

Network Working Group
Ed.
Request for Comments: 5322
Incorporated
Obsoletes: 2822
2008
Updates: 4021
Category: Standards Track

P. Resnick,
Qualcomm
October

Формат сообщений Internet

Internet Message Format

Статус документа

Это документ содержит проект стандарта протокола Internet для сообщества Internet и служит приглашением к дискуссии в целях развития и совершенствования протокола. Текущее состояние стандартизации и статус протокола можно узнать из текущей версии документа Internet Official Protocol Standards (STD 1). Допускается свободное распространение документа.

Аннотация

Этот документ задаёт формат сообщений Internet (IMF¹) - синтаксис текстовых сообщений, передаваемых между пользователями компьютеров через систему электронной почты. Данная спецификация является пересмотром RFC 2822, который, в свою очередь, пересматривает RFC 822, Standard for the Format of ARPA Internet Text Messages², с учётом опыта использования и накопленных изменений, отраженных в других RFC.

Оглавление

1. Введение.....	2
1.1. Рамки документа.....	2
1.2. Система обозначений.....	2
1.2.1. Обозначение уровня требований.....	2
1.2.2. Синтаксические обозначения.....	3
1.2.3. Структура документа.....	3
2. Лексический анализ сообщений.....	3
2.1. Общее описание.....	3
2.1.1. Предельные размеры строк.....	3
2.2. Поля заголовков.....	4
2.2.1. Бесструктурные поля заголовков.....	4
2.2.2. Структурированные поля заголовков.....	4
2.2.3. Длинные поля заголовков.....	4
2.3. Тело письма.....	4
3. Синтаксис.....	4
3.1. Введение.....	4
3.2. Лексемы.....	5
3.2.1. Квотирование символов.....	5
3.2.2. Пробелы для фальцовки и комментарии.....	5
3.2.3. Атом.....	5
3.2.4. Строки в кавычках.....	6
3.2.5. Прочие лексемы.....	6
3.3. Задание даты и времен.....	6
3.4. Задание адреса.....	7
3.4.1. Задание addr-спес.....	7
3.5. Общий синтаксис сообщений.....	8
3.6. Определения полей.....	8
3.6.1. Поле даты создания.....	9
3.6.2. Поля источника.....	9
3.6.3. Поля адресов получателей.....	9
3.6.4. Поля идентификации.....	10
3.6.5. Информационные поля.....	11
3.6.6. Поля пересылки.....	11
3.6.7. Поля трассировки.....	12
3.6.8. Дополнительные поля.....	12
4. Устаревший синтаксис.....	12
4.1. Прочие устаревшие маркеры.....	13
4.2. Устаревшие пробелы для фальцовки.....	13
4.3. Устаревшие форматы даты и времени.....	13

¹Internet Message Format.

²Стандарт для формата текстовых сообщений ARPA Internet.

4.4. Устаревшая адресация.....	14
4.5. Устаревшие поля заголовков.....	14
4.5.1. Устаревшее поле даты создания.....	14
4.5.2. Устаревшие поля отправителя.....	15
4.5.3. Устаревшие поля адресов получателей.....	15
4.5.4. Устаревшие поля идентификации.....	15
4.5.5. Устаревшие информационные поля.....	15
4.5.6. Устаревшие поля пересылки.....	15
4.5.7. Устаревшие поля трассировки.....	15
4.5.8. Устаревшие дополнительные поля.....	15
5. Вопросы безопасности.....	15
6. Взаимодействие с IANA.....	16
Приложение А. Примеры сообщений.....	17
Приложение А.1. Примеры адресации.....	17
Приложение А.1.1. Сообщение от одного лица другому с простой адресацией.....	18
Приложение А.1.2. Различные типы почтовых ящиков.....	18
Приложение А.1.3. Групповые адреса.....	18
Приложение А.2. Ответные сообщения.....	18
Приложение А.3. Пересылка сообщений.....	19
Приложение А.4. Сообщения с полями трассировки.....	19
Приложение А.5. Пробелы, комментарии и другие странности.....	20
Приложение А.6. Устаревшие формы.....	20
Приложение А.6.1. Устаревшая адресация.....	20
Приложение А.6.2. Устаревшие даты.....	20
Приложение А.6.3. Устаревшие пробелы и комментарии.....	20
Приложение В. Отличия от ранних спецификаций.....	21
Приложение С. Благодарности.....	22
7. Литература.....	23
7.1. Нормативные документы.....	23
7.2. Дополнительные ссылки.....	23

1. Введение

1.1. Рамки документа

Этот документ задает формат сообщений Internet (IMF) - синтаксис текстовых сообщений, передаваемых между пользователями компьютеров через систему электронной почты. Данная спецификация является пересмотром RFC 2822 [RFC2822], который, в свою очередь, пересматривает RFC 822, Standard for the Format of ARPA Internet Text Messages [RFC0822], с учетом опыта использования и накопленных изменений, отраженных в других RFC (таких, как [RFC1123]).

В этом документе задан синтаксис только для текстовых сообщений. В частности, не рассматриваются вопросы передачи изображений, звука и иной структурированной информации в сообщениях электронной почты. Опубликовано несколько расширений (таких, как серия документов MIME - RFC2045, [RFC2046], [RFC2049]), описывающих механизмы передачи таких данных в сообщениях электронной почты за счет расширения описанного здесь синтаксиса или за счет структурирования сообщений в соответствии с описанным синтаксисом. Такие механизмы выходят за пределы настоящей спецификации.

В контексте электронной почты сообщения представляются состоящими из конверта и тела (содержимого) письма. Конверт содержит информацию, требуемую для передачи и доставки сообщения (см. [RFC5321]). Тело письма является объектом, который доставляется адресату. Данная спецификация применима только к формату и части семантики содержимого писем. Документ не включает спецификации данных, включаемых в конверт сообщения.

Однако некоторые системы обработки сообщений могут использовать информацию из тела письма для создания конверта. Одной из задач настоящей спецификации является упрощение извлечения такой информации программными средствами.

Назначением этой спецификации является задание формата содержимого писем, передаваемых между системами. Хотя некоторые системы локально хранят сообщения в описанном здесь формате (это избавляет от необходимости преобразования форматов), а другие используют свои форматы хранения, вопросы локального хранения сообщений выходят за пределы данной спецификации.

Примечание. В задачи этой спецификации не входит задание внутренних форматов, используемых сайтами, конкретных характеристик систем обработки сообщений или пользовательского интерфейса программ для создания и просмотра сообщений. Кроме того, данный документ не задает кодирования символов для транспортировки или хранения почты и не задает число и битов в представлении символов и способы передачи или хранения этих битов.

1.2. Система обозначений

1.2.1. Обозначение уровня требований

В этом документе используются термины, выделенные **полужирным** шрифтом. Ключевые слова **необходимо** (MUST), **недопустимо** (MUST NOT), **требуется** (REQUIRED), **нужно** (SHALL), **не нужно** (SHALL NOT), **следует** (SHOULD), **не следует** (SHOULD NOT), **рекомендуется** (RECOMMENDED), **возможно** (MAY), **необязательно** (OPTIONAL) в данном документе используются для обозначения уровня требований и интерпретируются в соответствии с [RFC2119].

1.2.2. Синтаксические обозначения

В этой спецификации используется нотация ABNF¹ [RFC5234] для формального определения синтаксиса сообщений. Символы задаются десятичным кодом (например, значение %d65 представляет латинский символ A, %d97 - а) или регистронезависимыми литералами, заключенными в кавычки (например, "A" для представления A или a).

1.2.3. Структура документа

Документ делится на несколько частей.

В первом разделе содержится вводная часть документа.

В разделе 2 приведено общее описание сообщения и его компонент. Этот обзор поможет понять некоторые общие принципы, используемые в оставшейся части документа. Приведенные в этом разделе примеры **недопустимо** трактовать, как формальный синтаксис каких-либо компонент сообщения.

В разделе 3 даны формальные правила ABNF для структуры каждой части сообщения (синтаксис) и описаны связи между этими частями, а также их трактовка в контексте сообщения (семантика). Т. е., в этом разделе дается структурная схема каждой части сообщения (синтаксис), а также описание компонент и инструкции по их интерпретации (семантика). Раздел включает анализ семантики и синтаксиса субкомпонент сообщения, имеющих свою структуру. Рассмотренный в разделе 3 синтаксис представляет сообщения в том виде, в котором они **должны** создаваться. В этом разделе также указано, какие опции синтаксиса **следует** использовать.

Разделы 2 и 3 описывают сообщения, соответствующие данной спецификации.

В разделе 4 рассматривается «устаревший» синтаксис, на элементы которого приведены ссылки в разделе 3. Правила устаревшего синтаксиса являются элементами, которые были включены в ранние версии этой спецификации или широко использовались в почте Internet. Поэтому такие элементы **должны** интерпретироваться средствами разбора сообщений (parser) для обеспечения соответствия настоящей спецификации. Однако, поскольку элементы такого синтаксиса были сочтены неинтероперабельными или вызывающими проблемы у получателей сообщений, использование таких элементов при создании сообщений **недопустимо**.

В разделе 5 рассматриваются вопросы безопасности, которые следует принимать во внимание при реализации данной спецификации.

Приложение A содержит примеры сообщения. Приведенный набор примеров не является исчерпывающим списком всех появляющихся в Internet типов сообщений, но может служить иллюстрацией к некоторым синтаксическим формам.

В Приложении B перечислены отличия данной спецификации от более ранних версий спецификации формата сообщений Internet.

В Приложении C содержатся благодарности.

2. Лексический анализ сообщений

2.1. Общее описание

На базовом уровне сообщение представляет собой последовательность символов. Сообщения, соответствующие данной спецификации, включают символы с десятичными кодами от 1 до 127, интерпретируемые в соответствии с кодировкой US-ASCII² [ANSI.X3-4.1986]. Для краткости в этом документе такие символы будут называться просто «символами US-ASCII».

Символы сообщения делятся на строки. Строка представляет собой последовательность символов, завершающуюся кодами возврата каретки и перевода строки - в конце строки помещается символ возврата каретки (CR, десятичный код ASCII — 13), непосредственно за которым следует символ перевода строки (LF, десятичный код ASCII - 10). В данном документе такая последовательность обозначается «CRLF».

Сообщение состоит из полей заголовков³ (совокупность этих полей называют разделом заголовков сообщения), за которыми может следовать тело сообщения. Раздел заголовков представляет собой последовательность символьных строк, синтаксис которых описан в данной спецификации. Тело сообщения представляет собой последовательность символов, которая следует после раздела заголовков и отделена от него пустой строкой (строкой, содержащей только CRLF).

2.1.1. Предельные размеры строк

Данная спецификация вносит два ограничения на число символов в строке. Строка **должна** содержать не более 988 символов; **следует** использовать строки размером не более 78, без учета CRLF.

Ограничение в 988 обусловлено возможностями множества реализаций в части передачи, приема и хранения, которые не позволяют работать с сообщениями IMF, содержащими строки размером более 998 символов. Системы приема могут из соображений отказоустойчивости отказываться от ограничения на размер строк. Однако существует множество реализаций, которые (в соответствии с транспортными требованиями [RFC5321]) не принимают сообщения со строками размером более 1000 (включая CR и LF в конце строки), - разработчикам важно принимать этот факт во внимание.

Рекомендация использовать строки размером не более 78 обусловлена параметрами пользовательского интерфейса многих реализаций, которые могут отсекают лишние символы или неаккуратно переносить слова при наличии в строке

¹Augmented Backus-Naur Form — расширенная форма Бэкуса-Наура.

²В этом документе сказано, что сообщения создаются с использованием только символов US-ASCII с кодами от 1 до 127. Существуют другие документы (в частности, серия документов MIME [RFC2045], [RFC2046], [RFC2047], [RFC2049], [RFC4288], [RFC4289]), расширяющие данную спецификацию в части диапазона используемых в сообщениях символов. Обсуждение таких механизмов расширения выходит за пределы данной спецификации.

³В разговорной практике и ранних версиях спецификации термин «заголовок» используется для обозначения всего раздела заголовков или отдельных полей заголовков. Во избежание путаницы в этом документе для обозначения отдельных полей используется термин «поле заголовка», а для обозначения заголовка сообщения в целом служит термин «раздел заголовков».

более 78 символов, несмотря на то, что такие реализации не соответствуют требованиям данной спецификации (и [RFC5321], если приводят к потере символов. Однако наличие такого ограничения не запрещает реализациям корректно отображать сообщения со строками произвольной длины (по крайней мере, строками, содержащими не более 998 символов) для повышения уровня отказоустойчивости.

2.2. Поля заголовков

Поля заголовков представляют собой строки, начинающиеся с имени поля, за которым следует двоеточие (":"), содержимое поля и знак завершения строки CRLF. Имя поля **должно** состоять только из печатаемых символов US-ASCII (т. е., символов с кодами от 33 до 126, включительно), исключая двоеточие. Значение поля может включать печатаемые символы US-ASCII, символы пробела (SP, код ASCII - 32) и горизонтальной табуляции (HTAB, код ASCII — 9), которые вместе называют также пробельными символами¹. В значение поля **недопустимо** включать символы CR и LF, за исключением их использования в «фальцованных» и «нефальцованных» полях, как описано в параграфе 2.2.3. Значения полей **должны** соответствовать синтаксису, описанному в разделах 3 и 4 настоящей спецификации.

2.2.1. Бесструктурные поля заголовков

Некоторые поля заголовков в этой спецификации определены просто как неструктурированные (в параграфе 3.2.5 указано, что эти поля содержат произвольный набор печатаемых символов US-ASCII и пробельных символов) без дополнительных ограничений. Такие поля будем называть бесструктурными. Семантически бесструктурное поле трактуется просто как строка символов без дополнительной обработки (за исключением фальцовки, описанной в параграфе 2.2.3).

2.2.2. Структурированные поля заголовков

Синтаксис некоторых полей в данной спецификации вносит дополнительные ограничения по сравнению с описанными выше бесструктурными полями. Такие поля называются структурированными. Структурированное поле представляет собой последовательность лексем, описанных в разделах 3 и 4 данной спецификации. Многие из таких лексем (в соответствии с их синтаксисом) допускают включение комментариев в начале или в конце лексемы (см. параграф 3.2.2), а также пробельных символов, которые могут использоваться для фальцовки, как описано в параграфе 2.2.3. Семантический анализ структурированных полей приводится вместе с описанием их синтаксиса.

2.2.3. Длинные поля заголовков

Каждое поле заголовка логически представляет собой строку символов, состоящую из имени поля, двоеточия и тела (значения) поля. Однако для удобства и с учетом ограничения размеров строки (998/78 символов), значение поля может быть разбито на несколько строк; это называется «фальцовкой» (folding). Общим правилом заключается в том, что данная спецификация разрешает включение последовательности CRLF (новая строка) перед любыми пробельными символами.

Например, поле заголовка

```
Subject: This is a test
```

можно записать в форме

```
Subject: This  
is a test
```

Note: Хотя структурированные поля определены таким образом, что фальцовка может выполняться между множеством лексем (и даже внутри некоторых лексем), фальцовку **следует** ограничивать включением CRLF на синтаксических границах верхних уровней. Например, если тело поля определено, как разделенные запятыми значения, рекомендуется выполнять фальцовку после запятой, разделяющей элементы структуры, а не в других местах (даже если это разрешено).

Процесс преобразования фальцованного многострочного представления поля в обычное однострочное называется расфальцовкой (unfolding) и выполняется путем простого удаления всех последовательностей CRLF, непосредственно за которыми следуют пробельные символы (WSP). Каждое поле заголовка для дальнейшего синтаксического и семантического анализа следует трактовать в его нефальцованном представлении. На нефальцованные поля заголовков не накладываются ограничения по размеру и они, следовательно, могут иметь любую длину.

2.3. Тело письма

Тело сообщения представляет собой простые строки символов US-ASCII. Для содержимого сообщения существует только два типа ограничений²:

- символы CR и LF **должны** использоваться только совместно, как CRLF; **недопустимо** использование этих символов в теле сообщения по-отдельности;
- строки символов **должны** быть не длиннее 998 символов, **следует** ограничивать размер строк 78 символами (без учета CRLF).

3. Синтаксис

3.1. Введение

Описанный в этом разделе синтаксис задает корректный формат почтовых сообщений Internet. Сообщения, соответствующие данной спецификации, **должны** использовать описанный в этом разделе синтаксис. При наличии опций в этом разделе указано (в тексте или примечании), когда **следует** генерировать одну опцию.

¹White space characters - WSP.

²Как указано выше, существует ряд документов (в частности, серия MIME [RFC2045], [RFC2046], [RFC2049], [RFC4288], [RFC4289]), расширяющих (и ограничивающих) данную спецификацию в части содержимого сообщений. Отметим еще раз, что такие механизмы выходят за пределы спецификации.

Для определяемых выражений дается краткое описание синтаксиса и применения, после чего приводится синтаксис в формате ABNF и семантический анализ. Часть примитивов, используемых в документе, не определена здесь и заимствована из основных правил, приведенных в Приложении В.1 документа [RFC5234]. К таким примитивам относятся: CR, LF, CRLF, HTAB, SP, WSP, DQUOTE, DIGIT, ALPHA и VCHAR.

В некоторых определениях будут встречаться элементы, имена которых начинаются с префикса "obs-". Такие элементы обозначают лексемы, относящиеся к устаревшему синтаксису и описанные в разделе 4. Во всех случаях такие конструкции при генерации корректных почтовых сообщений Internet игнорируются и их **недопустимо** использовать в качестве компонент сообщения. Однако при интерпретации сообщений такие лексемы **должны** рассматриваться как синтаксически корректные. В этом смысле раздел 3 описывает элементы "obs-", как игнорируемые, а раздел 4 добавляет описание интерпретации сообщений с такими элементами.

3.2. Лексемы

В этом разделе описаны правила, используемые для определения базового лексического анализатора, который представляет лексемы анализаторам верхнего уровня. Раздел определяет лексемы, используемые в структурированных полях заголовков¹.

3.2.1. Квотирование символов

Некоторые символы имеют специальное значение (например, используются в качестве границ лексем). Для использования таких символов в общепринятом смысле служит механизм квотирования (добавления «кавычек»).

```
quoted-pair = ("\\" (VCHAR / WSP)) / obs-qp
```

При появлении любой пары с квотированием (quoted-pair) она интерпретируется как отдельный символ. Т. е., символ \², являющийся частью пары с квотированием, становится семантически «невидимым».

3.2.2. Пробелы для фальцовки и комментарии

Пробельные символы, включая и те, которые служат для фальцовки (см. параграф 2.2.3), могут появляться между разными элементами в теле полей заголовков. Строки символов, трактуемые, как комментарии, также могут включаться в тело структурированных полей заголовков с заключением их в скобки. Ниже определяется конструкция пробелов для фальцовки (FWS³) и комментариев.

Строки символов, заключенные в скобки, рассматриваются, как комментарии, если они не являются частью строки с квотированием, определенной в параграфе 3.2.4. Комментарии могут быть вложенными.

В этой спецификации есть несколько мест, где комментарии и FWS могут вставляться свободно. Для согласования с таким синтаксисом определена дополнительная лексема CFWS, показывающая возможность включения комментариев и/или FWS. Однако в тех случаях, где в данной спецификации разрешается CFWS, **недопустимо** использовать их так, чтобы фальцованное поле заголовка включало только символы WSP и ничего другого.

```
FWS          = ([*WSP CRLF] 1*WSP) / obs-FWS
              ; пробельные символы для фальцовки
ctext        = %d33-39 /
              ; печатаемые символы US-ASCII,
              %d42-91 /
              ; не включая
              %d93-126 /
              ; "(", ")", " " и "\"
              obs-ctext
ccontent     = ctext / quoted-pair / comment
comment      = "(" *([FWS] ccontent) [FWS] ")"
CFWS         = (1*([FWS] comment) [FWS]) / FWS
```

В этой спецификации появление FWS (пробельные символы для фальцовки) означает указание места, где возможно выполнение фальцовки, как описано в параграфе 2.2.3. Всякий раз при использовании фальцовки (т. е., поля заголовка, содержащего последовательность CRLF, за которой следует любой символ WSP) в сообщении должна выполняться расфальцовка (удаление CRLF) до любого семантического анализа, выполняемого по отношению к заголовку в соответствии с данной спецификацией. Т. е., любые последовательности CRLF, включенные в FWS, являются семантически невидимыми.

Комментарии обычно используются в теле структурированных полей для обеспечения некой дополнительной информации для человека. Поскольку в комментариях могут содержаться символы FWS, это позволяет выполнять фальцовку внутри комментария. Отметим также, что возможность использования в комментариях пар с квотированием, позволяет включать в комментарии скобки и символы обратной дробной черты (\), если они задаются в форме пары с квотированием. Семантически внешние скобки не являются частью комментария — комментарием является то, что заключено в эти скобки. Как было отмечено выше, символы «\» в парах с квотированием и последовательности CRLF внутри FWS внутри комментариев являются семантически невидимыми и, следовательно, не являются частью комментария.

FWS в качестве комментария (CFWS) между лексемами в структурированном поле заголовка семантически интерпретируется как один символ пробела.

3.2.3. Атом

Некоторые конструкции в теле структурированных полей заголовков представляют собой просто строки некоторых базовых символов. Такие конструкции называют атомами.

¹Читателям следует обратить особое внимание на способы использования этих лексем в синтаксисе верхнего и нижнего уровня в оставшейся части документа. В частности, лексемы пробельных символов и комментариев, определенные в параграфе 3.2.2, будут использоваться в определенных здесь лексемах нижнего уровня, которые, в свою очередь, используются в определенных ниже лексемах верхнего уровня. Следовательно, пробельные символы и комментарии могут включаться в лексемы верхнего уровня, даже в тех случаях, когда они явно не указаны в конкретном определении.

²Символ \ может появляться в сообщении и как обычный символ (не элемент пары с квотированием). В этом случае символ \ не является семантически невидимым. В данной спецификации символ \, как таковой появляется только в описаниях ccontent, qcontent и obs-dtext.

³Folding white space.

В некоторых структурированных полях заголовков допускается включение точки («.»), код ASCII - 46) в atext. Для таких конструкций определена дополнительная лексема «атом с точкой» (dot-atom).

```

atext      = ALPHA / DIGIT /           ; Печатаемые символы US-ASCII,
          "!" / "#" /                 ; не включая специальных символов.
          "$" / "%" /                 ; Используются для атомов.
          "&" / "'" /
          "*" / "+" /
          "-" / "/" /
          "=" / "?" /
          "^" / "_" /
          "`" / "{" /
          "|" / "}" /
          "~"

atom       = [CFWS] 1*atext [CFWS]
dot-atom-text = 1*atext *("." 1*atext)
dot-atom   = [CFWS] dot-atom-text [CFWS]
specials   = "(" / ")" /             ; Специальные символы1, которые не
          "<" / ">" /                 ; появляются в atext
          "[" / "]" /
          ":" / ";" /
          "@" / "\" /
          ", " / "." /
DQUOTE

```

Лексемы atom и dot-atom интерпретируются, как единый элемент, включающий строку символов. Семантически дополнительные комментарии и FWS, окружающие остальные символы, не являются частью атома — атом представляет собой только символы atext (или atext и «.» для dot-atom).

3.2.4. Строки в кавычках

Строки, включающие символы, недопустимые для использования в атомах, могут быть представлены с использованием двойных кавычек (DQUOTE, код ASCII - 34), окружающих такие символы.

```

qtext     = %d33 /                    ; Печатаемые символы US-ASCII,
          %d35-91 /                   ; не включая "\"
          %d93-126 /                  ; и символа кавычек
          obs-qtext
qcontent  = qtext / quoted-pair
quoted-string = [CFWS]
          DQUOTE *([FWS] qcontent) [FWS] DQUOTE
          [CFWS]

```

Строка в кавычках трактуется как единый элемент. Т. е. семантически строка в кавычках эквивалентна атому. Поскольку строки в кавычках могут содержать FWS, в них возможна фальцовка. Отметим также, что благодаря возможности использования внутри кавычек пар с кватированием, строки в кавычках могут содержать также символы кавычек и обратной дробной черты, если они представлены в форме пар с кватированием.

Семантически, ни опциональные CFWS за пределами кавычек, ни сами символы кавычек не являются частью строки в кавычках — к такой строке относятся только символы, расположенные между кавычками. Как было отмечено выше, символ «\» в паре с кватированием или CRLF в FWS/CFWS, включенные в строку в кавычках, семантически невидимы и, следовательно, не являются частью строки в кавычках.

3.2.5. Прочие лексемы

Определены три дополнительных лексемы: слово (word) и фраза (phrase) для комбинаций атомов и/или строк в кавычках и неструктурированный текст (unstructured) для использования в бесструктурных полях заголовков и в некоторых местах структурированных полей.

```

word      = atom / quoted-string
phrase    = 1*word / obs-phrase
unstructured = (*( [FWS] VCHAR ) *WSP) / obs-unstruct

```

3.3. Задание даты и времен

Значения даты и времени появляются в нескольких полях заголовка. В этом параграфе определяет синтаксис для задания даты и времени. Хотя спецификация даты/времени допускает включение пробельных символов фальцовки, **рекомендуется** использовать один пробел в каждом случае включения FWS (так, где это требуется или допускается); некоторые старые реализации неспособны корректно интерпретировать пробельные символы фальцовки.

```

date-time = [ day-of-week "," ] date time [CFWS]
day-of-week = ([FWS] day-name) / obs-day-of-week
day-name   = "Mon" / "Tue" / "Wed" / "Thu" / "Fri" / "Sat" / "Sun"
date       = day month year           ; число, месяц год
day        = ([FWS] 1*2DIGIT FWS) / obs-day
month      = "Jan" / "Feb" / "Mar" / "Apr" / "May" / "Jun" /
          "Jul" / "Aug" / "Sep" / "Oct" / "Nov" / "Dec"
year       = (FWS 4*DIGIT FWS) / obs-year
time       = time-of-day zone         ; часовой пояс
time-of-day = hour ":" minute [ ":" second ]
hour       = 2DIGIT / obs-hour
minute     = 2DIGIT / obs-minute
second     = 2DIGIT / obs-second
zone       = (FWS ( "+" / "-" ) 4DIGIT) / obs-zone

```

¹Лексема specials не используется в данной спецификации. Это просто видимые символы (т. е., не управляющие коды и не пробелы), которые не появляются в atext. Появление таких лексем обусловлено их полезностью для реализаций, использующих инструменты лексического анализа сообщений. Каждый символ в specials может использоваться для индикации точки маркировки при лексическом анализе.

Дата (day) показывает порядковый номер дня в месяце. Год (year) может принимать любое значение, не ранее 1900.

Время суток (time-of-day) показывает число часов, минут и (опционально) секунд, прошедших с полуночи указанной даты.

Значения date и time-of-day **следует** указывать по местному времени.

Часовой пояс (zone) показывает смещение относительно универсального времени (UTC¹, ранее использовался термин GMT²) значений местного времени, указанного в date и time-of-day. Знаки «+» и «-» показывают направление смещения — вперед (т. е., к востоку) или назад (к западу) от универсального времени. Первые две цифры показывают разницу с универсальным временем в часах, а две последних — дополнительную разницу в минутах. Следовательно, +hhmm означает смещение на +(hh * 60 + mm) минут от универсального времени, а -hhmm — на -(hh * 60 + mm) минут. Для индикации часового пояса, время которого совпадает с универсальным, **следует** использовать форму +0000. Хотя вариант -0000 указывает на тот же часовой пояс, этот вариант используется для индикации того, что время было указано системой, которая может находиться в часовом поясе, отличном от UT, а значение date-time не содержит информации о часовом поясе.

Значение date-time **должно** быть семантически корректным. Т. е., значение day-of-week (при его наличии) **должно** содержать день недели, числовое значение day-of-month **должно** находиться в диапазоне от 1 до числа дней в соответствующем месяце (указанного года), значение time-of-day **должно** находиться в диапазоне от 00:00:00 до 23:59:60 (число секунд, допустимое для смены суток; см. [RFC1305]), а две последних цифры поля zone **должны** иметь значение от 00 до 59.

3.4. Задание адреса

Адреса появляются в нескольких полях заголовка для индикации отправителей и получателей сообщения. Адрес может задавать персональный почтовый ящик или группу почтовых ящиков.

```
address      = mailbox / group
mailbox      = name-addr / addr-spec
name-addr    = [display-name] angle-addr
angle-addr   = [CFWS] "<" addr-spec ">" [CFWS] / obs-angle-addr
group        = display-name ":" [group-list] ";" [CFWS]
display-name = phrase
mailbox-list = (mailbox *("," mailbox)) / obs-mbox-list
address-list = (address *("," address)) / obs-addr-list
group-list   = mailbox-list / CFWS / obs-group-list
```

Почтовый ящик принимает почту. Это концептуальный объект, который не обязательно связан с файловым хранилищем. Например, некоторые сайты могут выводить почтовые сообщения на принтер или передавать их на специальные адресуемые устройства вывода.

Обычно почтовый ящик состоит из двух частей: (1) необязательное отображаемое имя, которое идентифицирует имя получателя (человека или системы) и может выводиться пользователю в почтовых программах и (2) поле addr-spec, заключенное в угловые скобки («<» и «>»). Существует дополнительная форма указания почтового ящика, в которой имя получателя отсутствует, а addr-spec указывается без угловых скобок. Описание addr-spec для адресов Internet приведено в параграфе 3.4.1.

Примечание. Некоторое унаследованные реализации используют простую форму, где addr-spec указывается без угловых скобок, а имя получателя указывается в скобках, как комментарий вслед за addr-spec. Поскольку трактовка комментариев не задается спецификацией, реализациям для задания связанного с почтовым ящиком отображаемого имени **следует** использовать полную форму name-addr взамен такой унаследованной формы. Кроме того, поскольку некоторые унаследованные реализации интерпретируют комментарии, в общем случае **не следует** использовать комментарии в поле адреса во избежание возможной путаницы.

Когда желательно трактовать несколько почтовых ящиков, как один объект (например, список рассылки), может использоваться групповая конструкция. Такая конструкция позволяет отправителю указать именованную группу получателей. Это обеспечивается путем создания для группы отображаемого имени, за которым следует список разделенных запятыми почтовых ящиков произвольного (включая 0) размера, завершающийся точкой с запятой (;). Поскольку список почтовых ящиков может быть пустым, использование групповой конструкции также обеспечивает простой способ взаимодействия с адресатами, когда сообщение передается одной или множеству именованных групп адресатов без указания индивидуальных почтовых ящиков этих адресатов.

3.4.1. Задание addr-spec

Поле addr-spec представляет собой специфический для Internet идентификатор, содержащий локально интерпретируемую строку, за которой следует символ @ (код ASCII - 64) и доменное имя Internet. Эта локально интерпретируемая строка представляет собой строку в кавычках или атом с точкой. Если строка может быть представлена атомом с точкой (т. е., не содержит символов, кроме atext и завершающей точки или, за которой следуют символы atext), **следует** использовать форму dot-atom и **не следует** применять форму quoted-string. Комментарии и пробельные символы фальцовки не следует помещать рядом с @ в поле addr-spec.

Примечание. Здесь приведен либеральный синтаксис для доменной части addr-spec. Однако доменная часть содержит адресную информацию, задаваемую и используемую другими протоколами (например, [RFC1034], [RFC1035], [RFC1123], [RFC5321]). Следовательно, на реализации возлагается ответственность за соответствие синтаксиса адресов контексту, в котором они используются.

```
addr-spec    = local-part "@" domain
local-part   = dot-atom / quoted-string / obs-local-part
domain       = dot-atom / domain-literal / obs-domain
domain-literal = [CFWS] "[" *([FWS] dtext) [FWS] "]" [CFWS]
dtext        = %d33-90 / ; Печатаемые символы US-ASCII
```

¹Coordinated Universal Time — скоординированное универсальное время.

²Greenwich Mean Time — время по Гринвичу.

```
%d94-126 / ; не включая
obs-dtext ; "[", "]", "\""
```

Доменная часть идентифицирует точку, в которую доставляется почта. В форме dot-atom она интерпретируется, как доменное имя Internet (имя хоста или узла MX) в соответствии с описаниями в [RFC1034], [RFC1035] и [RFC1123]. В форме domain-literal доменная часть интерпретируется, как «дословный» Internet-адрес конкретного хоста. В обоих случаях использование адресации и транспортировка почты на конкретный хост определяются отдельными документами (такими, как [RFC5321]). Эти механизмы выходят за пределы настоящей спецификации.

Локальная часть (local-part) зависит от домена. В адресах она просто интерпретируется на конкретном хосте, как имя почтового ящика.

3.5. Общий синтаксис сообщений

Сообщение состоит из полей заголовка, за которыми может следовать тело сообщения. Строки сообщения **должны** иметь размер не более 998 символов, не считая CRLF, но **рекомендуется** использовать строки не длиннее 78 без учета CRLF (см. параграф 2.1.1). Хотя в теле сообщения **могут** появляться любые символы, перечисленные в правиле text, использование управляющих символов US-ASCII (коды 1- 8, 11, 12, 14 — 31) не является хорошим тоном, поскольку их интерпретация при отображении на приемной стороне не гарантируется.

```
message = (fields / obs-fields)
         [CRLF body]
body = (*998text CRLF) *998text / obs-body
text = %d1-9 / ; Символы, за исключением CR и LF
      %d11 /
      %d12 /
      %d14-127
```

Поля заголовков, несущие основную семантическую информацию, определены в параграфе 3.6. Тело сообщения представляет собой просто набор строк текста, интерпретируемых в соответствии с настоящей спецификацией.

3.6. Определения полей

В этом параграфе определены поля заголовка сообщения. Все поля заголовка имеют одинаковую синтаксическую структуру: имя поля, за которым следует двоеточие (:) и тело (значение) поля. Конкретный синтаксис каждого поля заголовка определен в соответствующем подпараграфе¹.

Важно подчеркнуть, что порядок следования полей заголовка не гарантируется. Поля заголовков могут появляться в произвольном порядке. Более того, известно, что порядок полей заголовков может изменяться при передаче сообщений через Internet. Однако в соответствии с данной спецификацией порядок полей заголовка **не следует** менять при передаче или преобразовании сообщений. Более важно отметить, что порядок трассировочных полей и полей resent изменять **недопустимо** и следует сохранять эти поля в блоках, добавляемых в начало сообщения (prepend). Дополнительная информация об этих полях содержится в параграфах 3.6.6 и 3.6.7.

Обязательными полями заголовка являются только поле даты и поле адреса отправителя сообщения. Все остальные поля являются синтаксически опциональными. Дополнительная информация приведена в таблице вслед за определением.

```
fields = *(trace
          *optional-field /
          *(resent-date /
            resent-from /
            resent-sender /
            resent-to /
            resent-cc /
            resent-bcc /
            resent-msg-id)
          *(orig-date /
            from /
            sender /
            reply-to /
            to /
            cc /
            bcc /
            message-id /
            in-reply-to /
            references /
            subject /
            comments /
            keywords /
            optional-field)
```

Приведенная ниже таблица показывает минимальное и максимальное число полей каждого типа в разделе заголовков сообщения, а также ограничения на использование полей. Звездочка (*) после в колонке минимального или максимального числа полей говорит о наличии дополнительных ограничений, указанных в колонке «Примечания».

Поле	Минимум	Максимум	Примечания
trace	0	Не ограничен	Блок добавляется в начало, см. параграф 3.6.7
resent-date	0*	Не ограничен*	Одно на блок; требуется при наличии других полей resent, см. параграф 3.6.6
resent-from	0	Не ограничен*	Одно на блок, см. параграф 3.6.6
resent-sender	0*	Не ограничен*	Одно на блок; должно присутствовать при наличии множества адресов, см. параграф 3.6.6
resent-to	0	Не ограничен*	Одно на блок, см. параграф 3.6.6

¹Синтаксис ABNF для каждого поля в последующих подпараграфах содержит двоеточие после имени поля. Однако для краткости двоеточия могут не упоминаться в текстовом описании синтаксиса. В любом случае двоеточия обязательны.

resent-cc	0	Не ограничен*	Одно на блок, см. параграф 3.6.6
resent-bcc	0	Не ограничен*	Одно на блок, см. параграф 3.6.6
resent-msg-id	0	Не ограничен*	Одно на блок, см. параграф 3.6.6
orig-date	1	1	
from	1	1	См. sender и параграф 3.6.2
sender	0*	1	Должно присутствовать при наличии множества адресов, см. параграф 3.6.2
reply-to	0	1	
to	0	1	
cc	0	1	
bcc	0	1	
message-id	0*	1	Следует включать, см. параграф 3.6.4
in-reply-to	0*	1	Следует включать, см. параграф 3.6.4
references	0*	1	Следует включать, см. параграф 3.6.4
subject	0	1	
comments	0	Не ограничен	
keywords	0	Не ограничен	
optional-field	0	Не ограничен	

Точная интерпретация каждого поля рассмотрена в последующих параграфах.

3.6.1. Поле даты создания

Поле даты создания состоит из имени поля «Date», за которым следуют значения даты и времени.

```
orig-date      = "Date:" date-time CRLF
```

Поле указывает дату и время, когда отправитель, указанный в сообщении, завершил подготовку сообщения к передаче системе доставки. Например, это может быть момент нажатия пользователем кнопки «send» или «submit» в почтовой программе. В любом случае, это поле не предназначено для указания времени реальной передачи сообщения, а содержит значение времени, когда человек или другой создатель сообщения завершил подготовку сообщения к передаче (например, пользователь портативного компьютера, не подключенного к сети, мог просто поместить сообщение в очередь для доставки; поле даты создания предназначено для хранения даты и времени в момент постановки письма в очередь, а не момента подключения пользователя к сети).

3.6.2. Поля источника

В число полей источника сообщения входят from и sender (когда применимо), а также опционально — reply-to. Поле from содержит имя «From» и разделенный запятыми список из одного или нескольких имен почтовых ящиков. Если это поле содержит более одного адреса почтового ящика в списке, в сообщение **должно** включаться поле «Sender» с указанием одного почтового ящика. Дополнительно **может** также включаться поле «Reply-To», содержащее список разделенных запятыми почтовых ящиков (не менее одного).

```
from           = "From:" mailbox-list CRLF
sender         = "Sender:" mailbox CRLF
reply-to      = "Reply-To:" address-list CRLF
```

Поле источника показывает почтовый ящик(и) источника сообщения. Поле «From:» задает автора (авторов) сообщения, т. е., почтовый ящик(и) человека или системы, ответственных за создание данного сообщения. Поле «Sender:» указывает почтовый ящик агента, ответственного за реальную передачу сообщения. Например, если секретарь отправил письмо от имени своего руководителя, почтовый ящик секретаря будет указан в поле «Sender:», а адрес действительного автора сообщения — в поле «From:». Если источник сообщения может быть задан единственным почтовым ящиком (автор и отправитель совпадают), поле «Sender:» использовать **не следует**¹. В остальных случаях это поле **следует** включать в заголовок.

Поля источника также обеспечивают информацию, требуемую для ответа на сообщение. При наличии поля «Reply-To:» оно указывает адрес(а), по которому отправитель сообщения предлагает направлять ответ. В отсутствие поля «Reply-To:» ответ **следует** по умолчанию отправлять по адресу (адресам), указанному в поле «From:», если автором ответа явно не указан иной адрес.

В любом случае в поле «From:» **не следует** указывать адрес, не имеющий отношения к автору письма. Формирование адресов для отправки ответов на сообщения дополнительно рассматривается в параграфе 3.6.3.

3.6.3. Поля адресов получателей

К полям получателей относятся три однотипных поля, каждое из которых содержит имя («To», «Cc» или «Bcc») и список из одного или множества разделенных запятыми адресов (почтовых ящиков или групп).

```
to            = "To:" address-list CRLF
cc           = "Cc:" address-list CRLF
bcc          = "Bcc:" [address-list / CFWS] CRLF
```

Эти поля задают получателей сообщения. Каждое из полей может содержать один или множество адресов, идентифицирующих получателей сообщения. Различие между полями заключается только в способе их использования.

Поле «To:» содержит адрес(а) основного полячателя (получателей) сообщения.

Поле «Cc:» (от «Carbon Copy» - копия по аналогии с печатью под копиру на пишущей машинке) содержит адреса других лиц, которым направляется это сообщение, хотя содержимое сообщения может быть не адресовано им напрямую.

¹Информация о передающем сообщении лице (системе) присутствует всегда. Отсутствие поля «Sender:» иногда ошибочно трактуется, как отсутствие информации об агенте, ответственном за передачу сообщения. Реальное отсутствие этого поля означает лишь совпадение отправителя и автора, что позволяет избавиться от ненужного дублирования информации в поле «Sender:».

Поле «Vcc:» (от «Blind Carbon Copy» - «слепая копия» по аналогии с последним экземпляром при печати под копиру) содержит адреса получателей, которые не будут показаны другим получателям этого сообщения. Существует три варианта использования поля «Vcc:». В первом случае, когда сообщение с полем «Vcc:» готовится к отправке, строка «Vcc:» удаляется из него, хотя все адресаты (включая указанные в поле «Vcc:») получают копию этого письма. Во втором варианте получатели, указанные в полях «To:» и «Cc:», получают сообщение с удаленной строкой «Vcc:», но адресаты, указанные в поле «Vcc:» получают копию письма с сохраненной строкой «Vcc:» (при указании в поле «Vcc:» множества адресатов некоторые реализации на самом деле шлют каждому адресату из поля «Vcc:» отдельную копию письма, содержащую в этом поле только адрес данного получателя). И, наконец, когда поле «Vcc:» не содержит адресов, это поле может включаться во все копии сообщения для адресатов, заданных другими полями. Выбор конкретного варианта использования поля «Vcc:» определяется реализацией, но при этом следует принимать во внимание информацию, приведенную ниже в разделе «Вопросы безопасности».

Когда сообщение является ответом на другое письмо, почтовые ящики авторов исходного письма (значение поля «From:») или почтовые ящики, указанные в поле «Reply-To:» (если оно есть), **могут** появляться в поле «To:» ответного письма, поскольку эти адресаты являются явными отправителями исходного письма. Если сообщение передается в ответ на письмо, имеющее поля получателей, зачастую бывает полезно отправить копию ответа всем получателям сообщения в дополнение к отправке письма автору. При создании такого отклика адреса из полей «To:» и «Cc:» исходного сообщения **могут** появляться в поле «Cc:» ответного письма, поскольку эти адресаты были открыто указаны в числе получателей копии исходного письма. Если в исходном письме присутствует поле «Vcc:», адреса из этого поля **могут** появляться в поле «Vcc:» ответа на письмо, но их **не следует** включать в поле «To:» или «Cc:».

Примечание. Некоторые почтовые программы поддерживают команды автоматического ответа, при котором адреса получателей исходного письма включаются в число адресов получателей ответа. Поведение таких команд зависит от реализации и выходит за пределы настоящего документа. В частности, вопрос включения адресов получателей исходного письма при наличии в нем поля «Reply-To:» здесь не рассматривается.

3.6.4. Поля идентификации

Хотя в таблице параграфа 3.6 поле «Message-ID:» указано, как необязательное, это поле **следует** включать в каждое сообщение. Более того, в ответные сообщения **следует** включать поля «In-Reply-To:» и «References:» в соответствии с приведенным ниже описанием.

Поле «Message-ID:» содержит один уникальный идентификатор сообщения. Каждое из полей «References:» и «In-Reply-To:» содержит один или множество уникальных идентификаторов сообщений, опционально разделенных символами CFWS.

Синтаксис идентификатора сообщения (msg-id) является ограниченной версией конструкции addr-spec, заключенной в угловые скобки «<» и «>». В отличие от addr-spec, этот синтаксис разрешает форму dot-atom-text слева от символа @ и не имеет символов CFWS где-либо внутри идентификатора.

Примечание. Как и в случае addr-spec, для правой части msg-id (после символа @) дается либеральный синтаксис. Однако ниже в этом параграфе сказано, что **рекомендуется** использовать справа от знака @ доменное имя. Как и прежде, синтаксис доменного имени задается и используется в других протоколах (например, [RFC1034], [RFC1035], [RFC1123], [RFC5321]). Следовательно, на реализации возлагается вопрос соответствия адресов контексту, в котором они используются.

```
message-id      = "Message-ID:" msg-id CRLF
in-reply-to     = "In-Reply-To:" 1*msg-id CRLF
references      = "References:"  1*msg-id CRLF
msg-id          = [CFWS] "<" id-left "@" id-right ">" [CFWS]
id-left         = dot-atom-text / obs-id-left
id-right        = dot-atom-text / no-fold-literal / obs-id-right
no-fold-literal = "[" *dtext "]"
```

Поле «Message-ID:» обеспечивает уникальный идентификатор сообщения, который указывает на конкретный вариант конкретного письма. Уникальность идентификатора гарантируется хостом, создающим сообщение (см. ниже). Этот идентификатор предназначен для машинной обработки и не имеет большого смысла для человека. Идентификатор сообщения относится только к одному варианту конкретного письма — для следующего экземпляра сообщения будет создан другой идентификатор.

Примечание. Существует множество ситуаций, когда сообщения «изменяются», но эти изменения не порождают нового экземпляра данного сообщения и, следовательно, сообщение не получает нового идентификатора. Например, когда сообщения вводятся в транспортную систему, в начало сообщения зачастую добавляются дополнительные поля заголовков, такие, как поля трассировки (параграф 3.6.7) и поля пересылки (параграф 3.6.6). Добавление таких полей в заголовки не меняет сообщения, как такового, и, следовательно, значение «Message-ID:» сохраняется. Во всех случаях содержимое, которое желает передать отправитель (то же самое сообщение или другое), а не какое-либо синтаксическое отличие, которое появляется (или не появляется) в сообщении, определяет сохранение или изменение поля «Message-ID:».

Поля «In-Reply-To:» и «References:» используются при создании ответных сообщений. Они содержат идентификатор исходного сообщения и идентификаторы других сообщений (например, в случае ответа на письмо, которое само является ответом на другое письмо). Поле «In-Reply-To:» может использоваться для идентификации сообщения (сообщений), на которое отвечает данное сообщение, тогда как поле «References:» может служить для идентификации «ветви» разговора.

При создании ответа на сообщение поля «In-Reply-To:» и «References:» в ответном письме строятся в соответствии с приведенным ниже описанием.

Поле «In-Reply-To:» ответного письма будет содержать информацию из поля «Message-ID:» исходного («родительского») сообщения. Если ответ дается на несколько писем сразу, поле «In-Reply-To:» будет содержать

информацию из полей «Message-ID:» всех родительских сообщений¹. Если в родительских сообщениях нет полей «Message-ID:», в ответе не будет содержаться поля «In-Reply-To:».

Поле «References:» будет включать содержимое поля «References:» из родительского письма (если там это поле присутствует), вслед за которым будет включено родительское поле «Message-ID:» (при его наличии). Если в родительском сообщении нет поля «References:», но имеется поле «In-Reply-To:» с единственным идентификатором, в ответе поле «References:» будет включать содержимое родительского поля «In-Reply-To:», за которым будет следовать содержимое родительского поля «Message-ID:» (при его наличии). Если в родительском сообщении нет полей «References:», «In-Reply-To:» и «Message-ID:», в ответе не будет поля «References:».

Идентификатор сообщения (msg-id) **должен** быть уникальным в глобальном масштабе. Эту уникальность **должен** обеспечивать генератор идентификаторов сообщений. Существует несколько алгоритмов, обеспечивающих решение этой задачи. Поскольку синтаксис msg-id подобен синтаксису addr-spec (за исключением запрета на включение строк в кавычках, комментариев и фальцовочных пробелов), хорошим методом является включение в идентификатор доменного имени (или полного адреса IP) хоста, на котором создается идентификатор сообщения, справа от знака @ (доменные имена и адреса IP обычно уникальны) и включение текущего абсолютного значения даты и времени в комбинации с неким другим уникальным в данный момент (возможно последовательным) идентификатором, доступным в системе (например, идентификатором процесса), слева от @. Хотя будут работать и другие алгоритмы, **рекомендуется** использовать в правой части некий идентификатор домена (имя хоста или нечто иное), чтобы генератор идентификатора сообщения мог гарантировать уникальность левой части идентификатора в масштабе данного домена.

Семантически угловые скобки не являются частью msg-id; идентификатором является строка символов между скобками.

3.6.5. Информационные поля

Информационные поля являются необязательными. Поля «Subject:» и «Comments:» являются бесструктурными, как определено в параграфе 2.2.1, и, следовательно, могут содержать текст или пробельные символы для фальцовки. Поле «Keywords:» содержит список из одного или более разделенных запятыми слов или строк в кавычках.

```
subject      = "Subject:" unstructured CRLF
comments    = "Comments:" unstructured CRLF
keywords    = "Keywords:" phrase *(", " phrase) CRLF
```

Эти три поля предназначены только для включения понятной человеку информации о сообщении. Поле «Subject:» используется наиболее широко и содержит короткую строку, описывающую тему сообщения. При использовании в ответах это поле может начинаться с префикса «Re: » (сокращение от латинского «in re», означающего «по вопросу ..»), за которым следует содержимое поля «Subject:» исходного письма. В таких случаях следует использовать префикс «Re: » только один раз, поскольку использование другого текста или включение нескольких префиксов может приводить к нежелательным последствиям. Поле «Comments:» содержит произвольную информацию, дополняющую текст в теле письма. Поле «Keywords:» содержит список разделенных запятыми слов и фраз " field contains a comma-separated list of important words and phrases that might be useful for the recipient.

3.6.6. Поля пересылки

Поля пересылки **следует** добавлять во все сообщения, которые повторно вводятся пользователем в транспортную систему. Каждый раз, когда это происходит, **следует** добавлять отдельный набор полей пересылки. Все поля пересылки, соответствующие отдельному факту пересылки, **следует** группировать. Каждый набор полей пересылки помещается в начало сообщения; т. е. самый свежий набор таких полей появляется в сообщении первым. При добавлении полей пересылки никакие другие поля сообщения не меняются.

Каждое поле пересылки синтаксически соответствует определенному полю при обычной передаче. Например, поле «Resent-Date:» соответствует полю «Date:», а «Resent-To:» - полю «To:». Во всех таких случаях синтаксис тела полей пересылки идентичен синтаксису соответствующего поля, описанному выше.

При использовании полей пересылки **должны** передаваться поля «Resent-From:» и «Resent-Date:». **Следует** передавать также поле «Resent-Message-ID:». Поле «Resent-Sender:» **не следует** использовать, если это поле будет идентично полю «Resent-From:».

```
resent-date  = "Resent-Date:" date-time CRLF
resent-from  = "Resent-From:" mailbox-list CRLF
resent-sender = "Resent-Sender:" mailbox CRLF
resent-to    = "Resent-To:" address-list CRLF
resent-cc    = "Resent-Cc:" address-list CRLF
resent-bcc   = "Resent-Bcc:" [address-list / CFWS] CRLF
resent-msg-id = "Resent-Message-ID:" msg-id CRLF
```

Поля пересылки² служат для идентификации сообщений, как повторно вводимых пользователем в транспортную систему. Цель использования этих полей заключается в предоставлении конечному адресату сообщения в таком виде, как будто оно было получено непосредственно от исходного отправителя с сохранением всех исходных полей. Каждый набор полей пересылки соответствует одному факту пересылки. Т. е., при многократной пересылке сообщения каждый из таких наборов описывает отдельный факт пересылки. Поля пересылки являются информационными. Эти поля

¹Некоторые реализации анализируют поле «References:» для вывода информации о «течении дискуссии». Такие реализации предполагают, что каждое новое сообщение является ответом только на одно родительское письмо и, следовательно, они способны обеспечить обратную трассировку поля «References:» для нахождения «родителя» каждого перечисленного здесь сообщения. Следовательно, попытки формирования поля «References:» для ответа на множество сообщений сразу, являются недопустимыми; рассмотрение этого вопроса выходит за рамки документа.

²Повторный ввод сообщения в транспортную систему и использование полей пересылки отличаются от операции «forwarding» (пересылка). В последней используется два сообщения — суть этой операции заключается в том, что пользователь почтовой программы, получивший сообщение, может отправить копию этого сообщения другому лицу, которое в результате получает новое сообщение («копию»). Пересланное таким способом сообщение приходит не от исходного отправителя, а от того, кто принял решение о его пересылке и является, по сути дела, новым сообщением. Пересылка может заключаться также в приеме сообщения почтовой транспортной системой и отправке его в другое место для окончательной доставки. Поля пересылки не предназначены для использования при такой пересылке.

недопустимо использовать в процессах автоматической генерации ответов и других операциях автоматизированной обработки сообщений.

Поля инициатора пересылки указывают почтовый ящик лиц(а) или систем(ы), переславших это сообщение. Как и для обычных полей инициатора существует две формы — простой вариант «Resent-From:», содержащий почтовый ящик выполняющего пересылку лица, и более сложный вариант, когда одно лицо (идентифицируется полем «Resent-Sender:») пересылает сообщение от имени одного или множества других лиц (идентифицируются полем «Resent-From:»).

Примечание. При ответе на пересланное сообщение исходные поля «From:», «Reply-To:», «Message-ID:» и др. используются в обычном порядке. Поля пересылки являются только информационными и **недопустимо** использовать их при ответе на сообщение.

Поле «Resent-Date:» указывает дату и время диспетчеризации сообщения при его пересылке. Подобно полю «Date:», оно содержит не дату и время реальной отправки сообщения, а дату и время постановки в очередь.

Функционально поля «Resent-To:», «Resent-Cc:» и «Resent-Bcc:» идентичны полям «To:», «Cc:» и «Bcc:», соответственно, за исключением того, что они указывают получателей пересланного сообщения, а не исходного.

Поле «Resent-Message-ID:» содержит уникальный идентификатор пересланного сообщения.

3.6.7. Поля трассировки

Поля трассировки представляют собой группу полей заголовка, включающую опциональное поле «Return-Path:» и одно или множество полей «Received:». Поле заголовка «Return-Path:» содержит пару угловых скобок, в которые заключено необязательное значение addr-spec. Поле «Received:» содержит (возможно пустой) список маркеров, за которым следует точка с запятой (;) и значение даты и времени. Каждый маркер должен представлять собой элемент типа word, angle-addr, addr-spec или domain. Спецификации, описывающие использование полей трассировки (такие, как [RFC5321]), могут вносить дополнительные ограничения.

```
trace           = [return]
                  1*received
return          = "Return-Path:" path CRLF
path            = angle-addr / ([CFWS] "<" [CFWS] ">" [CFWS])
received        = "Received:" *received-token ";" date-time CRLF
received-token = word / angle-addr / addr-spec / domain
```

Полное описание использования полей трассировки в почте Internet содержится в [RFC5321]. В данной спецификации трассировочные поля рассматриваются исключительно в качестве информационных и любая формальная интерпретация этих полей выходит за рамки документа.

3.6.8. Дополнительные поля

В сообщениях могут появляться поля, не рассмотренные в этом документе. Такие поля **должны** соответствовать синтаксису optional-field. Этот синтаксис включает имя поля, состоящее из печатаемых символов US-ASCII без двоеточий и пробелов (SP), за которым следует двоеточие и произвольный текст, соответствующий синтаксису бесструктурных полей.

Недопустимо совпадение имен дополнительных полей с именами полей, указанными в данном документе.

```
optional-field = field-name ":" unstructured CRLF
field-name     = 1*ftext
ftext          = %d33-57 / ; Печатаемые символы US-ASCII,
                  %d59-126 ; за исключением «:».
```

В настоящей спецификации дополнительные поля считаются неинтерпретируемыми.

4. Устаревший синтаксис

В ранних версиях спецификации допускался иной (обычно, более либеральный) синтаксис по сравнению с дозволенным в этой версии. Кроме того, некоторые синтаксические элементы, используемые в почтовых сообщениях Internet, никогда не были документированы. Хотя генерация таких синтаксических форм **недопустима** в соответствии с разделом 3, они **должны** восприниматься и разбираться соответствующим спецификации получателем. В этом разделе документированы многие элементы такого типа. Использование грамматики раздела 3 с добавлением содержащихся в данном разделе определений, создает грамматику для использования при интерпретации сообщений.

Примечание. В этом разделе рассматриваются лишь те синтаксические формы, которые любая реализация **должна** подобающим образом интерпретировать. Однако существуют сообщения Internet, которые не соответствуют спецификации даже с учетом приведенного в этом разделе дополнительного синтаксиса. Отсутствие той или иной формы в данном документе отнюдь не означает, что она предназначена для нарушения работы компьютерной программы, а некорректный формат не обязательно ведет к потере данных в любой реализации. Устойчивость реализации к ошибкам обеспечивается самой реализацией.

Важное различие между устаревшим (интерпретация) и современным (генерация) синтаксисом заключается в том, что структурированные поля заголовков (т. е., текст между двоеточием и завершающей строку последовательностью CRLF) могут содержать пробелы (включая фальцовочные) и комментарии между любыми лексемами. Это позволяет создавать множество сложных форм, разбор которых может оказаться затруднительным для некоторых реализаций.

Другим ключевым различием между устаревшим и современным синтаксисом является правило, приведенное в параграфе 3.2.2 и запрещающее включение фальцовочных символов в строки, содержащие только пробелы в комментарии (см. обсуждение фальцовочных пробелов в параграфе 4.2).

В этом разделе также рассматриваются некоторые символы, которые раньше разрешалось использовать в сообщениях. Символ NUL (код ASCII - 0) считался разрешенным, но сейчас к таковым не относится. Аналогично, управляющие символы US-ASCII, отличные от CR, LF, SP и HTAB (коды ASCII от 1 до 8, 11, 12, 14 — 31 и 127)

разрешалось использовать в полях заголовков. Символы CR и LF разрешалось использовать в сообщениях не только в форме последовательности CRLF; такое использование также рассматривается здесь.

Остальные различия в синтаксисе и семантике рассматриваются в соответствующих подпараграфах.

4.1. Прочие устаревшие маркеры

Описанные здесь синтаксические элементы используются в устаревшем или основном синтаксисе. Отдельные символы CR, LF и NUL добавлены в obs-qp, obs-body и obs-unstruct. Управляющие символы US-ASCII добавлены в obs-qp, obs-unstruct, obs-ctext и obs-qttext. Символ точки (.) добавлен в obs-phrase¹. Поддерживается лексема obs-phrase-list для (возможно пустых) списков разделенных запятыми фраз, которые могут включать «пустые» элементы. Т. е. в таком списке могут быть две и более запятых, между которыми не содержится ничего; возможны также запятые в начале и в конце списка.

```
obs-NO-WS-CTL = %d1-8 / ; Управляющие символы US-ASCII,
                %d11 / ; не включая символов
                %d12 / ; возврата каретки,
                %d14-31 / ; перевода строки и
                %d127 ; пробельных символов

obs-ctext = obs-NO-WS-CTL
obs-qttext = obs-NO-WS-CTL
obs-utext = %d0 / obs-NO-WS-CTL / VCHAR
obs-qp = "\" (%d0 / obs-NO-WS-CTL / LF / CR)
obs-body = *((*LF *CR *((%d0 / text) *LF *CR)) / CRLF)
obs-unstruct = *((*LF *CR *(obs-utext *LF *CR)) / FWS)
obs-phrase = word *(word / "." / CFWS)
obs-phrase-list = [phrase / CFWS] *(", " [phrase / CFWS])
```

Отдельные символы CR и LF, появляющиеся в сообщениях, могут иметь двоякий смысл. Во многих случаях одиночные символы CR или LF некорректно используются вместо CRLF для индикации завершения строк. В остальных случаях одиночные символы CR и LF просто используются в качестве управляющих символов US-ASCII в традиционном их смысле.

4.2. Устаревшие пробелы для фальцовки

В устаревшем синтаксисе **можно** включать любое количество фальцовочных пробелов в тех случаях, когда разрешено правило obs-FWS. Это позволяет включить в строку две фальцовки подряд и, следовательно, делает возможными строки в заголовках полей, содержащие только пробельные символы.

```
obs-FWS = 1*WSP *(CRLF 1*WSP)
```

4.3. Устаревшие форматы даты и времени

Устаревший синтаксис для полей даты допускает использование 2 цифр для указания года, а также разрешает буквенные обозначения часовых поясов, которые использовались в ранних версиях спецификации. Этот синтаксис также позволяет включение комментариев и фальцовочных пробелов между большинством лексем.

```
obs-day-of-week = [CFWS] day-name [CFWS]
obs-day = [CFWS] 1*2DIGIT [CFWS]
obs-year = [CFWS] 2*DIGIT [CFWS]
obs-hour = [CFWS] 2DIGIT [CFWS]
obs-minute = [CFWS] 2DIGIT [CFWS]
obs-second = [CFWS] 2DIGIT [CFWS]
obs-zone = "UT" / "GMT" / ; Смещение от UT для Северной Америки
           "EST" / "EDT" / ; восток: - 5/ - 4
           "CST" / "CDT" / ; центр: - 6/ - 5
           "MST" / "MDT" / ; горы: - 7/ - 6
           "PST" / "PDT" / ; тихоокеанское побережье: - 8/ - 7
           ;
           %d65-73 / ; Военные часовые пояса — от A
           %d75-90 / ; до I и от K
           %d97-105 / ; до Z,
           %d107-122 ; в верхнем и нижнем регистре
```

При использовании 2 или трех цифр в поле года интерпретация выполняется следующим образом — если 2-значное обозначение года лежит в диапазоне от 00 до 49, к значению прибавляется 2000, т. е., год принимает значение от 2000 до 2049; если 2-значное обозначение года лежит в диапазоне от 50 до 99, к значению прибавляется 1900.

У устаревших часовых поясов идентификаторы «UT» и «GMT» указывают на «универсальное время» и «время по Гринвичу», соответственно; оба значения семантически эквивалентны «+0000».

Оставшиеся три символа часового пояса являются обозначениями часовых поясов США. Первая буква («E», «C», «M» или «P») означает «Eastern» (восточный), «Central» (центральный), «Mountain» (горный) и «Pacific» (тихоокеанский). Вторая буква может быть «S» (Standard — стандартное время) или «D» (Daylight Savings — летнее время).

Ниже приводится интерпретация используемых значений.

EDT семантически эквивалентно -0400

EST семантически эквивалентно -0500

CDT семантически эквивалентно -0500

¹Точка в obs-phrase не являлась допустимой в ранних версиях этой и всех других спецификаций. Точка (а не какой-либо другой символ из числа специальных) не допускалась во фразах, поскольку она при разборе создавала трудности в дифференциации фраз и частей в addr-spec (см. параграф 4.4). Здесь точка добавляется по той причине, что в настоящее время этот символ используется во множестве сообщений в поле отображаемого имени для адресов (особенно, в инициалах) и, следовательно, должен корректно интерпретироваться.

CST семантически эквивалентно -0600

MDT семантически эквивалентно -0600

MST семантически эквивалентно -0700

PDT семантически эквивалентно -0700

PST семантически эквивалентно -0800

Один символ военного часового пояса был определен нестандартным способом в [RFC0822] и, следовательно, его значение непредсказуемо. Исходные определения военных часовых поясов от «А» до «I» эквивалентны поясам от «+0100» до «+0900», соответственно; «К», «L» и «M» эквивалентны «+1000», «+1100» и «+1200», соответственно; «N» - «Y» эквивалентны поясам от «-0100» до «-1200», соответственно; «Z» эквивалентно «+0000». Однако в результате ошибки в [RFC0822] все эти обозначения **следует** трактовать, как эквивалентные «-0000» если явно не задано иное их толкование.

В почтовых сообщениях Internet применяются и другие многосимвольные (обычно от 3 до 5 букв) обозначения часовых поясов. Все обозначения, смысл которых непонятен, **следует** трактовать, как эквивалент «-0000» если явно не указано иное их толкование.

4.4. Устаревшая адресация

В адресации имеется четыре основных различия. Во-первых, в адресе почтового ящика разрешалось использовать перед addr-spec маршрутную часть, заключенную в угловые скобки. Маршрут является просто списком разделенных запятыми доменных имен, каждое из которых имеет префикс «@»; для завершения списка используется двоеточие (:). Во-вторых, разрешалось включать символы CFWS между разделенными точками элементами локальной части и домена (т. е., лексема dot-atom не использовалась). Кроме того, в local-part можно было в дополнение к атомам включать строки в кавычках. В-третьих, разрешалось включать в списки почтовых ящиков (mailbox-list) и списки адресов (address-list) пустые (null) элементы. Т. е., в списке могли следовать две или более запятых подряд, а также разрешалось присутствие запятых в начале и в конце списков. В-четвертых, разрешалось использовать управляющие символы US-ASCII и пары с кватированием в доменных литералах.

```
obs-angle-addr = [CFWS] "<" obs-route addr-spec ">" [CFWS]
obs-route      = obs-domain-list ":"
obs-domain-list = *(CFWS / "," ) "@" domain
               *("," [CFWS] ["@" domain])
obs-mbox-list  = *([CFWS] ",") mailbox *("," [mailbox / CFWS])
obs-addr-list  = *([CFWS] ",") address *("," [address / CFWS])
obs-group-list = 1*([CFWS] ",") [CFWS]
obs-local-part = word *("." word)
obs-domain     = atom *("." atom)
obs-dtext      = obs-NO-WS-CTL / quoted-pair
```

При интерпретации адресов маршрутную часть **следует** игнорировать.

4.5. Устаревшие поля заголовков

Синтаксически, основным различием является то, что устаревший синтаксис полей разрешал многократное включение любых полей в произвольном порядке. Кроме того, устаревший синтаксис разрешает включать любое количество пробелов перед двоеточием в конце имени поля.

```
obs-fields      = *(obs-return /
                  obs-received /
                  obs-orig-date /
                  obs-from /
                  obs-sender /
                  obs-reply-to /
                  obs-to /
                  obs-cc /
                  obs-bcc /
                  obs-message-id /
                  obs-in-reply-to /
                  obs-references /
                  obs-subject /
                  obs-comments /
                  obs-keywords /
                  obs-resent-date /
                  obs-resent-from /
                  obs-resent-send /
                  obs-resent-rply /
                  obs-resent-to /
                  obs-resent-cc /
                  obs-resent-bcc /
                  obs-resent-mid /
                  obs-optional)
```

За исключением поля адреса получателя (см. параграф 4.5.3) интерпретация множественного вхождения полей не специфицирована. Также нет спецификации для интерпретации полей трассировки и пересылки, которые не располагаются в блоке, предшествующем сообщению. Если в следующих подпараграфах явно не указано иное, интерпретация таких полей происходит идентично интерпретации подобных полей современного синтаксиса, описанных в разделе 3.

4.5.1. Устаревшее поле даты создания

```
obs-orig-date  = "Date" *WSF ":" date-time CRLF
```

4.5.2. Устаревшие поля отправителя

```
obs-from      = "From" *WSP ":" mailbox-list CRLF
obs-sender    = "Sender" *WSP ":" mailbox CRLF
obs-reply-to  = "Reply-To" *WSP ":" address-list CRLF
```

4.5.3. Устаревшие поля адресов получателей

```
obs-to        = "To" *WSP ":" address-list CRLF
obs-cc        = "Cc" *WSP ":" address-list CRLF
obs-bcc       = "Bcc" *WSP ":"
              (address-list / (*(CFWS) ",") [CFWS])) CRLF
```

При наличии в сообщении множества полей с адресом получателя их **следует** трактовать, как объединение адресного списка из первого поля с адресными списками из последующих полей путем добавления запятых и конкатенации строк.

4.5.4. Устаревшие поля идентификации

Устаревшие поля «In-Reply-To:» и «References:» отличаются от современного синтаксиса тем, что в них допускается включение фраз (слова или строки в кавычках). Устаревшие формы ревой и правой сторон msg-id разрешают промежуточные CFWS, что делает эти части синтаксически эквивалентными local-part и domain, соответственно.

```
obs-message-id = "Message-ID" *WSP ":" msg-id CRLF
obs-in-reply-to = "In-Reply-To" *WSP ":" *(phrase / msg-id) CRLF
obs-references  = "References" *WSP ":" *(phrase / msg-id) CRLF
obs-id-left     = local-part
obs-id-right    = domain
```

При интерпретации фразы в полях «In-Reply-To:» и «References:» игнорируются.

Семантически, ни один из дополнительных символов CFWS в local-part и domain не является частью obs-id-left и obs-id-right, соответственно.

4.5.5. Устаревшие информационные поля

```
obs-subject    = "Subject" *WSP ":" unstructured CRLF
obs-comments   = "Comments" *WSP ":" unstructured CRLF
obs-keywords   = "Keywords" *WSP ":" obs-phrase-list CRLF
```

4.5.6. Устаревшие поля пересылки

В устаревшем синтаксисе имеется поле «Resent-Reply-To:», которое состоит из имени поля, необязательных комментариев и фальцовочных пробелов, двоеточия и списка разделенных запятыми адресов.

```
obs-resent-from = "Resent-From" *WSP ":" mailbox-list CRLF
obs-resent-send = "Resent-Sender" *WSP ":" mailbox CRLF
obs-resent-date = "Resent-Date" *WSP ":" date-time CRLF
obs-resent-to   = "Resent-To" *WSP ":" address-list CRLF
obs-resent-cc   = "Resent-Cc" *WSP ":" address-list CRLF
obs-resent-bcc  = "Resent-Bcc" *WSP ":" (address-list / (*(CFWS) ",") [CFWS])) CRLF
obs-resent-mid  = "Resent-Message-ID" *WSP ":" msg-id CRLF
obs-resent-rply = "Resent-Reply-To" *WSP ":" address-list CRLF
```

Как и остальные поля пересылки, поле «Resent-Reply-To:» трактуется исключительно, как информационное.

4.5.7. Устаревшие поля трассировки

Поля obs-return и obs-received приведены здесь, как шаблоны определений, идентичные return и received в разделе 3. Полный синтаксис описан в [RFC5321].

```
obs-return     = "Return-Path" *WSP ":" path CRLF
obs-received   = "Received" *WSP ":" *received-token CRLF
```

4.5.8. Устаревшие дополнительные поля

```
obs-optional   = field-name *WSP ":" unstructured CRLF
```

5. Вопросы безопасности

Требуются определенные меры предосторожности при отображении сообщений на терминале или в программе эмуляции терминала. Многофункциональные терминалы могут реагировать на escape-последовательности и другие комбинации управляющих символов US-ASCII, что может вызывать неожиданные эффекты. В число таких эффектов может входить изменение клавиатурной раскладки и другие эффекты, способные приводить к нарушению работы и даже к повреждению данных. Управляющие символы могут вызывать (иногда программируемо) генерацию ответов на сообщения, что позволяет вводить команды от имени пользователя. Возможно также воздействие на подключенные к терминалу устройства (например, принтеры). Программы просмотра сообщений могут по своему усмотрению удалить опасные escape-последовательности из сообщения перед его выводом. Однако некоторые escape-последовательности могут быть нужны в сообщениях (см. [ISO.2022.1994], [RFC2045], [RFC2046], [RFC2047], [RFC2049], [RFC4288], [RFC4289]) и поэтому не следует удалять escape-последовательности без разбора.

Передача отличных от текста объектов в сообщениях может приводить к возникновению дополнительных опасностей. Рассмотрению таких вопросов посвящены документы [RFC2045], [RFC2046], [RFC2047], [RFC2049], [RFC4288] и [RFC4289].

Многие реализации используют поле «Всс:» (последний экземпляр), описанное в параграфе 3.6.3, для доставки сообщений некоторым получателям без ведома других адресатов этого сообщения. Некорректная обработка полей «Всс:» может приводить к утечке конфиденциальной информации, что, в свою очередь, может снижать уровень безопасности за счет распространения информации о существовании определенных почтовых адресов. Например, при использовании первого метода, описанного в параграфе 3.6.3, когда строка «Всс:» удаляется из сообщения, скрытые получатели не имеют явного указания на то, что они были скрыты от других адресатов (за исключением того факта, что их адреса отсутствуют в заголовке сообщения). По этой причине кто-либо из скрытых получателей сообщения может

отправить свой ответ всем показанным в заголовке получателям и непреднамеренно показать, что часть адресатов сообщения была скрыта. При использовании второго метода скрытые адресаты указываются в поле «Всс:» отдельной копии сообщения. Если поле «Всс:» содержит все скрытые адреса, получатели узнают о других скрытых адресатах. Даже при создании отдельного поля «Всс:» для каждого скрытого адресата, реализациям следует с осторожностью относиться к обработке ответов на такие сообщения, как указано в параграфе 3.6.3, во избежание непреднамеренного распространения информации о скрытых получателях сообщения другим адресатам.

6. Взаимодействие с IANA

Этот документ обновляет регистрации, перечисленные в [RFC4021], которые относятся к определениям из [RFC2822]. Агентство IANA обновило реестр Permanent Message Header Field Repository, включив в него в соответствии с [RFC3864] перечисленные ниже поля заголовков.

Имя поля: Date
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.1)

Имя поля: From
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.2)

Имя поля: Sender
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.2)

Имя поля: Reply-To
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.2)

Имя поля: To
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.3)

Имя поля: Cc
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.3)

Имя поля: Bcc
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.3)

Имя поля: Message-ID
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.4)

Имя поля: In-Reply-To
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.4)

Имя поля: References
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.4)

Имя поля: Subject
Применимый протокол: Mail
Состояние: стандарт
Автор/контроль изменений: IETF
Спецификация: Данный документ (параграф 3.6.5)

Имя поля: Comments
Применимый протокол: Mail
Состояние: стандарт

Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.5)
Имя поля:	Keywords
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.5)
Имя поля:	Resent-Date
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Resent-From
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Resent-Sender
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Resent-To
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Resent-Cc
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Resent-Bcc
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Resent-Reply-To
Применимый протокол:	Mail
Состояние:	устарел
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 4.5.6)
Имя поля:	Resent-Message-ID
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.6)
Имя поля:	Return-Path
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.7)
Имя поля:	Received
Применимый протокол:	Mail
Состояние:	стандарт
Автор/контроль изменений:	IETF
Спецификация:	Данный документ (параграф 3.6.7)
Дополнительная информация:	[RFC5321]

Приложение А. Примеры сообщений

В этом разделе представлены различные примеры сообщений. Примеры предназначены для оказания помощи разработчикам при реализации данной спецификации, но их не следует воспринимать как нормативы. Хотя примеры в этом разделе подобраны аккуратно, при возникновении противоречий между примером и текстом раздела 3 или 4 данного документа, предпочтение должно отдаваться спецификации, а не примеру.

Примеры сообщений отделены от текста строками «----», которые не являются частью сообщений.

Приложение А.1. Примеры адресации

В этом параграфе приведены примеры сообщений, которыми могут обмениваться два индивидуальных адресата.

Приложение A.1.1. Сообщение от одного лица другому с простой адресацией

Этот пример можно назвать каноническим сообщением. Письмо имеет одного автора (John Doe), одного получателя (Mary Smith), тему, дату, идентификатор сообщения и текст в теле письма.

```
-----
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>
```

```
This is a message just to say hello.
So, "Hello".
-----
```

Если у John есть секретарь Michael, который на самом деле отправляет сообщение, автором которого является по-прежнему John и ответы на письмо должны направляться секретарю, следует использовать поле sender:

```
-----
From: John Doe <jdoe@machine.example>
Sender: Michael Jones <mjones@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>
```

```
This is a message just to say hello.
So, "Hello".
-----
```

Приложение A.1.2. Различные типы почтовых ящиков

Это сообщение содержит множество адресов в полях получателей и использует иной формат адресов.

```
-----
From: "Joe Q. Public" <john.q.public@example.com>
To: Mary Smith <mary@x.test>, jdoe@example.org, Who? <one@y.test>
Cc: <boss@nil.test>, "Giant; \"Big\" Box" <syservices@example.net>
Date: Tue, 1 Jul 2003 10:52:37 +0200
Message-ID: <5678.21-Nov-1997@example.com>
```

```
Hi everyone.
-----
```

Отметим, что отображаемые имена Joe Q. Public и Giant; "Big" Box требуется заключать в двойные кавычки, поскольку первое имя включает точку, а во втором содержится точка с запятой и символы двойных кавычек (в форме пар с кватированием). Отображаемое имя Who?, напротив, может использоваться без кавычек, поскольку в атомах допускается использование вопросительного знака. Отметим также, что в адресах jdoe@example.org и boss@nil.test не указаны связанные с ними отображаемые имена, а для адреса jdoe@example.org используется упрощенная форма без угловых скобок.

Приложение A.1.3. Групповые адреса

```
-----
From: Pete <pete@silly.example>
To: A Group:Ed Jones <c@a.test>,joe@where.test,John <jdoe@one.test>;
Cc: Undisclosed recipients;
Date: Thu, 13 Feb 1969 23:32:54 -0330
Message-ID: <testabcd.1234@silly.example>
```

```
Testing.
-----
```

В этом примере поле «To:» включает группу «A Group», в состав которой входят 3 адреса, а в поле «Cc:» указана пустая группы получателей Undisclosed recipients (нераскрытые получатели).

Приложение A.2. Ответные сообщения

Следующая группа из трех примеров показывает поток обмена сообщениями между John и Mary. Сначала John отправляет сообщение Mary, на которое Mary отвечает и John в ответ отправляет письмо Mary.

Отметим специально поля «Message-ID:», «References:» и «In-Reply-To:» в каждом сообщении.

```
-----
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>
```

```
This is a message just to say hello.
So, "Hello".
-----
```

При отправке ответов поле Subject зачастую сохраняется с добавлением префикса «Re: », как описано в параграфе 3.6.5.

```
-----
From: Mary Smith <mary@example.net>
To: John Doe <jdoe@machine.example>
```

```
Reply-To: "Mary Smith: Personal Account" <smith@home.example>
Subject: Re: Saying Hello
Date: Fri, 21 Nov 1997 10:01:10 -0600
Message-ID: <3456@example.net>
In-Reply-To: <1234@local.machine.example>
References: <1234@local.machine.example>
```

This is a reply to your hello.

Отметим поле «Reply-To:» в приведенном выше сообщении. Когда John отвечает на приведенное выше письмо Mary, ответ следует направлять по адресу из поля «Reply-To:», а не по адресу из поля «From:».

```
-----
To: "Mary Smith: Personal Account" <smith@home.example>
From: John Doe <jdoe@machine.example>
Subject: Re: Saying Hello
Date: Fri, 21 Nov 1997 11:00:00 -0600
Message-ID: <abcd.1234@local.machine.test>
In-Reply-To: <3456@example.net>
References: <1234@local.machine.example> <3456@example.net>
```

This is a reply to your reply.

Приложение А.3. Пересылка сообщений

Начнем с сообщения, которое будет использоваться в качестве примера несколько раз:

```
-----
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>
```

This is a message just to say hello.
So, "Hello".

Mary, получив это сообщение, хочет отправить копию Jane так, что (а) сообщение выглядит, как отправленное John; (b) если Jane ответит на это сообщение, ответ должен быть отправлен John; (c) вся исходная информация, включая дату письма, изначально посланного Mary, идентификатор сообщения и исходный адрес сохраняется. В этом случае в начало сообщения добавляются поля пересылки:

```
-----
Resent-From: Mary Smith <mary@example.net>
Resent-To: Jane Brown <j-brown@other.example>
Resent-Date: Mon, 24 Nov 1997 14:22:01 -0800
Resent-Message-ID: <78910@example.net>
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.machine.example>
```

This is a message just to say hello.
So, "Hello".

Если Jane, в свою очередь, решит переслать сообщение еще кому-либо, она будет добавлять свои поля пересылки перед сообщением, показанным выше (для краткости этот вариант примера опущен).

Приложение А.4. Сообщения с полями трассировки

При пересылке сообщения через транспортную систему, как описано в [RFC5321], в начало сообщения добавляются трассировочные поля. Ниже приведен пример, показывающий, как могут выглядеть эти поля. Отметим, что первая строка «Received:» оказалась слишком длинной и в нее включены фальцовочные пробелы.

```
-----
Received: from x.y.test
  by example.net
  via TCP
  with SMTP
  id ABC12345
  for <mary@example.net>; 21 Nov 1997 10:05:43 -0600
Received: from node.example by x.y.test; 21 Nov 1997 10:01:22 -0600
From: John Doe <jdoe@node.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: Fri, 21 Nov 1997 09:55:06 -0600
Message-ID: <1234@local.node.example>
```

This is a message just to say hello.
So, "Hello".

Приложение А.5. Пробелы, комментарии и другие странности

Пробелы, включая фальцовочные, и комментарии могут вставляться между разными элементами поля. Возьмем пример из параграфа А.1.3 и добавим в него пробелы и комментарии в полях заголовка.

```

----
From: Pete (A nice \) chap) <pete(his account)@silly.test(his host)>
To:A Group (Some people)
   :Chris Jones <c@(Chris's host.)public.example>,
   :joe@example.org,
   :John <jdoe@one.test> (my dear friend); (the end of the group)
Cc:(Empty list)(start)Hidden recipients : (nobody(that I know)) ;
Date: Thu,
     13
     Feb
     1969
     23:32
     -0330 (Newfoundland Time)
Message-ID: <testabcd.1234@silly.test>

```

Testing.

Приведенный пример выглядит странно, но является совершенно допустимым. Отметим специально (1) комментарий в поле «From:» (включая скобку в паре с кватированием); (2) отсутствие пробела после двоеточия в поле «To:», а также комментарий и фальцовочные пробелы после имени группы, специальный символ (.) в комментарии поля с адресом Chris Jones и фальцовочные пробелы перед и после «joe@example.org,»; (3) множество вложенных комментариев в поле «Cc:», а также комментарий, следующий непосредственно за двоеточием после «Cc»; (4) фальцовочный пробел (но не комментарий, за исключением комментария в конце поля даты), а также отсутствие значения секунд; (5) пробелы после (но не внутри) идентификатора в поле «Message-ID:».

Приложение А.6. Устаревшие формы

Здесь приведены примеры устаревших синтаксических элементов (которые **недопустимо** создавать), описанных в разделе 4 данного документа.

Приложение А.6.1. Устаревшая адресация

Отметим в приведенном ниже примере отсутствие кавычек вокруг Joe Q. Public, маршрут в адресе Mary Smith, две запятых в поле «To:» и пробелы рябом с точкой в адресе jdoe.

```

----
From: Joe Q. Public <john.q.public@example.com>
To: Mary Smith <@node.test:mary@example.net>, , jdoe@test . example
Date: Tue, 1 Jul 2003 10:52:37 +0200
Message-ID: <5678.21-Nov-1997@example.com>

```

Hi everyone.

Приложение А.6.2. Устаревшие даты

В приведенном ниже сообщении используется устаревший формат даты, включая символьное представление часового пояса и 2-значное обозначение года. Отметим, что отсутствие дня недели не является спецификой устаревшего синтаксиса — это поле является опциональным и в современном синтаксисе.

```

----
From: John Doe <jdoe@machine.example>
To: Mary Smith <mary@example.net>
Subject: Saying Hello
Date: 21 Nov 97 09:55:06 GMT
Message-ID: <1234@local.machine.example>

```

This is a message just to say hello.
So, "Hello".

Приложение А.6.3. Устаревшие пробелы и комментарии

Наличие пробелов и комментариев в устаревшем синтаксисе разрешается в большем числе мест, нежели позволяет современный синтаксис. Однако фальцовочные строки, содержащие только пробелы, по-прежнему допускаются.

```

----
From : John Doe <jdoe@machine(comment). example>
To   : Mary Smith
-----
      <mary@example.net>
Subject : Saying Hello
Date   : Fri, 21 Nov 1997 09(comment): 55 : 06 -0600
Message-ID : <1234 @ local(blah) .machine .example>

```

This is a message just to say hello.
So, "Hello".

Особо отметим вторую строку поля «To:», начинающуюся с двух пробельных символов. (отметим, «_» представляет пробелы). Следовательно, это является рассматриваемой частью фальцовки, как описано в параграфе 4.2. Комментарии и пробелы в адресах, датах и идентификаторах сообщений являются частью устаревшего синтаксиса.

Приложение В. Отличия от ранних спецификаций

В этом приложении приведен список отличий формата сообщений Internet (IMF) от более ранних спецификаций, в частности [RFC0822], [RFC1123] и [RFC2822]. Элементы, отмеченные в списке звездочкой (*), относятся к описанным в разделе 4 данного документа устаревшим элементам, и в настоящее время они не должны создаваться.

Ниже перечислены изменения [RFC0822] и [RFC1123], внесенные в [RFC2822] и сохраненные здесь:

1. Допускается использование точки в устаревшей форме фраз.
2. Описание ABNF исключено из документа (в настоящее время оно содержится в [RFC5234]).
3. В поле года разрешается использовать 4 и более цифр.
4. Порядок полей заголовка (и отсутствие такового) указан явно.
5. Удалено шифрованное поле заголовка.
6. Явно разрешено обозначение часового пояса «-0000» и описано его значение.
7. Не разрешается использовать фальцовочные пробелы между любыми лексемами.
8. Снято требование относительно получателей.
9. Заново определены понятия пересылки (forwarding и resending).
10. Расширенные поля заголовка больше не вызываются специфически.
11. Отменено использование символа ASCII 0 (null).*
12. Строки продолжения фальцовки не могут содержать только пробелы.*
13. Не разрешается произвольная вставка комментариев в даты.*
14. Не разрешаются символьные обозначения часовых поясов.*
15. Не разрешается двухзначное представление года.*
16. Трехзначное представление года интерпретируется, но не должно использоваться.*
17. Не разрешается включать маршруты в адреса.*
18. Не разрешается использовать CFWS в локальной и доменной части адреса.*
19. Не разрешается использование пустых элементов в списках адресов.*
20. Не разрешается использование фальцовочных пробелов между именем поля и двоеточием.*
21. Не разрешается использование комментариев между именем поля и двоеточием.
22. Более жестко задан синтаксис in-reply-to и references.*
23. Не разрешается использование CFWS в msg-id.*
24. Семантика полей resent отнесена исключительно к информационной.
25. Не разрешается использовать Resent-Reply-To.*
26. Не разрешается повторение полей (за исключением resent и received).*
27. Не разрешается использование символов CR и LF по-отдельности.*
28. Задано ограничение на размер строк.
29. Разъяснено назначение поля Bcc.

Далее перечислены отличия настоящего документа от [RFC2822].

1. Исправлены найденные опечатки и приведены разъяснения.
2. Термин «стандарт» применительно к данному документу заменен на «документ» или «спецификация».
3. Разделены понятия «поле заголовка» (header field) и «раздел заголовков» (header section).
4. Удалено NO-WS-CTL из ctext, qtext, dtext и бесструктурных полей.*
5. Исправлено обсуждение специальных символов (specials) в параграфе «Atom». Текст перенесен в параграф «3.5. Общий синтаксис сообщений».
6. Упрощен синтаксис CFWS.
7. Исправлен синтаксис бесструктурных полей.
8. Изменен синтаксис полей даты и времени в части пробелов в устаревшем синтаксисе дат.
9. Удалены пары с кватированием из доменных литералов и идентификаторов сообщений.*
10. Разъяснены ограничения других спецификаций для синтаксиса доменных имен.
11. Упрощен синтаксис «Bcc:» и «Resent-Bcc:».
12. Разрешено включение дополнительного поля в трассировочную информацию.
13. Удалено no-fold-quote из msg-id. Разъяснены синтаксические ограничения.
14. Обобщен синтаксис «Received:» для исправления ошибок и удаления определения из данного документа.

15. Упрощено obs-qr. Исправлено и обобщено obs-utext (появляется не только в устаревшем синтаксисе). Удалены obs-text и obs-char, добавлено obs-body.
16. Исправлен устаревший синтаксис дат, чтобы разрешить больше (или меньше) комментариев и пробелов.
17. Исправлен синтаксис всех устаревших списков (obs-domain-list, obs-mbox-list, obs-addr-list, obs-phrase-list и вновь добавленный obs-group-list).
18. Исправлен синтаксис obs-reply-to.
19. Исправлены obs-bcc и obs-resent-bcc, чтобы разрешить пустые списки.
20. Удалено obs-path.

Приложение С. Благодарности

В создании этого документа участвовало множество людей. В их число входят участники рабочей группы Detailed Revision and Update of Messaging standards (DRUMS) под эгидой IETF, председатель DRUMS, руководители направления (Area Directors) IETF и все, кто просто присылал свои комментарии по электронной почте. Редактор документа выражает всем этим людям глубокую благодарность и искреннюю признательность. Ниже приведен список всех людей, которые присылали по электронной почте свои комментарии к данному документу и [RFC2822].

Matti Aarnio	Tanaka Akira	Russ Allbery	Eric Allman
Harald Alvestrand	Ran Atkinson	Jos Backus	Bruce Balden
Dave Barr	Alan Barrett	John Beck	J Robert von Behren
Jos den Bekker	D J Bernstein	James Berriman	Oliver Block
Norbert Bollow	Raj Bose	Antony Bowesman	Scott Bradner
Randy Bush	Tom Byrer	Bruce Campbell	Larry Campbell
W J Carpenter	Michael Chapman	Richard Clayton	Maurizio Codogno
Jim Conklin	R Kelley Cook	Nathan Coulter	Steve Coya
Mark Crispin	Dave Crocker	Matt Curtin	Michael D'Errico
Cyrus Daboo	Michael D Dean	Jutta Degener	Mark Delany
Steve Dorner	Harold A Driscoll	Michael Elkins	Frank Ellerman
Robert Elz	Johnny Eriksson	Erik E Fair	Roger Fajman
Patrik Faeltstroem	Claus Andre Faerber	Barry Finkel	Erik Forsberg
Chuck Foster	Paul Fox	Klaus M Frank	Ned Freed
Jochen Friedrich	Randall C Gellens	Sukvinder Singh Gill	Tim Goodwin
Philip Guenther	Arnt Gulbrandsen	Eric A Hall	Tony Hansen
John Hawkinson	Philip Hazel	Kai Henningsen	Robert Herriot
Paul Hethmon	Jim Hill	Alfred Hoenes	Paul E Hoffman
Steve Hole	Kari Hurтта	Marco S Hyman	Ofer Inbar
Olle Jarnefors	Kevin Johnson	Sudish Joseph	Maynard Kang
Prabhat Keni	John C Klensin	Graham Klyne	Brad Knowles
Shuhei Kobayashi	Peter Koch	Dan Kohn	Christian Kutzt
Anand Kumria	Steen Larsen	Eliot Lear	Barry Leiba
Jay Levitt	Bruce Lilly	Lars-Johan Liman	Charles Lindsey
Pete Loshin	Simon Lyall	Bill Manning	John Martin
Mark Martinec	Larry Masinter	Denis McKeon	William P McQuillan
Alexey Melnikov	Perry E Metzger	Steven Miller	S Moonesamy
Keith Moore	John Gardiner Myers	Chris Newman	John W Noerenberg
Eric Norman	Mike O'Dell	Larry Osterman	Paul Overell
Jacob Palme	Michael A Patton	Uzi Paz	Michael A Quinlan
Robert Rapplean	Eric S Raymond	Sam Roberts	Hugh Sasse
Bart Schaefer	Tom Scola	Wolfgang Segmuller	Nick Shelness
John Stanley	Einar Stefferud	Jeff Stephenson	Bernard Stern
Peter Sylvester	Mark Symons	Eric Thomas	Lee Thompson
Karel De Vriendt	Matthew Wall	Rolf Weber	Brent B Welch
Dan Wing	Jack De Winter	Gregory J Woodhouse	Greg A Woods
Kazu Yamamoto	Alain Zahn	Jamie Zawinski	Timothy S Zurcher

7. Литература

7.1. Нормативные документы

- [ANSI.X3-4.1986] American National Standards Institute, "Coded Character Set - 7-bit American Standard Code for Information Interchange", ANSI X3.4, 1986.
- [RFC1034] Mockapetris, P., "Domain names - concepts and facilities", STD 13, [RFC 1034](#), November 1987.
- [RFC1035] Mockapetris, P., "Domain names - implementation and specification", STD 13, [RFC 1035](#), November 1987.
- [RFC1123] Braden, R., "Requirements for Internet Hosts - Application and Support", STD 3, [RFC 1123](#), October 1989.
- [RFC2119] Bradner, S., "Key words for use in RFCs to Indicate Requirement Levels", BCP 14, [RFC 2119](#), March 1997.
- [RFC5234] Crocker, D. and P. Overell, "Augmented BNF for Syntax Specifications: ABNF", STD 68, [RFC 5234](#), January 2008.

7.2. Дополнительные ссылки

- [RFC0822] Crocker, D., "Standard for the format of ARPA Internet text messages", STD 11, RFC 822, August 1982.
- [RFC1305] Mills, D., "Network Time Protocol (Version 3) Specification, Implementation", [RFC 1305](#), March 1992.
- [ISO.2022.1994] International Organization for Standardization, "Information technology - Character code structure and extension techniques", ISO Standard 2022, 1994.
- [RFC2045] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part One: Format of Internet Message Bodies", RFC 2045, November 1996.
- [RFC2046] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Two: Media Types", RFC 2046, November 1996.
- [RFC2047] Moore, K., "MIME (Multipurpose Internet Mail Extensions) Part Three: Message Header Extensions for Non-ASCII Text", RFC 2047, November 1996.
- [RFC2049] Freed, N. and N. Borenstein, "Multipurpose Internet Mail Extensions (MIME) Part Five: Conformance Criteria and Examples", RFC 2049, November 1996.
- [RFC2822] Resnick, P., "Internet Message Format", [RFC 2822](#), April 2001.
- [RFC3864] Klyne, G., Nottingham, M., and J. Mogul, "Registration Procedures for Message Header Fields", BCP 90, RFC 3864, September 2004.
- [RFC4021] Klyne, G. and J. Palme, "Registration of Mail and MIME Header Fields", RFC 4021, March 2005.
- [RFC4288] Freed, N. and J. Klensin, "Media Type Specifications and Registration Procedures", BCP 13, RFC 4288, December 2005.
- [RFC4289] Freed, N. and J. Klensin, "Multipurpose Internet Mail Extensions (MIME) Part Four: Registration Procedures", BCP 13, RFC 4289, December 2005.
- [RFC5321] Klensin, J., "Simple Mail Transfer Protocol", [RFC 5321](#), October 2008.

Адрес автора

Peter W. Resnick (редактор)

Qualcomm Incorporated

5775 Morehouse Drive

San Diego, CA 92121-1714

US

Phone: +1 858 651 4478

EMail: presnick@qualcomm.com

URI: <http://www.qualcomm.com/~presnick/>

Перевод на русский язык

Николай Малых

nmalykh@protokols.ru

Полное заявление авторских прав

Copyright (C) The IETF Trust (2008).

К этому документу применимы права, лицензии и ограничения, указанные в BCP 78, и, за исключением указанного там, авторы сохраняют свои права.

Этот документ и содержащаяся в нем информация представлены "как есть" и автор, организация, которую он/она представляет или которая выступает спонсором (если таковой имеется), Internet Society и IETF отказываются от каких-либо гарантий (явных или подразумеваемых), включая (но не ограничиваясь) любые гарантии того, что использование представленной здесь информации не будет нарушать чьих-либо прав, и любые предполагаемые гарантии коммерческого использования или применимости для тех или иных задач.

Интеллектуальная собственность

IETF не принимает какой-либо позиции в отношении действительности или объема каких-либо прав интеллектуальной собственности (Intellectual Property Rights или IPR) или иных прав, которые, как может быть заявлено, относятся к реализации или использованию описанной в этом документе технологии, или степени, в которой любая лицензия, по которой права могут или не могут быть доступны, не заявляется также применение каких-либо усилий для определения таких прав. Сведения о процедурах IETF в отношении прав в документах RFC можно найти в BCP 78 и BCP 79.

Копии раскрытия IPR, предоставленные секретариату IETF, и любые гарантии доступности лицензий, а также результаты попыток получить общую лицензию или право на использование таких прав собственности разработчиками или пользователями этой спецификации, можно получить из сетевого репозитория IETF IPR по ссылке <http://www.ietf.org/ipr>.

IETF предлагает любой заинтересованной стороне обратить внимание на авторские права, патенты или использование патентов, а также иные права собственности, которые могут потребоваться для реализации этого стандарта. Информацию следует направлять в IETF по адресу ietf-ipr@ietf.org.