

## Примеры использования PSCAD

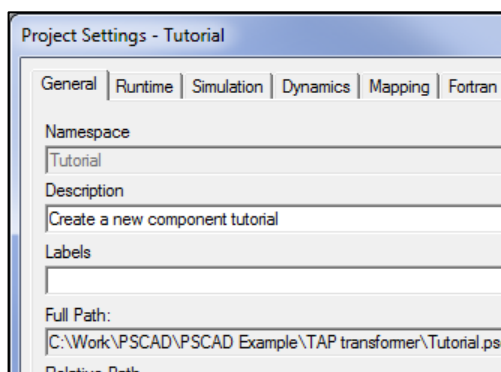
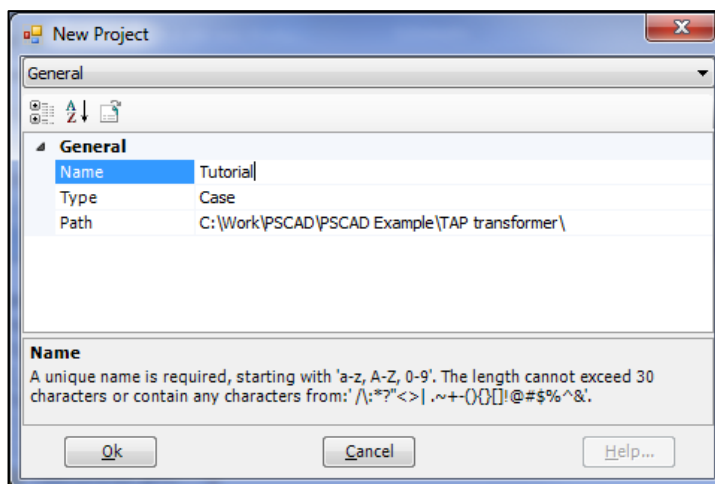
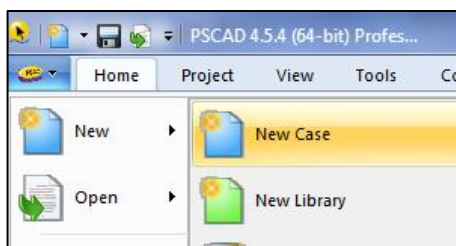


## ПРИМЕР №2. СОЗДАНИЕ НОВОГО КОМПОНЕНТА

**Цель работы.** Создание собственного электрического компонента источника переменного тока, описание его портов и алгоритма работы, внедрение в модель и выполнение симуляции.

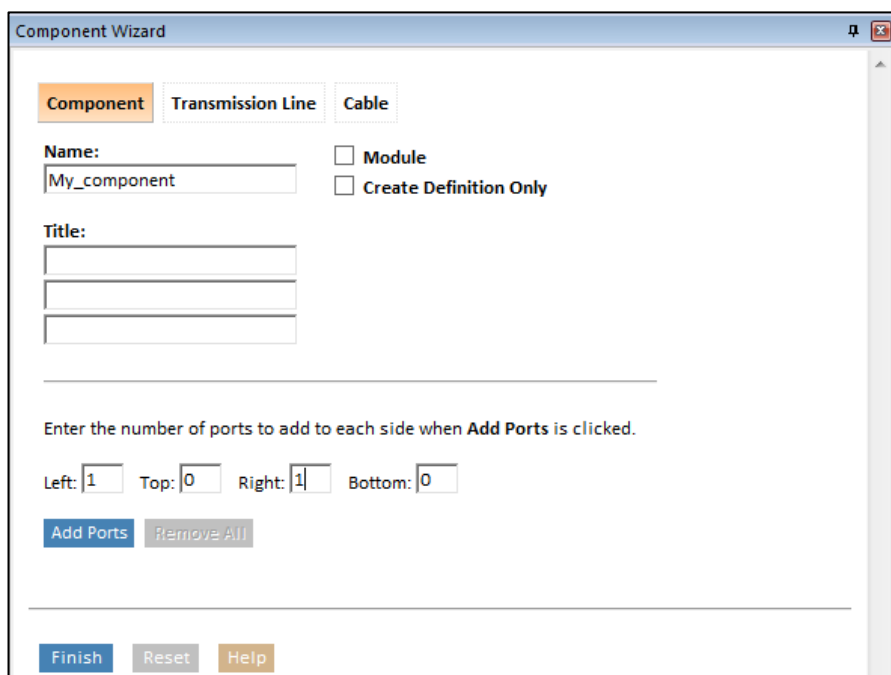
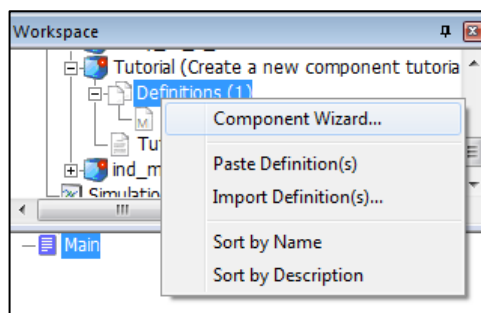
### 1. СОЗДАНИЕ ПРОЕКТА И НОВОГО КОМПОНЕНТА

Запустите PSCAD, создайте новый проект **case**, сохраните его под уникальным именем **Tutorial**, отредактируйте параметры проекта и добавьте его краткое описание "Create a new component tutorial", как это показано на рисунках ниже.

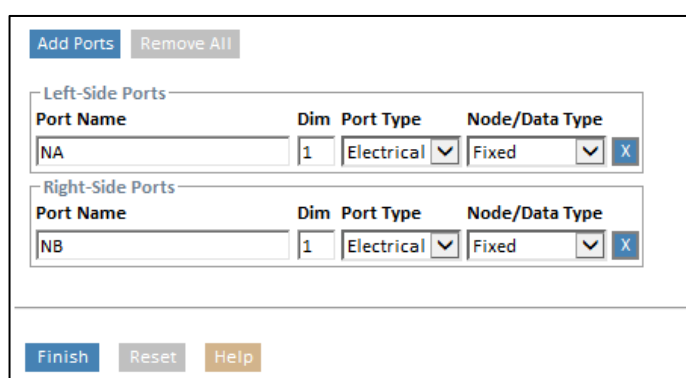


В упомянутом проекте запустите мастер создания нового компонента **Component Wizard**. Назовите новый компонент **My\_component**. Укажите, что компонент будет иметь 1 порт слева **Left** и 1 порт справа **Right**, нажмите кнопку добавления портов **Add Ports**.



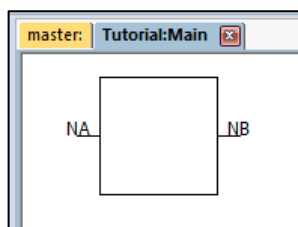


В открывшемся окне настройки создаваемых портов задайте их тип, как электрический порт **Electrical**, и признак размещения порта, как фиксированный **Fixed**. Присвойте индивидуальные наименования портам слева **NA** и справа **NB** и нажмите кнопку **Finish**.

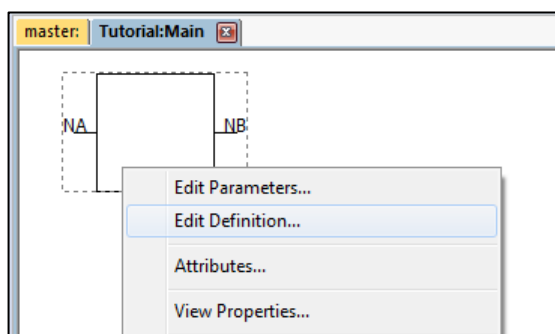


На рабочем поле появится прямоугольник с портами, который можно разместить в удобном месте по клику мышкой. Таким образом, заготовка нового компонента с заданным количеством, типом и наименованием портов успешно осуществлена и помещена на рабочее поле.

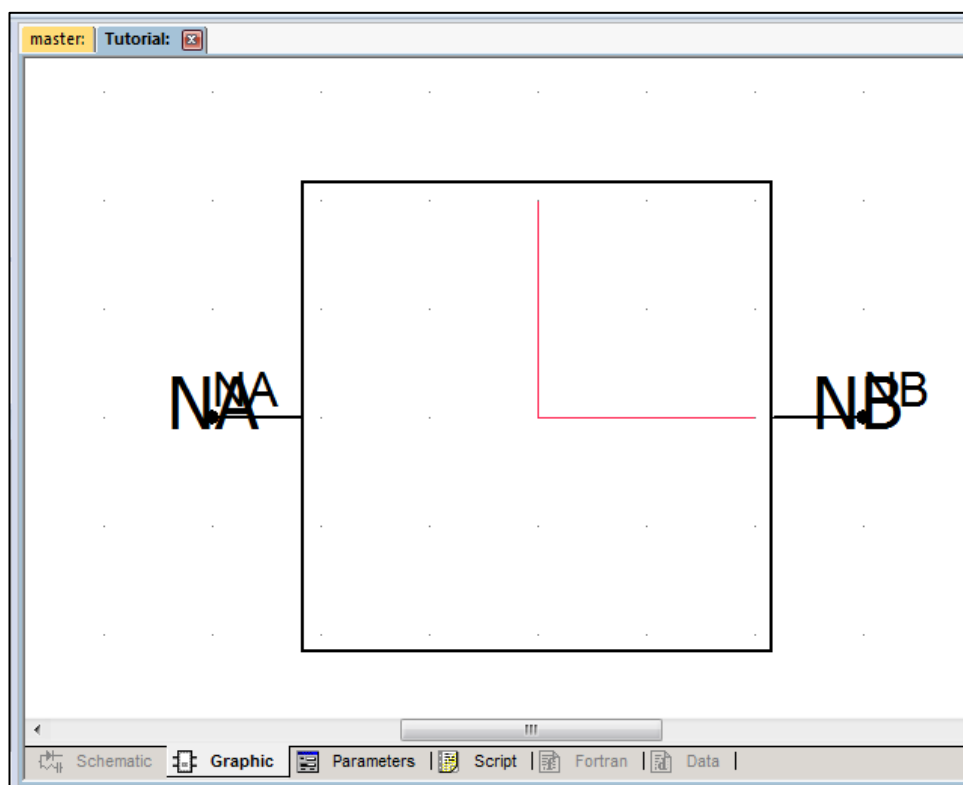




Необходимо задать внутренний алгоритм и параметры функционирования созданного компонента. Для этого кликните правой кнопкой мыши на компонент и выберите пункт редактирования описания компонента **Edit Definition**.

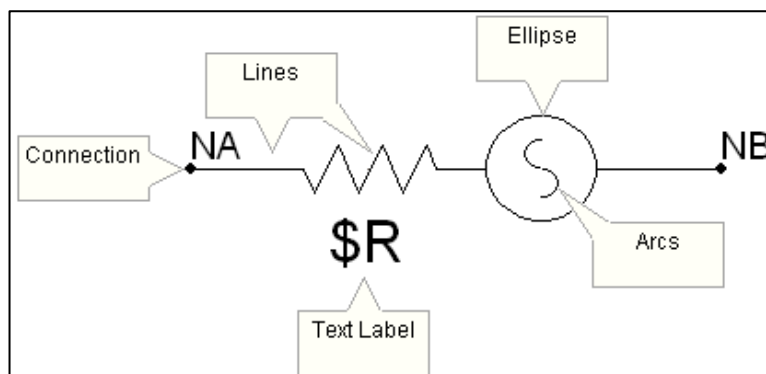


После этого откроется окно графического редактора компонента на активной закладке **Graphic**, находящейся внизу окна вместе с закладками **Parameters**, **Script**. Остальные закладки в режиме редактирования описания компонента становятся недоступными.



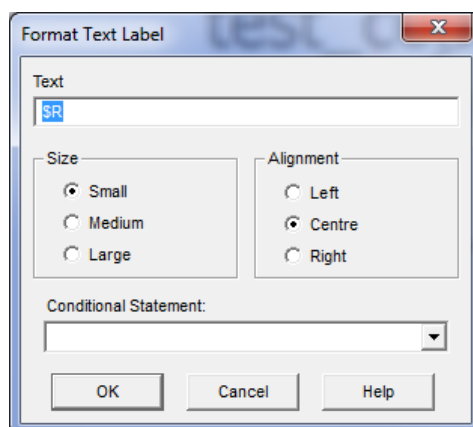
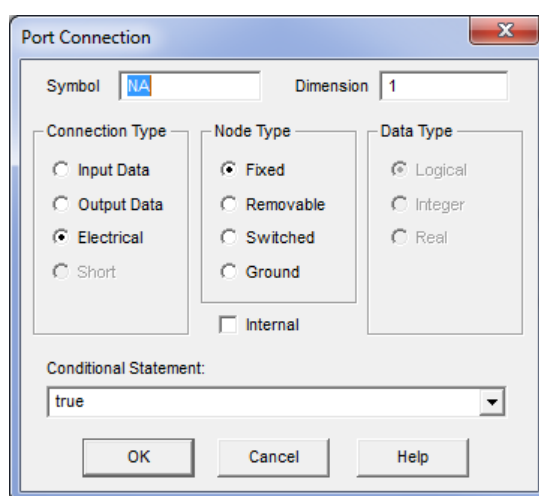
В этом редакторе можно сформировать внешний вид компонента с использованием стандартных графических примитивов (линия **line**, прямоугольник **rectangle**, дуга **arc**, окружность **ellipse**, порт **port**), как это показано на рисунке ниже.





ПРИМЕЧАНИЕ: Следите, чтобы графическое изображение компонента имело размер, соответствующий другим компонентам из библиотеки.

При необходимости измените настройки портов NA и NB.

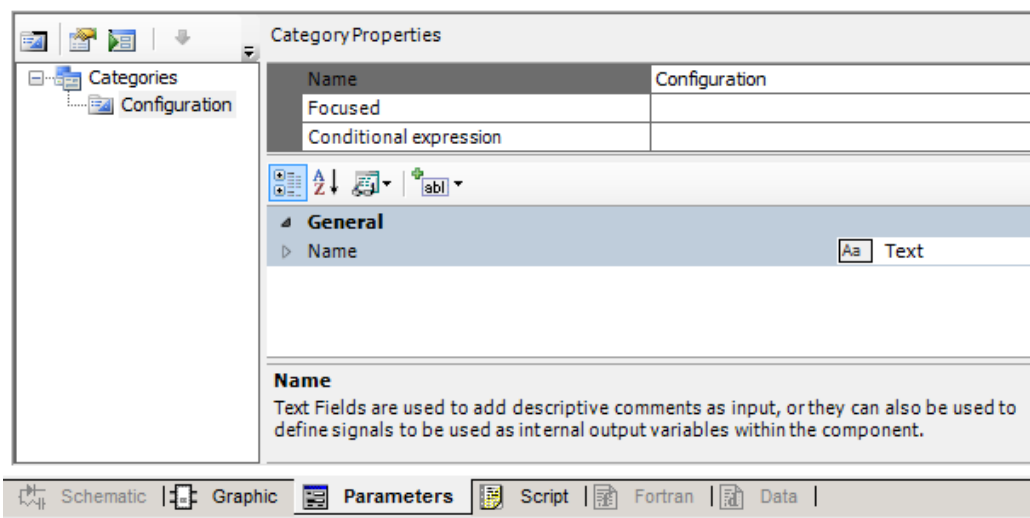


Добавьте надпись Label с текстом \$R для отображения значения внутреннего сопротивления компонента на схеме и задайте форматирование надписи по размеру – малый Small, и центровку – по центру Centre:

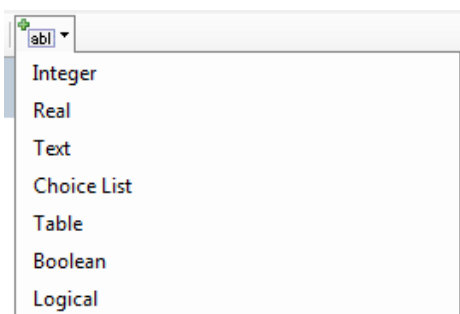
## 2. СОЗДАНИЕ ИЗМЕНЯЕМЫХ ПАРАМЕТРОВ КОМПОНЕНТА

Не выходя из редактора описания компонента, перейдите на закладку параметров Parameters. При этом откроется менеджер параметров компонента с корневой категорией Configuration и пока единственным в ней полем параметра Name.





Добавьте новые ячейки параметров нажатием на кнопку Add Parameter Field.



Последовательно добавьте следующие параметры, задаваемые как действительные числа **Real**, и заполните их поля описаний **Description**:

- действующее значение напряжения **Voltage Magnitude (RMS)**;
- частота **Frequency**;
- время нарастания **Ramp Up Time**;
- сопротивление **Resistance**.

Поля символа **Symbol** используются при задании имени переменной для каждого параметра. Присвойте каждому из них уникальное имя, **Vrms**, **f**, **tr** и **R**, соответственно. Укажите в поле единицы измерения по умолчанию **Defaults Units** следующие обозначения киловольт **kV**, Герц **Hz**, секунда **sec** и Ом **ohm**, соответственно, а также минимальный **Minimum** и максимальный **Maximum** пределы изменения вводимых для каждого параметра значений. Поле типа данных **Data Type** каждого из параметров укажите, как константа **Constant**. Это позволит изменять конфигурацию компонента до симуляции, но обеспечит возможность принимать разные значения параметров при многократных вхождении (запусках симуляции). Подробнее это пояснено в следующей главе этого урока при описании сценария компонента. Кроме этого, в разделах справки по настройке симуляции **Runtime Configuration** и при описании модуля многократного вхождения **Multiple Instance Modules** приведены подробные разъяснения об использовании типа данных константа.



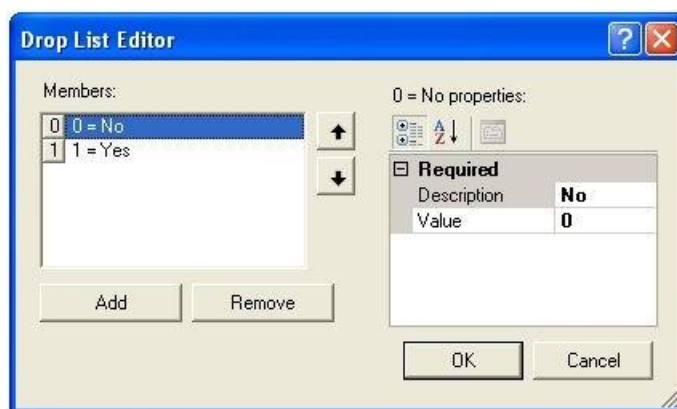
Input Fields:		
<input checked="" type="checkbox"/>	Voltage Magnitude (RMS)	abl Real
<input checked="" type="checkbox"/>	Frequency	abl Real
	Description	Frequency
	Symbol	f
	Default units	Hz
	Minimum value	-1e+38
	Maximum value	1e+38
	Data type	Constant
	Default value	60.0
	Conditional statement	
<input checked="" type="checkbox"/>	Ramp Up Time	abl Real
<input checked="" type="checkbox"/>	Resistance	abl Real
	Description	Resistance
	Symbol	R
	Default units	ohm
	Minimum value	-1e+38
	Maximum value	1e+38
	Data type	Constant
	Default value	1.0
	Conditional statement	

### Параметры полей ввода для частоты и сопротивления

Добавьте еще один параметр, выбираемый из списка вариантов Choice List и введите его описание **Is This Source Grounded?** (источник заземлен?). Присвойте имя переменной для этого параметру **gnd**, а затем в поле **EditDroplist** кликните по символу многоточия и в открывшемся окне создайте список на два варианта: 1 = Да Yes, 0 = Нет No.

Input Fields:		
<input checked="" type="checkbox"/>	Voltage Magnitude (RMS)	abl Real
<input checked="" type="checkbox"/>	Frequency	abl Real
<input checked="" type="checkbox"/>	Ramp Up Time	abl Real
<input checked="" type="checkbox"/>	Resistance	abl Real
<input checked="" type="checkbox"/>	Is This Source Grounded?	abl Choice
	Description	Is This Source Grounded?
	Symbol	gnd
	Default value	0
	Conditional statement	
	Choice List	(Collection)

### Параметры списка вариантов (Is This Source Grounded?)



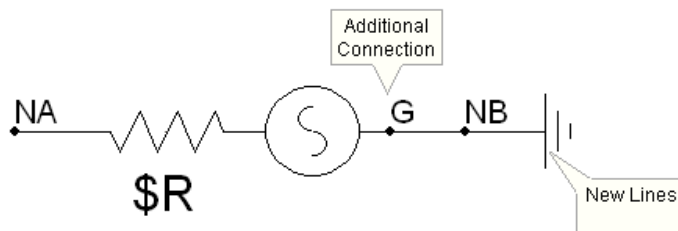
### Редактор списка вариантов



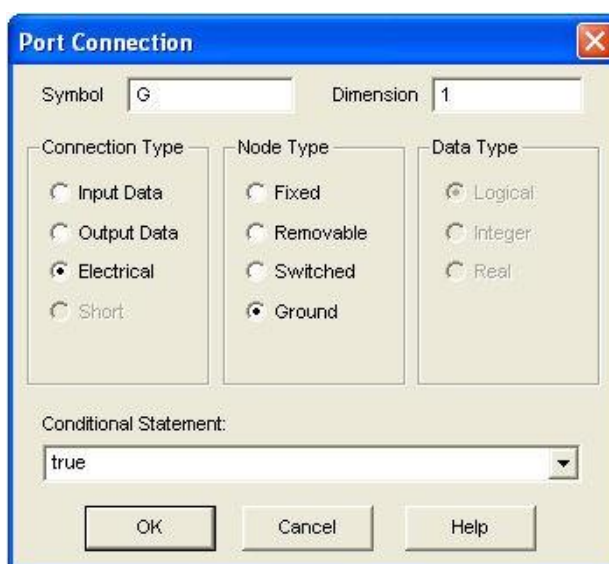
### 3. УСЛОВНЫЕ КОНСТРУКЦИИ И ИЗМЕНЯЕМЫЙ ВНЕШНИЙ ВИД

Для большей практичности создаваемый компонент наделим возможностью заземления в зависимости от изменяемого пользователем параметра по выбору **Is This Source Grounded?**. Для этого параметра по выбору вслед за заданным значением автоматически будет изменяться внешний вид компонента и значения его электрических параметров.

В графическом редакторе компонент **Graphic section** отредактируйте графическое изображение компонентов, как показано на рисунке ниже, где **G** будет портом для заземления.



Внешний вид компонента с заземлением

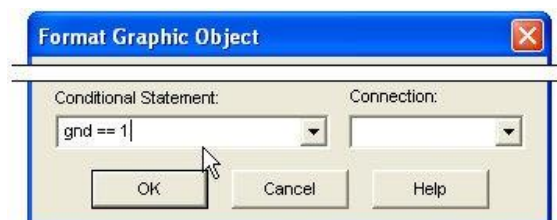


Настройка порта G

В поле задания условных выражений **Conditional Statement** для всех вновь добавленных графических примитивов (линий), изображающих знак заземления, добавьте условие, чтобы они становились видимыми / невидимыми **visible/invisible** в соответствии с состоянием параметра **Is This Source Grounded?**, а точнее в соответствии со значением его переменной **gnd**. Таким образом, если параметр **Is This Source Grounded?** задан как **Yes**, то должны отображаться графические примитивы символа заземления.

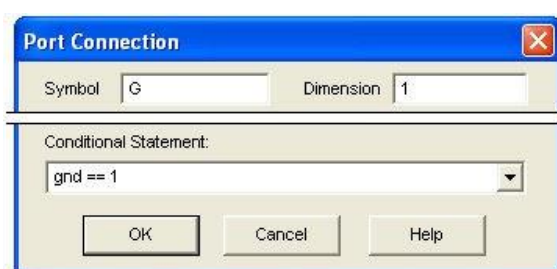




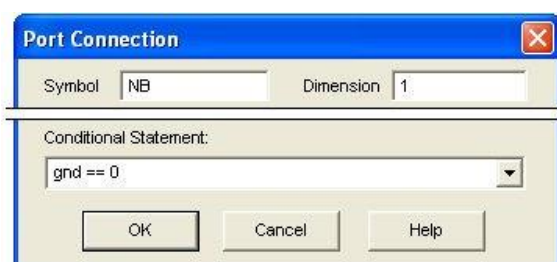


### Редактирование условных выражений для графических примитивов

В поле задания условных выражений **Conditional Statement** для портов **G** и **NB**, добавьте условия, чтобы они включались или отключались в зависимости от состояния переменной **gnd** параметра **Is This Source Grounded?**. Т.е. если параметр **Is This Source Grounded?** имеет значение да **Yes**, то порт **G** должен быть включен, а порт **NB** должен быть отключен и наоборот:



### Редактирование условных выражений порта G



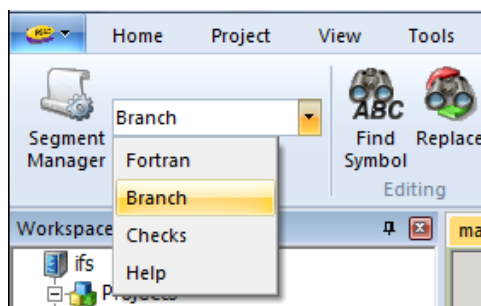
### Редактирование условных выражений порта NB

Не смущайтесь, если на схеме компонента порты **G** и **NB** расположены рядом друг с другом. При размещении портов на изображении компонента старайтесь подобрать наиболее интуитивно понятное положение для портов с учетом того, что они не могут включаться одновременно. Таким образом, порт **NB** было бы лучше поместить в конце линии, как наиболее очевидную точку для внешних подключений.

## 4. ОПИСАНИЕ ЭЛЕКТРИЧЕСКИХ ЦЕПЕЙ

Перейдите в закладку для редактирования сценариев **Script** и выберите в поле выбора разделов **Segment Manager** пункт **Branch** для редактирования описания электрически цепей.





Затем в текстовом редакторе введите текстовое описание электрических цепей компонента в формате EMTDC. Для простоты будет использоваться активное сопротивление. Так как значение сопротивления задано типом константа **Constant**, то его значение будет определяться в момент инициализации компонента (смотрите следующий раздел). Для начала задайте ненулевое значение сопротивления для существующей цепи, что будет служить признаком наличия сопротивления. Если используется тип **Literal**, с именем этого параметра, то при компиляции будет произведена замена ссылки на заданное значение (т.е. вместо ссылки **\$R** будет отображаться числовое значение 1.0).

```
#IF gnd == 1
  BRN = $NA $G SOURCE 1.0 0.0 0.0
#ELSE
  BRN = $NA $NB SOURCE 1.0 0.0 0.0
#ENDIF
```

Операторы компиляции по условию (**#IF**, **#ELSE** и **#ENDIF**) в зависимости от значения переменной **gnd**, соответствующей параметру списка **Is This Source Grounded?** описывают электрическую цепь компонента и между какими портами она пролегает: **NA** и **NB** или **NA** и **G**. Например, если для параметра **Is this source grounded?** задан ответ **YES**, то переменная **gnd** станет равной **1** и при компиляции проекта будет использована вторая строка сверху, которая описывает электрическую цепь между портами **NA** и **G**, содержащую в себе источник **SOURCE**, активное сопротивление 1.0 Ом, индуктивность 0.0 Гн, емкость 0.0 мкФ. Здесь, кроме указания на тип сопротивления, включенного между портами **NA** и **NB** или **NA** и **G**, также объявляется идеальный источник напряжения при помощи оператора **SOURCE**. Электрической цепи присваивается имя **BRN**. Это позволит симулятору EMTDC локализовать эту цепь в общей электрической схеме модели и управлять источником напряжения (использование цепи **BRN** будет показано далее).

Для получения дополнительной информации о присвоении имени электрической цепи и о синтаксисе смотрите раздел **Branch segment**.

## 5. ИНИЦИАЛИЗАЦИЯ ПАРАМЕТРОВ ИСТОЧНИКА НАПРЯЖЕНИЯ

Для упрощения работы по описанию параметров источник напряжения **SOURCE** можно воспользоваться внутренними библиотечными подпрограммами симулятора EMTDC. Для этого потребуется правильно задать аргументы вызова подпрограмм. Будут использоваться следующие подпрограммы:



- E\_BRANCH\_CFG – подпрограмма инициализации параметров электрических цепей.
- E\_1PVSRC\_CFG – подпрограмма инициализации параметров однофазного источника напряжения. Подпрограмма использует модель элемента основной библиотеки под названием Single-Phase Voltage Source Model 2.
- EMTDC\_1PVSRC – подпрограмма действующего однофазного источника напряжения из основной библиотеки Single-Phase Voltage Source Model 2.

Перейдите в закладку для редактирования сценариев Script и выберите в поле выбора разделов Segment Manager пункт Fortran.

Впишите следующий программный код скрипта в рабочем поле (скопируйте ниже-приведенный текст)

```
#BEGIN
    CALL E_BRANCH_CFG($BRN,$SS,1,0,0,$R,0.0,0.0)
    CALL E_1PVSRC_CFG(1,0,1,$Vrms,$f,0.0,$R,0.0,0.0,0.0,0.0,$tr)
#ENDBEGIN
#STORAGE LOGICAL:1 INTEGER:6 REAL:8 RTCF:4
#LOCAL REAL RVD1_1
#LOCAL REAL RVD1_2
#LOCAL REAL RVD1_3
#LOCAL REAL RVD1_4
! Single Phase AC source: Type: R
RVD1_1 = RTCF(NRTCF)
RVD1_2 = RTCF(NRTCF+1)
RVD1_3 = RTCF(NRTCF+2)
RVD1_4 = RTCF(NRTCF+3)
NRTCF = NRTCF + 4
CALL EMTDC_1PVSRC($SS,$BRN,RVD1_4,.TRUE.,RVD1_1,RVD1_2,RVD1_3)
```

**Внимание!** Синтаксис языка Fortran требует использования не менее 6 пробелов для отступа в начале каждой новой строки программы. Редактор автоматически подсвечивает зеленым цветом этот интервал отступа (смотрите рисунок ниже).

```
1 #BEGIN
2     CALL E_BRANCH_CFG($BRN,$SS,1,0,0,$R,0.0,0.0)
3     CALL E_1PVSRC_CFG(1,0,1,$Vrms,$f,0.0,$R,0.0,0.0,0.0,0.0,$tr)
4 #ENDBEGIN
5 #STORAGE LOGICAL:1 INTEGER:6 REAL:8 RTCF:4
6 #LOCAL REAL RVD1_1
7 #LOCAL REAL RVD1_2
8 #LOCAL REAL RVD1_3
9 #LOCAL REAL RVD1_4
10 ! Single Phase AC source: Type: R
11     RVD1_1 = RTCF(NRTCF)
12     RVD1_2 = RTCF(NRTCF+1)
13     RVD1_3 = RTCF(NRTCF+2)
14     RVD1_4 = RTCF(NRTCF+3)
15     NRTCF = NRTCF + 4
16     CALL EMTDC_1PVSRC($SS,$BRN,RVD1_4,.TRUE.,RVD1_1,RVD1_2,RVD1_3)
```

## БЛОК #BEGIN

Блок скрипта между операторами #BEGIN и #ENDBEGIN является кодом инициализации перед началом симуляции. При компиляции проекта этот код будет помещен в



общую секцию BEGIN. В этом примере производится вызов двух подпрограмм EMTDC, которые инициализируют параметры электрической цепи BRN и однофазного источника напряжения Source. Описание аргументов для каждой подпрограммы приведено ниже.

E\_BRANCH\_CFG (NBR, M, ER, EL, EC, RO, HL, FC)

- **NBR** количество настраиваемых ветвей электрических цепей.
- **M** номер подсистемы где находится ветвь.
- **ER** включение/исключение активного сопротивления (1 или 0).
- **EL** включение/исключение индуктивности (1 или 0).
- **EC** включение/исключение ёмкости (1 или 0).
- **RO** значение сопротивления, Ом [ $\Omega$ ]
- **HL** значение индуктивности, Гн [H]
- **FC** значение ёмкости, мкФ [ $\mu$ F]

E\_1PVSRG\_CFG (AC, SPEC, TYP, VM, F, PH, R, L, RP\_C, P, Q, TC)

- **AC** признак источника переменного AC или постоянного DC тока (1 или 0 соответственно).
- **SPEC** ЭДС или напряжение источника (0 или 1 соответственно).
- **TYP** тип импеданса (для R = 1).
- **VM** действующее напряжение, кВ [kV].
- **F** частота, Гц [Hz].
- **PH** фазовый угол, ° [deg]
- **R** значение сопротивления, Ом [ $\Omega$ ]
- **L** значение индуктивности, Гн [H]
- **RP\_C** значение ёмкости, мкФ [mF]
- **P** выходная активная мощность
- **Q** выходная реактивная мощность
- **TC** время нарастания напряжения источника, с [s].

## БЛОК РЕЗЕРВИРОВАНИЯ ПАМЯТИ И ВЫПОЛНЕНИЯ СИМУЛЯЦИИ

Оператор распределения памяти #STORAGE используется для резервирования необходимой памяти под переменные и используется контроллером динамической памяти симулятора EMTDC. В подпрограмме EMTDC\_1PVSRG используются переменные различных типов: целые INTEGER, вещественные REAL и логические LOGICAL, а также формат числа типа RTCF для резервирования памяти под вещественные переменные, применяемые при симуляции.

Перед каждым вызовом подпрограммы EMTDC\_1PVSRG производится извлечение переменных из массива RTCF. Переменные, хранящиеся в этом массиве, определяются в блоке BEGIN при инициализации (в подпрограмме E\_1PVSRG\_CFG) и остаются неизменными. Таким образом, при извлечении этих данных из памяти обеспечивает их индивидуальность и гарантируется, что заданные значения являются уникальными для этого программного кода.

Программа завершается командой вызова подпрограммы EMTDC\_1PVSRG, кото-



рая выполняет управление источником в ходе симуляции:

EMTDC\_1 PVSRC (M, NBR, RT, AC, VP, W, A)

- **M** номер подсистемы, к которой относится данная ветвь
- **NBR** количество настроенных ветвей
- **RT** время нарастания напряжения источника, с [s].
- **AC** логический признак источника переменного тока AC.
- **VP** амплитудное значение напряжения, кВ [kV].
- **W** частота, рад/с [rad/s].
  - **A** фазовый угол, рад [rad].

## БЛОК ПРОВЕРКИ НА ОШИБКИ

Во избежание ошибочных результатов при симуляции правильным решением является использование проверки корректности данных, введенных пользователем. Для этого в закладке редактирования сценариев **Script** кликните по кнопке выбора разделов **Segment Manager** и в открывшемся окне выберите в списке пункт проверок **Checks** и добавьте его кликом по кнопке **Add->** в список активных пунктов на правое поле, затем завершите настройку кликом по **Ок**. В списке разделов, доступных для редактирования, появится пункт проверок **Checks** - выберите его. Создайте правило, по которому будет выводиться сообщение об ошибке при неверном значении сопротивления источника **R**, т.е. при значении меньшем или равном 0.

**Внимание.** Правило для обнаружения ошибки должно быть составлено таким образом, чтобы при невыполнении условия было сформировано сообщение об ошибке.

**ERROR** Source resistance must be greater than zero : R > 0.0

## 6. ТЕСТИРОВАНИЕ НОВОГО КОМПОНЕНТА

После создания нового компонента необходимо проверить его на функционирование. Для этого подключите к вновь созданному компоненту внешние элементы и задайте следующие параметры модели:

- Действующее напряжение (RMS) = 0,7071 кВ
- Частота = 50 Гц
- Время нарастания = 0,02 с
- Сопротивление источника = 1,0 Ом



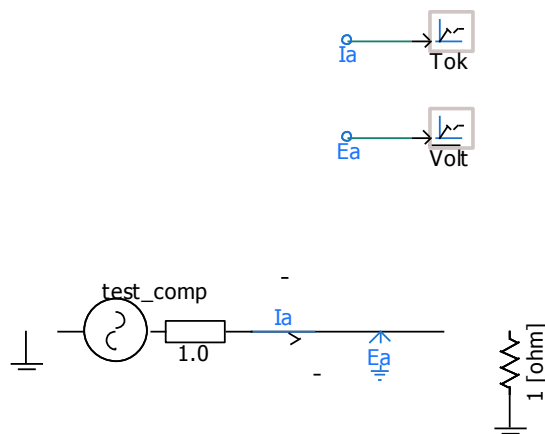
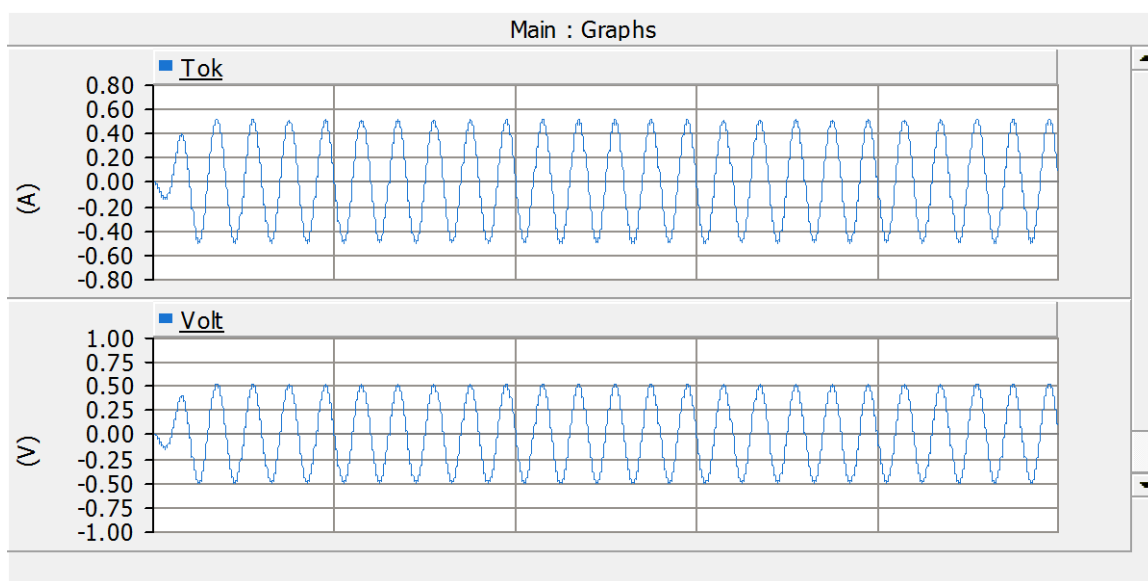


Рисунок. Модель для тестирования нового компонента

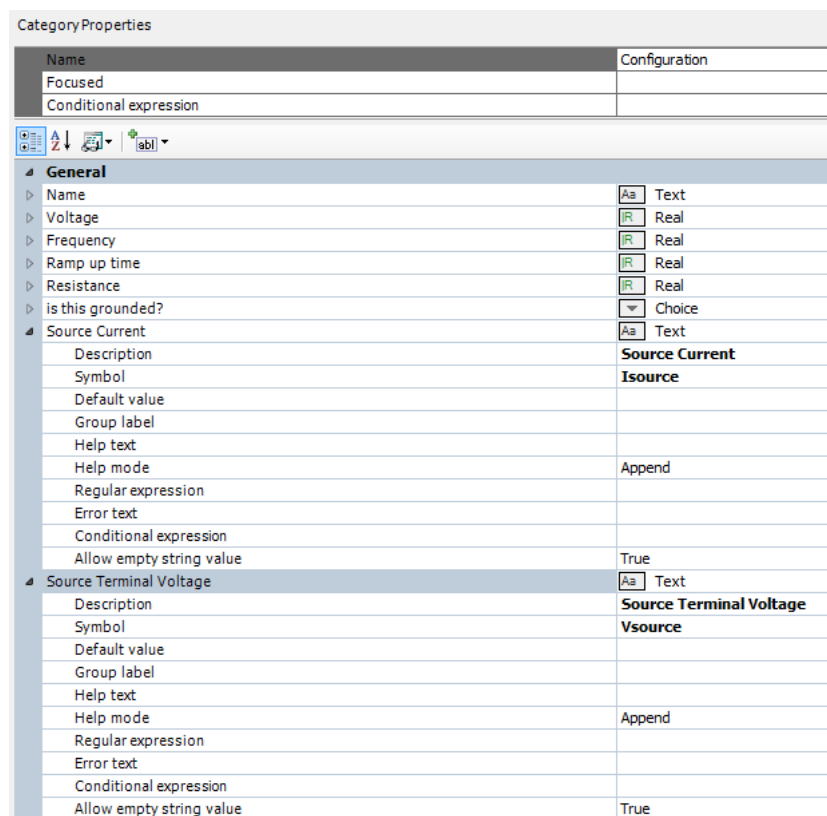


Графики результатов проверки нового компонента.

## 7. ИНТЕРФЕЙС ВНУТРЕННИХ ПЕРЕМЕННЫХ КОМПОНЕНТА

В PSCAD имеется возможность мониторинга значений внутренних переменных компонента для отображения их на графиках или использования в модели. Для этого используются механизм публикации внутренних переменных *internal output variable*. На примере ниже показана публикация внутренних переменных источника (напряжение и ток). Для этого в редакторе свойств компонента в разделе параметров **Parameters** добавьте два новых текстовых поля **Text**: первое – с наименованием *Isource* в поле обозначения **Symbol** и поясняющим текстом *Source Current (ток через источник)* в поле описания **Description**. Для второго – *Vsource* в поле **Symbol** и описанием *Source Terminal Voltage (напряжение на клеммах источника)*.





Перейдите к редактированию кода в закладку **Script** раздел **Fortran**. Используя директиву **#OUTPUT**, опубликуйте значение тока через источник (цепь **BRN**), посредством оператора **CBR**, который используется для измерения тока в цепи.

```
#OUTPUT REAL Isource {$CBR:BRN}
```

Для публикации напряжения между узлами портов **NA** и (**G** или **NB**) используется также директива **#OUTPUT** и оператор **VDC**. Так как используется компонент с задаваемой пользователем конфигурацией подключения к порту **NB** или земле **G**, то это следует учитывать при измерении напряжения и, соответственно, добавить условную компиляцию.

В итоге программный код Fortran должно выглядеть следующим образом:

```
#BEGIN
    CALL E_BRANCH_CFG($BRN,$SS,1,0,0,$R,0.0,0.0)
    CALL E_1PVSRC_CFG(1,0,1,$Vrms,$f,0.0,$R,0.0,0.0,0.0,0.0,$tr)
#ENDBEGIN
#STORAGE LOGICAL:1 INTEGER:6 REAL:8 RTCF:4
#LOCAL REAL RVD1_1
#LOCAL REAL RVD1_2
#LOCAL REAL RVD1_3
#LOCAL REAL RVD1_4
! Single Phase AC source: Type: R
    RVD1_1 = RTCF(NRTCF)
    RVD1_2 = RTCF(NRTCF+1)
    RVD1_3 = RTCF(NRTCF+2)
    RVD1_4 = RTCF(NRTCF+3)
    NRTCF = NRTCF + 4
```



```

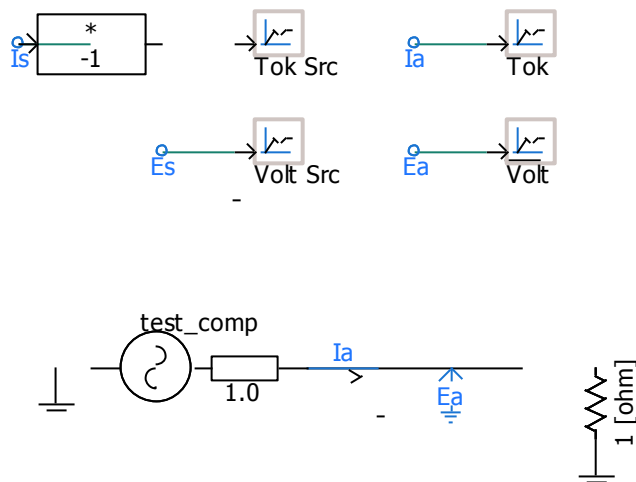
CALL EMTDC_1PVSRG ($SS, $BRN, RVD1_4, .TRUE., RVD1_1, RVD1_2, RVD1_3)
!
#OUTPUT REAL Isource {$CBR:BRN}
#IF gnd == 1
  #OUTPUT REAL Vsource {$VDC:NA:G}
#ELSE
  #OUTPUT REAL Vsource {$VDC:NA:NB}
#ENDIF

```

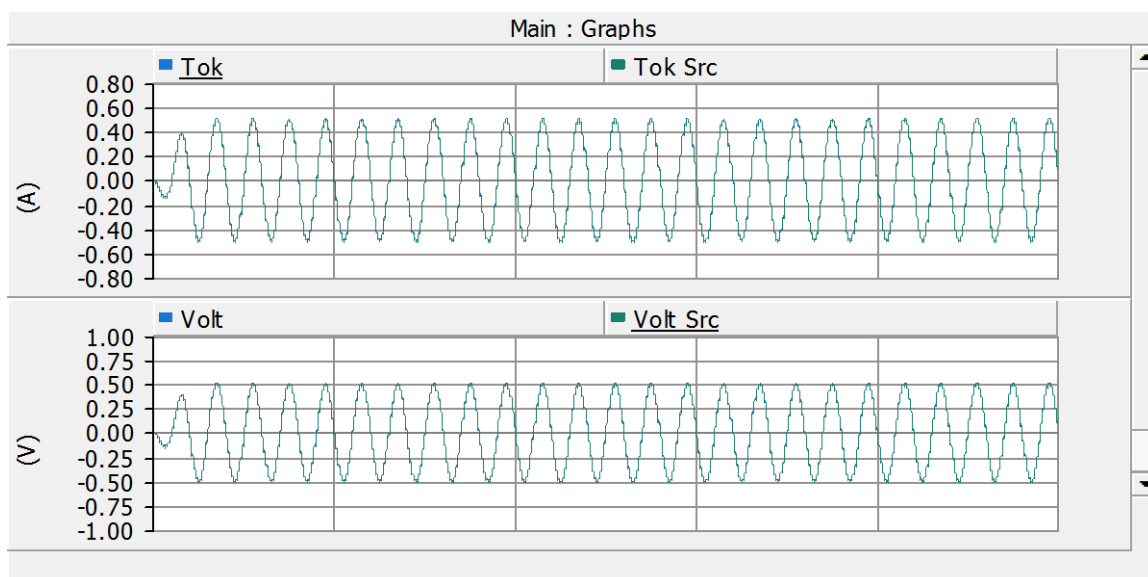
Параметры с наименованиями, приведенными выше (*Isource* и *Vsource*), будут выступать в качестве портов вывода значений внутренних переменных компонента. При построении модели у каждого приведенного параметра будет текстовое поле, в котором пользователь может задать уникальные для модели имена сигналов, и которые будут далее использованы при построении графиков или как входных для других элементов модели.

## 8. ЗАПУСК СИМУЛЯЦИИ С НОВЫМ КОМПОНЕНТОМ

Ранее созданную модель можно использовать для проверки нового компонента. Создайте два новых канала **Output Channel** для вывода мгновенных значений на графиках. В параметрах нового компонента присвойте наименование выходным сигналам компонента *Is* и *Es*. Добавьте новые каналы на графики уже на имеющиеся графики кривых, чтобы можно было провести сравнение. Для удобства сравнения силы тока внутри и снаружи источника используйте элемент коэффициента усиления **Gain** с коэффициентом  $-1$ . Итоговая схема и графики приведены ниже.







## 9. ВЫВОДЫ

Таким образом, достигнута поставленная цель по созданию собственного компонента, создание модели с его использованием и выполнена симуляция. Описанная методика и пример могут быть использованы при создании собственного компонента.

