



## 3. Tcl скрипирование

### Введение

Разработка и запуск Tcl скриптов для контроля над программой Altera Quartus II позволяет вам выполнять широкий набор функций, таких как компиляция проекта или написания процедур для автоматизации общих задач.

Вы можете использовать Tcl скрипты для управления проектом Quartus II, для создания назначений, для определения ограничений проекта, для создания назначений для чипа, для запуска компиляции, для выполнения временного анализа, для импорта назначений регионов LogicLock, для использования Quartus II редактора чипа и для доступа к отчётам. Вы можете автоматизировать ваши назначения, используя Tcl скрипты, в том случае, когда вы не желаете создавать их индивидуальными. Tcl скрипты также способствуют миграции проекта или назначений. Например, когда используется один прототип или плата разработки для различных проектов, вы можете автоматизировать назначения для размещения выводов в каждом новом проекте. Программа Quartus II может также генерировать Tcl скрипты, основываясь на текущих назначениях в проекте, что облегчает передачу назначений в другие проекты.

Команды Tcl в программе Quartus II соответствуют стандартам EDA индустрии Tcl программного интерфейса приложения (API) в части использования опций командной строки для задания аргументов. Т.е. Tcl команды просто выучить и использовать. Если вы сталкиваетесь с ошибкой в использовании аргумента команды, Tcl интерпретатор предоставляет вспомогательную информацию, показывающую корректное использование.

Эта глава содержит примеры Tcl скриптов для автоматизации программы Quartus II. Вы можете модифицировать эти примеры скриптов для использования в вашем проекте. Ещё больше примеров Tcl скриптов вы можете найти в секции "Примеры проектов" на части "Поддержка" на веб-сайте Altera.

Эта глава состоит из следующих секций:

- "Quartus II Tcl пакеты" на стр. 3–2
- "Quartus II Tcl API помощь" на стр. 3–3
- "Опции командной строки: -t, -s, и --tcl\_eval" на стр. 3–6
- "Сквозной процесс проектирования" на стр. 3–7
- "Создание проектов и назначений" на стр. 3–8
- "Компиляция проектов" на стр. 3–9
- "Отчёты" на стр. 3–10
- "Временной анализ" на стр. 3–12
- "Автоматическое исполнение скрипта" на стр. 3–12
- "Другие средства скрипта" на стр. 3–14
- "Использование оболочки Quartus II Tcl в интерактивном режиме" на стр. 3–19
- "Использование оболочки tclsh" на стр. 3–20

- “Основы Tcl скриптов” на стр. 3–20

### Что такое Tcl?

Tcl (выражаясь "просто") это популярный язык скриптов, очень похожий на многие оболочки скриптов и высокоуровневые языки программирования. Он предоставляет доступ к контролю над структурами, переменными, сетевыми адресами и API. Tcl – это язык скрипирования индустриального стандарта EDA, используемый в программах Synopsys, Mentor Graphics® и Altera. Он позволяет вам создавать собственные команды и работать равномерно на большинстве платформ разработки. За списком рекомендованной литературы по Tcl обратитесь к секции "Исходящие ссылки" на стр. 3-26.

Вы можете создавать собственные процедуры, написав скрипты, основанные на Tcl командах, определённых пользовательских процедурах и функциях Quartus II API. Вы можете этим автоматизировать ваш процесс разработки, запуская программу Quartus II в пакетном режиме, или исполняя отдельные Tcl команды интерактивно в интерактивной оболочке Quartus II Tcl.

Если вы незнакомы с Tcl скриптами, или только начинаете работать с Tcl скриптами, обратитесь к секции "Основы Tcl скриптов" на стр. 3-20 за вводной информацией.

Начиная с версии 4.1 программы Quartus II, поддерживаются Tcl/Tk версии 8.4, поставляемые Tcl DeveloperXchange на <http://tcl.activestate.com/>.

### Quartus II Tcl пакеты

Команды Quartus II Tcl группируются в пакеты по функциям. В табл. 3-1 описывается каждый Tcl пакет.

Табл. 3-1. Tcl пакеты (часть 1 из 2)

Имя пакета	Описание пакета
backannotate	Обратная аннотация назначений
chip_planner	Идентификация и модификация использования ресурсов в редакторе чипа
database_manager	Менеджер совместимых версий файлов баз данных
device	Получение информации о чипе и семействе от базы данных чипа
flow	Компиляция проекта, запуск исполняемых элементов командной строки и прочих процессов общего назначения
incremental_compilation	Управление разделами проекта и регионами LogicLock, а также настройками, связанными с инкрементной компиляцией
insystem_memory_edit	Чтение и редактирование содержимого внутренней памяти Altera
insystem_source_probe	Взаимодействие с исходниками и пробниками в системе в чипах Altera
jtag	Контролирование цепи JTAG
logic_analyzer_interface	Запрос и модификация состояния вывода интерфейса логического анализатора
misc	Выполнение разнообразных задач
project	Создание и управление проектами и их версиями, создание любых назначений проекта, включая временные ограничения
rapid_recompile	Управление средством быстрой перекомпиляции Quartus II
report	Получение информации из таблиц отчётов, создание собственных отчётов

Табл. 3-1. Tcl пакеты (часть 2 из 2)

rtl	Изучение и уточнение RTL списка соединений вашего проекта
sdc	Задание ограничений и исключений для временного анализатора TimeQuest
sdc_ext	Специфические SDC команды Altera
simulator	Конфигурирование и выполнение симуляции
sta	Содержит набор Tcl функций для получения расширенной информации из временного анализатора Quartus II TimeQuest
stp	Запуск логического анализатора SignalTap® II

По умолчанию загружается только минимальный набор пакетов для каждого исполняемого компонента Quartus II. Таким образом достигается минимально возможное использование памяти, необходимой для каждого исполняемого компонента. Поскольку загружается только минимальный набор пакетов, вы должны загрузить остальные пакеты, прежде чем запускать команды из этих пакетов.

Из-за того, что различные пакеты доступны в различных исполняемых компонентах, вы должны запустить ваши скрипты с исполняемыми компонентами, которые содержат пакеты, используемые вами в скриптах. Например, если вы используете команды в пакете **sdc\_ext**, вы должны использовать исполняемый компонент **quartus\_sta** для запуска скрипта, потому что исполняемый компонент **quartus\_sta** является единственным поддерживающим **timing** (временной) пакет.

Следующая команда выводит список пакетов, загружаемых или доступных для загрузки для исполняемого компонента в консоль:

```
<executable name> --tcl_eval help
```

Например, введите следующую команду, чтобы вывести список пакетов загружаемых или доступных для загрузки в исполняемом компоненте **quartus\_fit**:

```
quartus_fit --tcl_eval help
```

### Загрузка пакетов

Для загрузки Quartus II Tcl пакетов используйте команду *load\_package* следующим образом:

```
load_package [-version <version number>] <package name>
```

Эта команда похожа на пакет необходимой Tcl команды (описанный в табл. 3-2 на стр. 3-4), но при этом вы запросто сможете выбрать различные версии Quartus II Tcl пакетов с помощью команды *load\_package*.

За дополнительной информацией обратитесь к главе "[Скрипирование в командной строке](#)" в томе 2 настольной книги Quartus II.

### Quartus II Tcl API помощь

Доступ к разделу помощи Quartus II Tcl API открывается после ввода следующей команды в системной командной строке:

```
quartus_sh --qhhelp
```

Этой командой запускается командная строка Quartus II и браузер помощи Tcl API, в котором отображены все команды и опции Quartus II Tcl API. Он содержит подробное описание и примеры для каждой команды.

Quartus II Tcl API помощь предоставляет простой доступ к информации о командах Quartus II Tcl. Для доступа к разделу помощи введите *help* в строке Tcl, как показано в примере 3-1.

**Example 3-1. Help Output**

```
tcl> help
-----
-----
Available Quartus II Tcl Packages:
-----

Loaded                                Not Loaded
-----
::quartus::misc                       ::quartus::device
::quartus::old_api                    ::quartus::backannotate
::quartus::project                    ::quartus::flow
::quartus::timing_assignment          ::quartus::logiclock
::quartus::timing_report              ::quartus::report

* Type "help -tcl"
to get an overview on Quartus II Tcl usages.
```

Используйте опцию **-tcl**, как показано во введении в Quartus II Tcl API, чтобы сфокусироваться на способе получения помощи (краткой или расширенной) для Tcl команд и Tcl пакетов.

Раздел помощи Tcl API также доступен в он-лайн помощи Quartus II. Используйте поиск по имени команды или пакета для поиска подробностей их использования.

В таблице 3-2 обобщаются опции помощи, доступной в среде Tcl.

**Табл. 3-2.** Опции помощи, доступной в среде Quartus II Tcl (часть 1 из 2)

Команда помощи	Описание
help	Просмотр списка доступных Quartus II Tcl пакетов, загруженных и незагруженных.
help -tcl	Просмотр списка команд, используемых для загрузки Tcl пакетов и доступных в разделе помощи в командной строке.
help -pkg <b>&lt;package_name&gt;</b> [-version <b>&lt;version number&gt;</b> ]	Просмотр помощи для заданного пакета Quartus II, которая содержит список доступных Tcl команд. Для удобства, вы можете опускать префикс пакета <b>::quartus::</b> , и вводить <i>help -pkg &lt;package name&gt;</i> . Если вы не задали опцию <b>-version</b> , будет отображаться раздел помощи для текущего пакета по умолчанию. Если пакет, для которого вы хотите получить помощь, не загружен, по умолчанию загружается последняя версия пакета. Примеры: help -pkg ::quartus::project help -pkg project help -pkg project -version 1.0

Табл. 3-2. Опции помощи, доступной в среде Quartus II Tcl (часть 2 из 2)

Команда помощи	Описание
<b>&lt;command_name&gt;</b> -h или <b>&lt;command_name&gt;</b> -help	Просмотр короткой помощи для команды Quartus II Tcl, для которой загружен пакет. Примеры: project_open -h project_open -help
package require ::quartus:: <b>&lt;package name&gt;</b> [ <b>&lt;version&gt;</b> ]	Загрузка Quartus II Tcl пакета заданной версии. Если <b>&lt;version&gt;</b> не задана, по умолчанию загружается последняя версия. Пример: package require ::quartus::project 1.0 Эта команда похожа на команду <b>load_package</b> . Преимущество в использовании <b>load_package</b> – это то, что вы сможете свободно чередовать различные версии одного пакета. Введите <b>&lt;package name&gt;</b> [-version <i>&lt;version number&gt;</i> ] для загрузки Quartus II Tcl пакета с заданной версией. Если опция -version не задана, по умолчанию загружается последняя версия. Пример: load_package ::quartus::project 1.0
help -cmd <b>&lt;command_name&gt;</b> [-version <b>version&gt;</b> ] или <b>&lt;command_name&gt;</b> -long_help	Просмотр помощи для Quartus II Tcl команды. Для загрузки необходимого пакета достаточно ввести лишь: <b>&lt;command name&gt;</b> -long_help. Если опция -version не задана, по умолчанию загружается текущая версия пакета. Если пакет, для которого вы хотите получить помощь, не загружен, по умолчанию загружается последняя версия пакета. Примеры: project_open -long_help help -cmd project_open help -cmd project_open -version 1.0
help -examples	Просмотр примеров использования Quartus II Tcl.
help -quartus	Просмотр помощи в предопределённом глобальном Tcl массиве, который может быть использован для просмотра информации о текущих запущенных исполняемых компонентах Quartus II.
quartus_sh --qhelp	Для запуска Tk viewer для раздела помощи командной строки Quartus II и для отображения помощи для исполняемых компонентов командной строки и Tcl API пакетов. За дополнительной информацией об этой утилите обратитесь к главе " <a href="#">Скрипирование в командной строке</a> " в томе 2 настольной книги Quartus II.

**Опции командной строки: -t, -s, и --tcl\_eval**

В табл. 3-3 показаны три опции командной строки, которые вы можете использовать с исполняемыми компонентами, поддерживающими Tcl.

**Табл. 3-3.** Опции командной строки, поддерживающие Tcl скрипирование

Опция командной строки	Описание
-t <script file> [<script args>]	Запускает заданный Tcl скрипт с опциональными аргументами.
-s	Открывает исполняемый компонент в режиме интерактивной оболочки Tcl.
--tcl_eval <tcl command>	Определяет оставшиеся аргументы командной строки в качестве Tcl команд. Например, следующая команда отображает помощь для пакета проекта: quartus_sh --tcl_eval help -pkg project

**Запуск Tcl скрипта**

Запуск исполняемого компонента с опцией **-t** запускает соответствующий Tcl скрипт. Вы можете также задать аргументы для скрипта. Доступ к аргументам через переменную **argv** или с использованием пакета, такого как **cmdline**, поддерживается в следующей форме:

```
-<argument name> <argument value>
```

Пакет **cmdline** содержится в директории *<Quartus II directory>/common/tcl/packages*.

Например, для запуска скрипта с именем **myscript.tcl** с единственным аргументом **Stratix** введите следующую команду в системной командной строке:

```
quartus_sh -t myscript.tcl Stratix
```

Начиная с версии 4.1, программа Quartus II поддерживает переменную **argv**. В предыдущих версиях программы доступ к скриптам происходил через глобальную переменную **quartus(args)**.

Обратитесь к секции "Доступ к аргументам в командной строке" на стр. 3-17 за подробной информацией.

**Режим интерактивной оболочки**

Запуск исполняемого компонента с опцией **-s** запускает интерактивную оболочку Tcl. Например, для открытия исполняемого компонента временного анализатора Quartus II TimeQuest в режиме интерактивной оболочки введите следующую команду:

```
quartus_sta -s
```

Команды, которые вы вводите в оболочке Tcl интерпретируются, когда вы кликаете на **Enter**. Вы можете запустить скрипт в интерактивной оболочке с помощью следующей команды:

```
source <script name>
```

Если команда не распознаётся оболочкой, она считается внешней командой и выполняется с помощью команды **exec**.

**Определение в качестве Tcl**

Запуск исполняемого компонента с опцией `--tcl_eval` вызывает исполняемый компонент, определённый соответственно в оставшейся части аргументов командной строки, в качестве Tcl команд. Это полезно, когда вы хотите запустить простые Tcl команды, написанные на других языках скрипирования.

Например, следующая команда запускает Tcl команду, которая выводит команды, доступные в пакете проекта:

```
quartus_sh --tcl_eval help -pkg project
```

**Использование окна Quartus II Tcl консоли**

Вы можете запускать Tcl команды прямо в окне Quartus II Tcl консоли. В меню **View** кликните **Utility Windows**. По умолчанию, окно Tcl консоли пристыковано к правому нижнему углу в графической оболочке Quartus II. Всё введённое в Tcl консоли интерпретируется оболочкой Quartus II Tcl.

Окно Quartus II Tcl консоли поддерживает Tcl API в программе Quartus II версии 3.0 и раньше для поддержки совместимости со старыми проектами и устаревшими инструментами EDA.

Сообщения Tcl выводятся во вкладке **System** (окно сообщений). Ошибки и сообщения записываемые в stdout и stderr также отображаются в окне Quartus II Tcl консоли.

**Сквозной процесс проектирования**

Вы можете использовать Tcl скрипты для контроля над всеми аспектами процесса проектирования, включая контроль над другими программами, если оные содержат интерфейс скрипирования.

Обычно инструменты EDA содержат собственные интерпретаторы скрипта, которые расширяют основную функциональность языка специфическими командами инструментария. Например, интерпретатор Quartus II Tcl поддерживает все основные команды языка Tcl, и добавляет некоторое количество команд специально для программы Quartus II. Вы можете включить команды в один скрипт Tcl для запуска другого скрипта, который позволяет вам комбинировать или собирать вместе скрипты для контроля над различными инструментариями. Поскольку скрипты для различных инструментариев должны исполняться различными Tcl интерпретаторами, проблематично передать информацию между скриптами, за исключением момента, когда один скрипт пишет информацию в файл, а другой скрипт читает его.

Внутри программы Quartus II, вы можете выполнять несколько различных операций в процессе проектирования (таких как синтез, компоновка и временной анализ) в одном скрипте, упрощая получение информации о глобальном состоянии и сохраняя данные между операциями. Однако существуют ограничения на операции, которые вы можете выполнить в одном скрипте из-за различия в пакетах, поддерживаемых каждым исполняемым компонентом.

Не существует ограничений при запуске процесса от любого исполняемого компонента. Процессы содержат операции, найденные в секции Start в меню Processing в графической оболочке Quartus II, они также документируются с помощью Tcl команды `execute_flow`. Если вы создаёте настройки в программе Quartus II и запускаете процесс получения ожидаемого результата, вы можете сделать те же самые настройки и запустить тот же самый процесс в любом исполняемом компоненте командной строки.