 В следующей таблице приведены опции, обозначающие типы выводимых данных. Обратите внимание, что **они чувствительны к регистру** – 'c' означает совсем не то, что 'C'.

|  |  |
| --- | --- |
| **Опция** | **Значение, выводимое в лог** |
| **c** | **Категория сообщения**.  После символа категории в фигурных скобках может следовать указание – сколько частей имени категории выводить. Они отсчитываются с конца, что логично – это позволяет отсечь длинное имя пакета. Т.е., например, при имени категории*ru.skipy.logging.tests.Log4JTest* комбинация %c{3} приведет к выводу в лог *logging.tests.Log4JTest* (три части имени с конца). Если такого указания нет – имя выводится целиком.  У **org.apache.log4j.EnhancedPatternLayout** есть еще несколько вариантов сокращения имени. Отрицательное значение в скобках означает "убрать указанное количество частей сначала". Т.е. при имени категории *ru.skipy.logging.tests.Log4JTest*комбинация %c{-3} приведет к выводу в лог *Log4JTest* – три первых части убраны. Если убрать надо б**о**льшее количество частей, чем присутствует – будет выведено все имя целиком.  Еще один вариант сокращения – запись вида %c{1.2.3.}. Означает она, что от первой части остается одна буква, от второй – две, от третьей – три. На оставшиеся части распространяется последнее значение. Последняя часть имени выводится целиком. Т.е. из имени *ru.skipy.logging.tests.Log4JTest* форматом %c{1.2.1.} мы получим *r.sk.l.t.Log4JTest* – одна буква, две, далее опять одна. Можно задать еще и символ, которым будут замещаться убранные символы: %c{1\*.2#.1$} даст результат *r\*.sk#.l$.t$.Log4JTest*. При длинных именах категорий такой формат может оказаться удобным. |
| **C** | **Полное имя класса, в котором сгенерировано сообщение**  Не путайте это имя с именем категории. В разделе [Использование в программном коде](http://www.skipy.ru/useful/logging.html#log4j_prog) мы увидим, что это две большие разницы. Имя категории может не совпадать с именем класса.   После имени класса также может идти указание на то, сколько частей имени выводить – полностью аналогично опции '%c'.   **Важно!** Генерация имени класса – достаточно медленная процедура. Стоит ее избегать при наличии такой возможности. И уж точно не стоит оставлять эту опцию в настройках в режиме промышленной эксплуатации.  У **org.apache.log4j.EnhancedPatternLayout** есть все те же варианты сокращения имени класса, которые перечислены для категории. |
| **d** | **Дата и/или время**  Выводит в лог текущие дату и/или время. В фигурных скобках после данной опции указывается формат даты – либо шаблон java.text.SimpleDateFormat, либо один из предустановленных – DATE, ABSOLUTE или ISO8601. Сравните этот набор с таблицей [выше](http://www.skipy.ru/useful/logging.html#date_formats) – поддерживаются *не все* форматы, перечисленные для *TTCCLayout*. Если указания на шаблон нет, используется формат ISO8601.   Документация рекомендует использовать предопределенные форматы вместо собственных шаблонов – под них разработаны специальные классы для более оптимального форматирования, чем это делает java.text.SimpleDateFormat.  У **org.apache.log4j.EnhancedPatternLayout** есть еще возможность в фигурных скобках указать временную зону, например, {GMT+1}. В этом случае дата будет выводиться в указанной временной зоне, вне зависимости от установленной на сервере. |
| **F** | **Имя файла, в котором было сгенерировано сообщение**.  Не путайте с именем класса. В данном случае в лог выведется именно имя файла, в общем случае не совпадающее с именем класса, например, для любых внутренних классов.  **Важно!** Генерация имени класса – **крайне** медленная процедура. Стоит ее избегать при наличии любой возможности. И категорически не рекомендуется оставлять эту опцию в настройках в режиме промышленной эксплуатации. |
| **l** | **Полная информация о точке генерации сообщения**.  Содержит имя класса, имя метода, имя файла и строку, в которой было сгенерировано сообщение. Т.е. для нашего примера [выше](http://www.skipy.ru/useful/logging.html#log_ex1) (напомню, что вызов логгера производится в методе main класса ru.skipy.logging.tests.Log4JTest, в строке 19), в лог будет выведена строка ru.skipy.logging.tests.Log4JTest.main(Log4JTest.java:19).  **Важно!** Фактически эта опция является аналогом следующей конструкции: %C.%M(%F:%L). Генерация **каждой** из частей в этом наборе – **крайне**медленная процедура. Ну и вся комбинация, естественно, быстрой не будет. Поэтому опции %lнеобходимо категорически избегать в режиме промышленной эксплуатации. В то же время в процессе отладки она может оказать неоценимую помощь. |
| **L** | **Номер строки, в которой было сгенерировано сообщение**.  Имеется в виду номер строки в файле. В принципе информация полезная, хотя часто можно обойтись и без нее.  **Важно!** Генерация номера строки – **крайне** медленная процедура. Стоит ее избегать при наличии любой возможности. И категорически не рекомендуется оставлять эту опцию в настройках в режиме промышленной эксплуатации. |
| **m** | **Сообщение**  То самое сообщение, которое передается в метод логгера. Ради чего, в основном, всё и затевается. |
| **M** | **Имя метода, в котором было сгенерировано сообщение**.  **Важно!** Генерация имени метода – **крайне** медленная процедура. Стоит ее избегать при наличии любой возможности. И категорически не рекомендуется оставлять эту опцию в настройках в режиме промышленной эксплуатации. |
| **n** | **Перевод строки**  Переводит в логе строку. Это необходимо, иначе все сообщения будут писаться в одну строку.  Для чего нужна такая опция – конец строки, вообще-то, платформозависимый. Под Windows это '\r\n' (CRLF), под \*NIX – '\n' (LF), на Mac-ах до недавнего времени был '\r' (CR), в MacOS X, возможно, стал как в \*NIX. И стандартная для одной системы комбинация может быть неправильно интерпретирована в другой.  В общем, для того, чтобы нормально читать логи в той операционной системе, где они ведутся, рекомендуется для перевода строки использовать именно опцию %n. |
| **p** | **Приоритет сообщения**.  Выводит [уровень логирования](http://www.skipy.ru/useful/logging.html#levels) для сообщения. |
| **r** | **Количество миллисекунд с момента инициализации системы логирования**.  Аналог [формата даты](http://www.skipy.ru/useful/logging.html#date_formats) RELATIVE компоновщика *TTCCLayout*. Может использоваться вместо даты, если есть такая необходимость. |
| **t** | **Имя потока**.  Выводит имя потока, в котором сгенерировано сообщение. Эта информация бывает весьма полезна, особенно если не лениться и осмысленно именовать все порождаемые потоки. |
| **x** | **Вложенный диагностический контекст (NDC)**  Выводит связанный с текущим потоком [вложенный диагностический контекст](http://www.skipy.ru/useful/logging.html#log4j_ndc). |
| **X** | **Ассоциативный диагностический контекст (MDC)**.  Выводит связанный с текущим потоком [ассоциативный диагностический контекст](http://www.skipy.ru/useful/logging.html#log4j_mdc). После опции в фигурных скобках **должно** идти имя ключа, по которому выбирается значение из контекста: %X{username} – вывод из контекста имени пользователя, если оно там есть.  У **org.apache.log4j.EnhancedPatternLayout** имя ключа необязательно. Если его нет – выводится всё содержимое контекста. |
| **%** | **Знак процента**.  Поскольку знак '%' является частью формата, а необходимость в его выводе периодичеси присутстует, конструкция '%%' выводит в лог знак '%'. |
| **properties** | **Свойства, связанные с сообщением**.  Эта опция специфична только для **org.apache.log4j.EnhancedPatternLayout**, у org.apache.log4j.PatternLayout ее нет. Выводит свойство по имени, указанному в фигурных скобках после опции. Если имя не указано – выводятся все свойства.  Честно сказать, разницы между этой опцией и %X я не нашел. Более того, по коду свойства сообщения трактуются именно как MDC – название с использованием этого префикса, копирование из него и т.п. В общем, можно использовать то, что нравится. |
| **throwable** | **Информация об исключении**.  Эта опция специфична только для **org.apache.log4j.EnhancedPatternLayout**, у org.apache.log4j.PatternLayout ее нет. Она позволяет вывести информацию об исключении, если оно было передано в метод логгера. В фигурных скобках указывается количество строк, которые надо выводить. %throwable{1} или %throwable{short} выведет одну строку – как правило, это имя класса исключения и текстовое сообщение. %throwable{0} или %throwable{none} вообще подавит вывод информации об исключении. Указание положительного числа означает количество строк, которое надо оставить с начала стека сообщения об ошибке, отрицательное – сколько убрать с конца. Если фигурные скобки отсутствуют – выводится полный стек. Точно так же логгер поступает и при отсутствии этой опции вообще. |

**С учетом баланса между требованиями производительности и объемом информации, которого достаточно для анализа логов, в промышленном режиме рекомендовано использование следующих опций: %c, %d, %m, %n, %p, %t, %x, %X, %throwable, %%. Остальные – %C, %F, %** Данный компоновщик поддерживает, кроме всего прочего, позиционное форматирование. Означает оно, что под каждую опцию можно выделить некоторое место – задать минимальный и максимальный размер значения, а также выравнивание, если значение меньше минимальной выделенной области. Модификаторы форматирования задаются между символом '%' и опцией. На примере опции %c рассмотрим действие модификаторов:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| **Модификатор** | **Выравнивание** | **Минимальная ширина** | **Максимальная ширина** | **Действие** |
| **%10с** | вправо | 10 | нет | Отводит минимум 10 символов под имя категории, если длина значения меньше – выравнивает его по правому краю поля |
| **%-10с** | влево | 10 | нет | Отводит минимум 10 символов под имя категории, если длина значения меньше – выравнивает его по левому краю поля |
| **%.20с** | нет | нет | 20 | Отводит максимум 20 символов под имя категории, если длина значения больше – обрезает с начала, оставляя указаное количество символов. Поскольку длина значения не может быть меньше предопределенной, о выравнивании говорить не приходится. |
| **%10.20с** | вправо | 10 | 20 | Отводит минимум 10 и максимум 20 символов под имя категории, если длина значения меньше – выравнивает его по правому краю поля, если больше – обрезает с начала, оставляя 20 символов. |
| **%-10.20c** | влево | 10 | 20 | Отводит минимум 10 и максимум 20 символов под имя категории, если длина значения меньше – выравнивает его по левому краю поля, если больше – обрезает с начала, оставляя 20 символов. |

**l, %L, %M – способны вызвать сильное падение производительности**.