабочего стола, которые соответствуют этой кодировке. Десктом файлы ярлыков с этими кодировками в настоящее время редки или отсутствуют.

Псевдонимы Другие имена для кодировок, найденных в существующих файлах ярлыков рабочего стола.

Теги Тэги языков, для которых эта кодировка является кодировкой по умолчанию.

Эта таблица охватывает все теги и кодировки, которые, насколько известно, используются в настоящее время. Разработчики могут выбрать поддержку кодировок, не входящих в приведенный выше набор. Для тегов без значений по умолчанию, перечисленных в таблице выше, создатели файлов ярлыков рабочего стола должны указать часть *.ENCODING* тега *locale*.

Сопоставление части *.ENCODING* тега *locale* с именем или псевдонимом локали следует выполнить путем удаления всех знаков препинания как из тега, так и из имени или псевдонима, преобразования имен и псевдонимов в нижний регистр и сравнения результата. Это необходимо, потому что, например, Big5 часто встречается вместо BIG5 и georgianacademy вместо GEORGIAN-ACADEMY. Создатели файлов рабочего стола должны, однако, использовать имя в том виде, в котором оно указано в столбце «Кодировка» выше.