

# Процедуры

**Процедура** – это группа операторов, исполняемая как единое целое и предназначенная для выполнения некоторых действий. Процедуры не возвращают значения (в особых случаях могут возвращать команду возврата). Процедура может быть оформлена либо как обычная процедура, либо как процедура-метод.

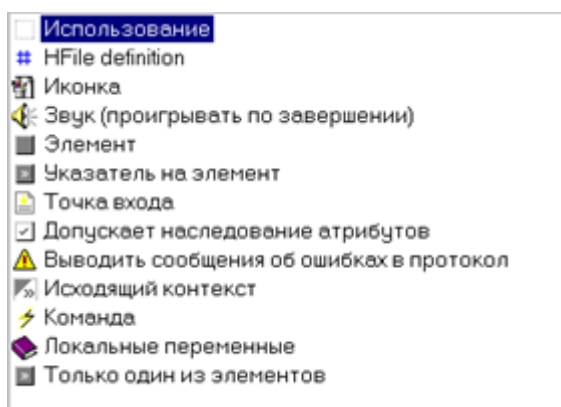
**Процедуры-методы** применяются только при разработке процедур акцепта, деакцепта и удаления объекта. В этих случаях ядро программы должно выполнять специальные действия, не требуемые в других ситуациях.

Процедуры различаются по способу создания. В одном случае процедура разрабатывается программистом на языке программирования (чаще всего на Visual C++) и подключается к Domino с помощью динамической библиотеки (файл с расширением dll). В проекте находится только декларация этой процедуры. Атрибут 'Точка входа' в описании такой процедуры обязательно должен быть заполнен.

В другом случае процедуру разрабатывает проектировщик на языке скриптов. Описание и код такой процедуры полностью находятся в проекте.

Раздел '*Процедуры*' может находиться либо на верхнем уровне любой библиотеки, либо внутри объектных разделов. В разделе '*Процедуры*' на верхнем уровне библиотеки описываются те процедуры, которые одновременно обрабатывают несколько объектов. В объектных разделах, как правило, находятся процедуры для обработки одного конкретного объекта.

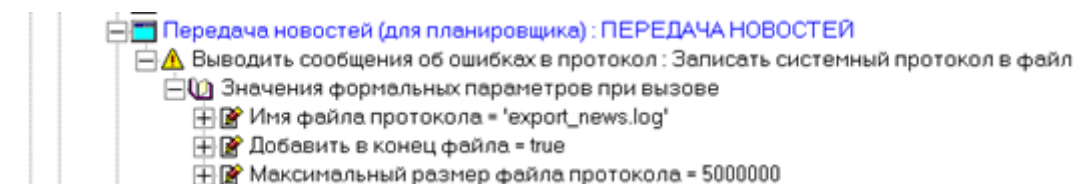
Для процедур поддерживается механизм наследования. Процедуры самого верхнего уровня являются базовыми. У базовых процедур имеются следующие атрибуты:



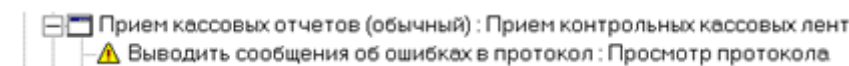
- **HFile definition** – используется программистами.
- **Использование** – условие доступа проектировщиков к процедуре.
- **Иконка** – для отображения процедуры в сценариях и панели инструментов.

- **Звук (проигрывать по завершении)** – Используется при вызове процедуры из панели инструментов или из сценария. По завершении выполнения процедуры будет проиграна указанная мелодия.
- **Элемент** – применяется для указания специальных атрибутов процедуры.
- **Указатель на элемент** – применяется для указания специальных атрибутов процедуры.
- **Точка входа** – место реализации процедуры (имя данной процедуры с точки зрения программиста и название dll, в которой данная процедура содержится).
- **Допускает наследование атрибутов** – при установленном признаке в описании наследника можно будет переопределить атрибуты.
- **Выводить сообщения об ошибках в протокол** – при заполненном атрибуте сообщения об ошибках будут записаны в протокол, а не высвечены на экране. Атрибут может содержать либо ссылку на вид просмотра (протокол будет отображен на экране), либо ссылку на процедуру, которая запишет сообщения об ошибках в файл (протокол будет сохранен в файле).

Примеры использования:



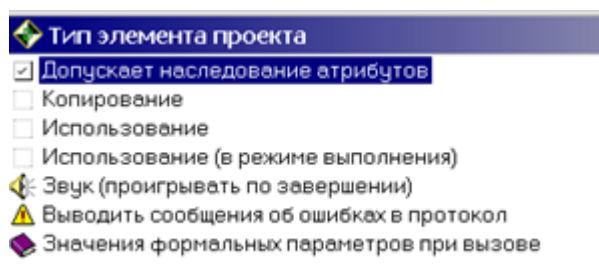
Сообщения об ошибках будут добавлены в файл `export_news.log`.



Протокол будет высвечен на экране.

- **Исходящий контекст** – в коде процедуры будут доступны параметры перечисленных объектов
- **Команда** – не используется.
- Локальные переменные – список локальных переменных.
- **Только один из элементов** – назначение аналогично атрибуту 'Указатель на элемент', но позволяет проектировщику выбирать только один атрибут из указанного списка.

Для процедур, не являющихся базовыми, можно указать следующие атрибуты:



- **Копирование** – условие копирования процедуры проектировщиками.
- **Использование** – условие доступа проектировщиков к процедуре.
- **Использование (в режиме выполнения)** – условие доступа пользователей к процедуре.
- **Значения формальных параметров при вызове** - список значений формальных параметров. Раздел заполняется при вызове процедуры.

Процедуры на языке скриптов имеют множество дополнительных атрибутов. Подробнее о таких процедурах написано в книге **‘Язык скриптов’**.

Процедуры могут иметь параметры настройки, назначение, количество и внешний вид которых зависит от разработчика.

Процедуры состоят из операторов и локальных данных (константы и переменные). Локальные данные существуют только во время выполнения объемлющей их процедуры, они создаются автоматически в начале выполнения процедуры и исчезают, как только выполнение процедуры заканчивается. Операторы процедуры могут обращаться к ее локальным данным или к тем данным, которые видны изнутри процедуры (глобальные и контекстные переменные, параметры объектов БД и проектные элементы).

Во время активизации процедуры ей можно передать параметры. Параметры – это, по сути, начальные значения локальных переменных.

Параметры могут быть обязательные, необязательные и неявные.

При вызове процедуры для обязательных параметров всегда должны быть указаны значения.

Для необязательных параметров значения можно не указывать. Обычно автор процедуры задает начальные значения таких параметров. Эти начальные значения и будут применены, если при вызове процедуры не указать иные значения.

Параметр является неявным в том случае, если заполнение этого параметра происходит вне вызова процедуры. Подробнее о видах параметров написано в предыдущей главе, поскольку параметры разных видов применяются для функций значительно чаще.

## Процедура-метод

Процедура может быть оформлена либо как обычная процедура, либо как процедура-метод.

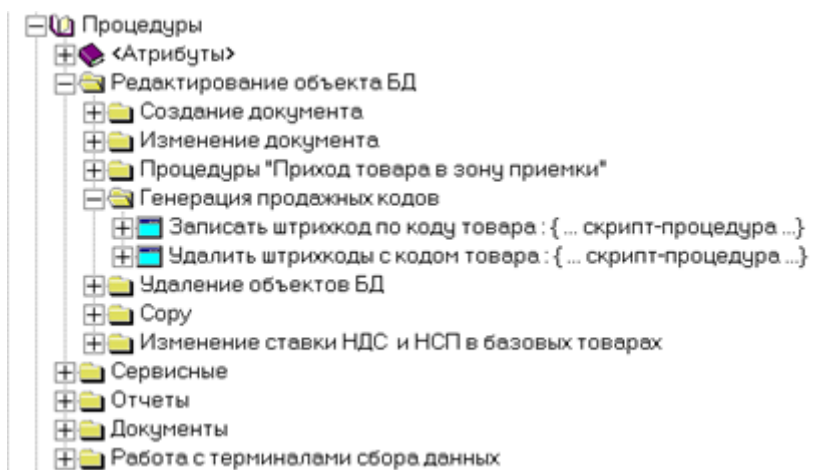
Процедуры-методы применяются только при разработке процедур акцепта, деакцепта и удаления объекта. В этих случаях ядро программы должно выполнять специальные действия, не требуемые в других ситуациях.

Отличие процедур от процедур-методов заключается в команде возврата при возникновении ошибки в процессе выполнения. Если произошла ошибка, то процедура возвращает команду `cmdCONTINUE`, а процедура-метод – команду `cmdRETURN_CANCEL`.

Если выполнение завершилось без ошибок, то и обычная процедура, и процедура-метод возвращают команду `cmdCONTINUE`. Для процедур и процедур-методов есть также возможность явного задания команды возврата с помощью оператора `RETURN`. Подробнее о командах возврата написано в книге **‘Язык скриптов’**.

## Создание группы процедур, объединенных в одной папке

Группа процедур создается для объединения в одной папке (под одним названием) нескольких процедур. Это делается для того, чтобы было легче ориентироваться в многообразии элементов одного вида в дереве проекта.



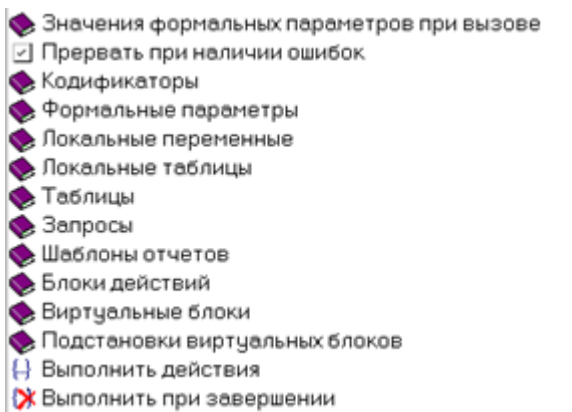
Группа процедур может содержать неограниченное число процедур, Групп процедур и Подстановок процедур.

## Подстановка процедуры

Подстановка применяется только для процедур на языке скриптов.

В подстановке можно добавить значения в описательные атрибуты процедуры и полностью изменить те атрибуты, которые задают алгоритм выполнения процедуры.

Рассмотрим основные атрибуты процедуры на языке скриптов.



Большинство атрибутов являются описательными. Если атрибут описан в подстановке, то значение из подстановки добавляется к значению соответствующего атрибута из описания процедуры.

Атрибуты *‘Подстановки виртуальных блоков’*, *‘Выполнить действия’*, *‘Выполнить при завершении’* задают алгоритм действий. Если атрибут описан в подстановке, то значение из подстановки заменяет значение соответствующего атрибута из описания процедуры.

Значение признака *‘Прервать при наличии ошибок’* в подстановке также заменяет значение этого признака в описании процедуры.

---

Версия #3

[Демонов Сергей](#) создал Thu, Mar 24, 2022 5:40 AM

[Демонов Сергей](#) обновил Thu, Mar 24, 2022 5:48 PM