

Интернационализация адреса электронной почты (EAI): Техническое описание



СОДЕРЖАНИЕ

Зачем нам нужна EAI?	5
Стандарты и функциональная совместимость	5
Компоненты почтовой экосистемы	6
Экосистема программного обеспечения электронной почты	6
Пользовательский почтовый агент (MUA)	6
Веб-почта	7
Агент отправки почты (MSA)	7
Транспортный почтовый агент (MTA)	7
Агент доставки почты (MDA)	8
Хранение почты	8
Сбор почты по POP/IMAP	9
Форматы почтовых сообщений	9
Сообщения и конверты	10
Почтовые адреса	11
Интернационализованные доменные имена	12
Нормализация	12
Включение	12
Кодирование ACE	12
Алфавиты и LGR	13
Почта MIME	14
Знакомство с почтой EAI	15
Транспортировка почты EAI	15
Распознавание сообщений EAI	16
Технические вопросы почты EAI	17
Назначение и интерпретация почтовых адресов EAI	17
Хранение почты EAI	18
Агенты MUA и почта EAI	18
Адресные книги	19
Почтовые адреса в сети	19
Отправка, доставка и сбор почты EAI	20
Агенты MTA и адреса входящих сообщений вида «А-метка» и «U-метка»	21
POP и IMAP с почтой EAI	21



Поддержка EAI в POP3	22
Поддержка EAI в IMAP	23
Заголовки сообщений EAI	23
Заголовки с адресами	24
Поля заголовков, которые видны пользователям	24
Поля заголовков с данными для компьютеров	24
Заголовки Received	25
Заголовки MIME	26
Устранение ошибок доставки и информирование об ошибках доставки	27
Обратная совместимость	28
Получатели с адресами EAI и адресами прежней версии	28
Целесообразность возврата к прежней версии	21
Где возможен возврат к прежней версии почты	29
Общие способы возврата к прежней версии	30
Заголовки и тела сообщений	30
Возврат к прежней версии заголовков с адресами	30
Возврат к прежней версии других заголовков, которые видны пользователям	31
Возврат к прежней версии заголовков, которые не видны пользователям	32
Возврат к прежней версии в POP и IMAP	32
Примечания касательно возврата к прежней версии	32
Сообщения об отсутствии на работе, уведомления о прочтении и другие автоответчики	32
Проверка подлинности почты и фильтры спама	33
Аутентификация SPF	33
Аутентификация DKIM	33
Аутентификация DMARC	34
Фильтрация спама	35
Приложение А. Обработка полей заголовка сообщения	35
Приложение Б. Перечень шагов, необходимых для перехода к EAI	40
Изменения агентов MTA	40
Изменения агентов MSA	40
Изменения агентов MUA	41
Серверы IMAP	41



Серверы POP	41
Приложение В. Как работает SMTP?	41
Добавление функций UTF-8 в пример использования SMTP	43
Приложение Г. RFC, относящиеся к рассматриваемой теме	43
Важные RFC, относящиеся к почте	44
Для SMTP:	44
Для форматов почтовых сообщений:	44
Для POP и IMAP:	45
Для IDN:	45
Для почты EAI:	45
Для PRECIS и сопоставления строк:	45
Для SASL (вход и аутентификация для POP, IMAP и отправки):	46
Для URL и URI:	46
Для аутентификации сообщений:	46
Для ввода и отображения RTL и разнонаправленного текста:	46
Приложение Д. Другие источники рекомендаций	47
Глоссарий	



Это введение в интернационализацию адреса электронной почты (EAI) предназначено для технических специалистов, разработчиков приложений электронной почты и системных администраторов, желающих добавить в состав функций почтового программного обеспечения поддержку EAI. Высокоуровневое описание EAI см. в документе [UASG014: Краткое руководство по EAI](#). Для получения справочной информации об электронной почте и соответствующих технологиях см. раздел «Другие источники рекомендаций» в конце настоящего документа.

Зачем нам нужна EAI?

При создании современного интернета в 1980-х годах рабочим языком сети был английский, а для всех почтовых сообщений использовался набор символов ASCII, в состав которого входят только прописные и строчные буквы латинского алфавита без диакритических знаков.

Но почти всем остальным письменным языкам необходимы символы, которых нет в наборе ASCII. Сегодня в интернете в каждой стране мира есть пользователи электронной почты, пишущие на множестве языков, а электронная почта продолжает медленно развиваться, чтобы обеспечить обработку всех этих языков. Unicode может отобразить все эти символы. EAI позволяет использовать символы Unicode, в частности символы Unicode в кодировке UTF-8,¹ в адресах электронной почты.

Стандарты и функциональная совместимость

Интернет работает благодаря тому, что программное обеспечение на подключенных к нему компьютерах обеспечивает *функциональную совместимость*. К примеру, программа пишется так, чтобы любой компьютер мог отправить электронное письмо на любой другой компьютер, где запущена почтовая программа. Все программное обеспечение обеспечивает реализацию стандартов, полностью определяющих принципы функционирования интернета, от коммутации пакетов самого нижнего уровня до команд форматирования текстовых страниц HTML и вложения файлов.

Для функциональной совместимости программного обеспечения каждый должен реализовать стандарты одинаковым образом. В настоящем документе мы акцентируем внимание на том, как это сделать и избежать риска несовместимости. В частности, хотя часто некоторые изменения могут показаться «улучшениями», если они не соответствуют стандартам, то не позволят взаимодействовать с другими системами, что, безусловно, вовсе не будет улучшением.

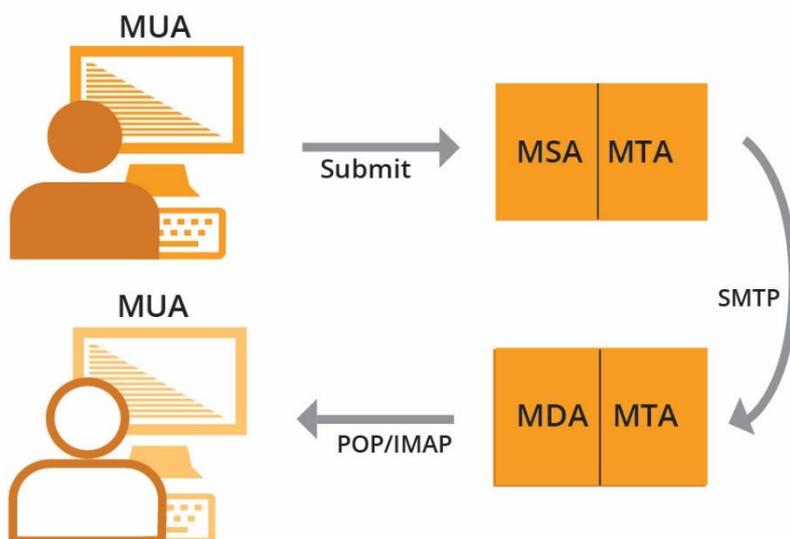
¹ UTF-8 — стандартный способ хранения кодовых точек Unicode (символов и модификаторов) в виде 8-битных байтов. В силу исторических причин часть программного обеспечения использует имя UTF8 без дефиса.

Компоненты почтовой экосистемы

Путь электронного письма от отправителя до получателя может быть довольно сложным. Чтобы упростить понимание, сообщество разработчиков электронной почты IETF приняло такую архитектуру почты, в которой используются стандартные названия различных компонентов почтовой системы и связей между ними. Эта архитектура подробно описана в стандарте [RFC 5598](#), *Архитектура почты интернета*.

В настоящем документе мы кратко рассмотрим неотъемлемые части этой архитектуры и схему взаимодействия между ними при передаче сообщения от автора к получателю.

Экосистема программного обеспечения электронной почты



В архитектуре почты интернета используются согласованные названия программ обработки почты, преимущественно обозначаемых трех- или четырехбуквенными аббревиатурами.

Пользовательский почтовый агент (MUA)

Пользователи почты читают и отправляют сообщения с помощью *пользовательского почтового агента* (MUA), единственной программы в почтовой экосистеме, с которой человек взаимодействует напрямую.

В число популярных MUA входят Outlook, Apple Mail и почтовые программы, используемые на смартфонах. Пользователь создает сообщение, возможно, применяя инструменты форматирования, по своему желанию добавляет другие вложения, а затем отправляет это сообщение с помощью протокола SUBMIT.

При получении входящей почты MUA загружает сообщения с почтового сервера (почтовых серверов), используя протоколы POP или IMAP, и отображает их для пользователя.



В обоих случаях формат сообщения, которое видит пользователь, обычно отличается от того вида, в котором сообщение отправлено или получено. Формат некоторых частей сообщения изменяется, напр., данные в поле заголовка Date: (Дата:) часто отображаются в местном формате и соответствуют часовому поясу пользователя. Содержание сообщения часто декодируется и отображается, напр., вложенный JPEG-файл будет отображаться как картинка.

Как правило, агенты MUA позволяют пользователю создавать адресные книги, где пользователь может хранить адрес (адреса) электронной почты и другую информацию о лицах, с которыми он часто переписывается. В некоторых случаях адресная книга хранится на компьютере пользователя, в то время как другие адресные книги могут использоваться совместно и быть доступными по сети. Записи в адресной книге могут быть достаточно сложными, с несколькими адресами электронной почты и другими контактными данными каждого корреспондента.

Веб-почта

В случае веб-почты почтовый клиент создается за счет объединения браузера пользователя и веб-серверов, к которым этот браузер подключается. В некоторых версиях веб-серверы используют стандартные почтовые протоколы для отправки почты на отдельные почтовые серверы и загрузки сообщений с таких серверов. В других версиях веб-сервер выполняет функции MUA или других компонентов почтовой системы. В любом случае веб-почта обеспечивает выполнение функций, аналогичных тем, которые выполнял бы отдельный MUA.

Агент отправки почты (MSA)

После того как пользователь создаст электронное письмо, он дает MUA команду отправить его для доставки. MUA подключается к *агенту отправки почты* (MSA). Затем MUA удостоверяет подлинность сообщения при помощи пароля (или косвенным образом по IP-адресу) и отправляет конверт с сообщением. Получив сообщение, MSA проверяет его и исправляет мелкие ошибки в формате сообщения, такие как отсутствие даты в поле Date: или заголовка Message-ID: (Идентификатор сообщения:). В зависимости от локальной политики он может принудительно заполнить заголовок From: (От:) данными об аутентифицированном пользователе. Кроме того, многие MSA на этом этапе добавляют подписи для криптографической аутентификации. (См. описание DKIM ниже.)

После подготовки сообщения MSA отправляет его *транспортному почтовому агенту* (MTA). В зависимости от структуры почтовой системы MSA и MTA могут быть отдельными программами или просто отдельными функциями одной программы.

Протокол взаимодействия MUA и MSA похож на SMTP (Simple Mail Transport Protocol — простой протокол передачи почты), который используется при обмене данными между MTA, но имеет несколько существенных отличий. Иногда его называют SUBMIT, чтобы отличить от SMTP.

Транспортный почтовый агент (MTA)

Транспортный почтовый агент (MTA) — это то, что чаще всего имеют в виду, когда говорят «почтовый сервер». Он получает полностью обработанное MSA сообщение, а затем определяет для каждого получателя, как и куда доставить это сообщение. Как правило, для каждого адреса получателя MTA сначала находит в адресе домен, чтобы



определить, обрабатывается ли этот домен MTA локально. Если нет, в DNS выполняется поиск записей **MX** (адресов почтового шлюза), чтобы обнаружить хост-компьютеры, которые выполняют такую обработку, создается подключение по SMTP к одному из этих компьютеров и сообщение отправляется.

В наиболее простых почтовых системах хост, указанный в списке MX, самостоятельно выполняет доставку почты, в то время как в более сложных системах сообщение может несколько раз передаваться с одного компьютера на другой.

Операция SMTP — это последовательность команд отправляющей системы и ответов принимающей системы. Первая команда и ответ устанавливают связь и позволяют получить список расширений SMTP, которые поддерживаются принимающей системой, таких как расширение SMTPUTF8, обеспечивающее обработку почты EAI. В последующих командах передаются данные конверта (обратный адрес и адреса получателей), а затем само сообщение.

Дополнительные сведения о SMTP см. в Приложении В: Как работает SMTP?

Агент доставки почты (MDA)

После того как сообщение пройдет весь путь до MTA получателя, оно отправляется агенту доставки почты (MDA) получателя для доставки. В простейшем случае MDA просто сохраняет сообщение как файл для последующей загрузки, но в большинстве случаев на этапе доставки выполняется некоторая обработка.

IETF разработала язык **SIEVE**, позволяющий пользователям почты писать сценарии, которые определяют процедуры обработки на этапе доставки, такие как сортировка почты по отдельным ящикам или пересылка на другие адреса. Существует множество других программ доставки, например, широко используемая procmail. Большинство систем Unix и Linux дают пользователям возможность писать произвольные программы для обработки входящей почты.

В системах веб-почты предусмотрены различные виды обработки на этапе доставки. Большинство систем позволяет пользователям сортировать почту в зависимости от найденного в сообщении текста (напр., если в теме письма есть слово «pickle-list», письмо сохраняется в папке с именем «pickle-list»), в то время как другие группируют почту по категориям, таким как «Новости» или «Форумы».

Хранение почты

Первоначально почтовый ящик электронной почты был аналогом обычного почтового ящика, то есть местом временного хранения входящих почтовых сообщений до их загрузки и обработки. По мере того как компьютерная память становилась быстрее и дешевле, размер почтовых ящиков вырос настолько, что теперь хранение корреспонденции за несколько лет — обычное явление. Как правило, в почтовых системах есть несколько папок для каждого почтового ящика, каждая из которых поименована в рамках иерархии, как и файлы на диске. Сообщениям могут присваиваться метки «не прочитано», «прочитано», «важно» или «отвечено». В почтовых программах обычно предусмотрены способы организации почты и поиска сообщений разными способами, по ключевым словам или адресам, а также в пределах ограниченного интервала дат.



Почтовые системы используют многочисленные внутренние схемы хранения. В простейшем случае вся почта в почтовом ящике хранится в единственном файле. В других каждое сообщение хранится в отдельном файле. Кроме того, почта может храниться в базе данных, при этом данные заголовков и информация в теле сообщения могут обрабатываться по-разному. Независимо от схемы хранения, агенты MUA рассматривают почту как список сообщений в каждой папке.

Сбор почты по POP/IMAP

В первых почтовых системах пользователи хранили свою электронную почту и запускали почтовые программы на одном и том же компьютере, при этом почтовая программа считывала письма непосредственно из локальных файлов. В наши дни пользователи постоянно читают почту на разных компьютерах, используя для ее загрузки протоколы POP и IMAP.

POP (Post Office Protocol — почтовый протокол), более старый и простой протокол, позволяет MUA загружать почту с почтового сервера и затем, по желанию, удалять ее с сервера. При этом MUA управляет почтой на собственном компьютере пользователя.

IMAP (Internet Message Access Protocol — протокол доступа к сообщениям в интернете) позволяет почтовой программе пользователя управлять почтовыми ящиками на сервере, копируя почту с сервера и на сервер по мере необходимости. Поскольку почта обычно остается на сервере, IMAP позволяет использовать для просмотра содержимого почтового ящика почтовые программы, установленные на разных компьютерах, например, ноутбуках и телефонах. Так как почтовой программе не нужно загружать всю почту, можно хранить огромный объем корреспонденции, запуская почтовые программы на небольших компьютерах, таких как планшеты и телефоны.

Еще одним важным отличием POP от IMAP является то, что POP видит только основной ящик для входящих писем учетной записи электронной почты, в то время как IMAP способен управлять всеми папками учетной записи. Предполагалось, что пользователи POP будут копировать всю почту на свой компьютер и управлять ей там.

Как POP, так и IMAP требуют аутентификации пользователя по пользовательскому имени и паролю перед загрузкой почты. В большинстве случаев именем пользователя POP/IMAP является адрес электронной почты этого пользователя.

Форматы почтовых сообщений

Основной формат интернет-сообщения первоначально был определен стандартом [RFC 724](#) в 1977 году и за последующие 40 лет претерпел очень мало изменений при неоднократном обновлении спецификации. Последним документом в этой серии является [RFC 5822](#). Указанный формат оказался достаточно гибким для удовлетворения меняющихся с течением лет потребностей пользователей почты.

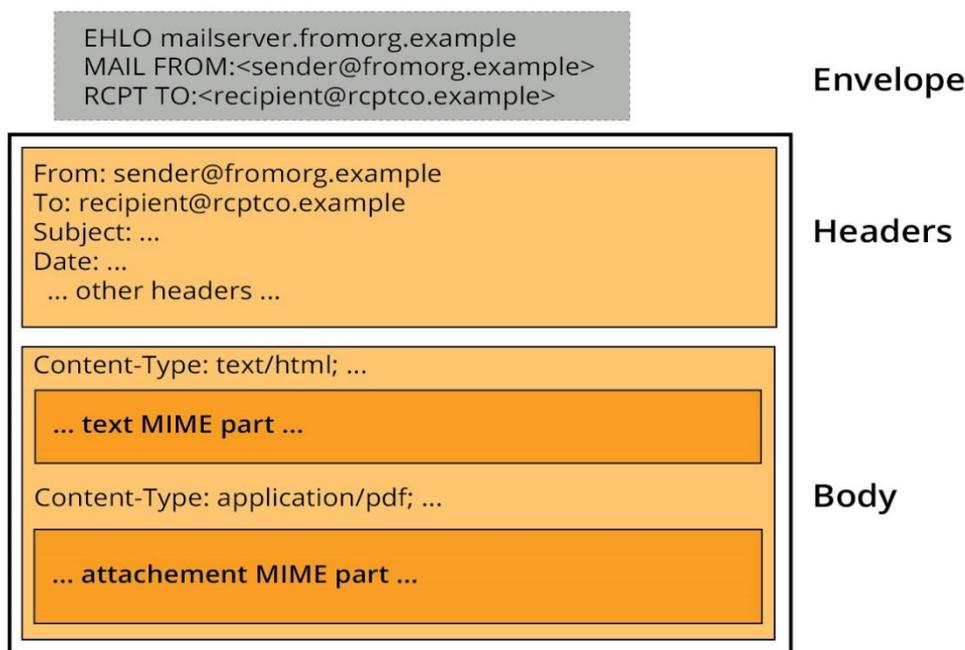
Любое сообщение прежней версии, то есть сообщение до EAI, представляет собой последовательность строк текста ASCII, заканчивающихся символами возврата каретки и перевода строки ASCII (CR и LF). Первая часть сообщения, которая обычно называется заголовком, это ряд структурированных полей заголовка, за которыми идет пустая строка и в конце неструктурированное тело сообщения. Каждое поле заголовка начинается с имени заголовка, например From: (От:) или Subject:



(Тема:) и двоеточия, за которым следует содержание заголовка. Содержание некоторых полей заголовка, таких как заголовок Subject: (Тема:) имеет произвольный формат, в то время как для других, таких как заголовки Date: (Дата:) и Message-ID: (Идентификатор сообщения:) установлен абсолютно фиксированный формат. При этом в ряде заголовков используется сочетание фиксированного и произвольного форматов, например, это заголовки To: (Кому:), From: (От:) и Cc: (Копия:), где адреса фиксированного формата сочетаются с произвольным текстом комментария.

Сообщения в обязательном порядке сопровождаются метаданными, которые относятся к сообщению, но не являются его частью. Важной частью метаданных является описанный в следующем разделе конверт. Другими компонентами метаданных могут быть дата получения сообщения, а также информация о том, прочитал ли получатель сообщение и пометил ли его как важное или нежелательное.

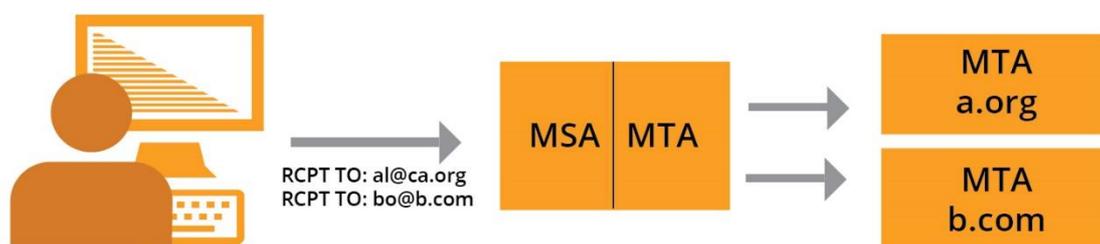
Сообщения и конверты



К транспортируемым сообщениям электронной почты, от момента отправки до момента доставки, прилагается *конверт*. Этот конверт содержит *обратный адрес* для сообщений об ошибках (он часто называется «адресом возврата»), *адреса получателей*, которым сообщение должно быть доставлено, и иногда другие данные о процессе доставки. Данные конверта отправляются параллельно с сообщением, но отдельно от него. При передаче сообщения по SMTP (которая подробнее описана ниже) одна из команд SMTP отправляет обратный адрес, дополнительные команды SMTP отправляют адрес каждого получателя, и затем еще одна команда SMTP отправляет сообщение.



При отправке сообщения обратный адрес в конверте обычно совпадает с адресом From: (От:) в сообщении, а адреса получателей в конверте совпадают с адресами в полях To: (Кому:), Сс: (Копия:) и Всс: (Скрытая копия:). Однако адреса в конвертах не обязательно должны совпадать с адресами в сообщении, и во многих распространенных случаях они не совпадают. Например, если сообщение отправляется на два адреса, Алисе и Бобу, использующим разные почтовые системы, в экземпляре, отправленном Алисе, адресом получателя будет только ее адрес, а в экземпляре, отправленном Бобу, адресом получателя будет только его адрес. При отправке сообщения подписчикам на лист рассылки, в строке адреса To: (Кому:) обычно указывается адрес листа рассылки, но программное обеспечение этого листа разошлет сообщение всем подписчикам. Это означает, что указанный в конверте сообщения список получателей будет содержать адреса всех подписчиков.



Помимо адресов в конверте могут быть указаны другие параметры, такие как размер сообщения или идентификационные данные отправителя нового сообщения. Как станет понятно далее, в почте EAI конверт указывает на то, является ли письмо сообщением EAI или сообщением традиционного формата.

Почтовые адреса

Любой адрес электронной почты прежней версии имеет вид *локальная-часть@доменноеимя*. Локальная-часть — произвольный текст ASCII, а доменное имя — имя хоста DNS. Имя хоста в прежней версии представляет собой последовательность разделенных точками меток. Каждая метка начинается и заканчивается буквой или цифрой ASCII и состоит только из букв, цифр и дефисов ASCII. Это называется правилом LDH.

В принципе, локальная часть прежней версии может содержать любые печатные символы ASCII, в том числе пробелы и знаки пунктуации. На практике имена с экзотической пунктуацией сложно набирать на клавиатуре, и они часто обрабатываются с ошибками при вводе в веб-формы.

В заголовках почтовых сообщений адреса обычно заключены в угловые скобки, чтобы отделить их от текста комментариев, напр.:

```
To: Jim Smith <james@example.com>, Mary Jones <mzjones@example.net>
```

Устаревшей, но все еще допустимой синтаксической структурой являются комментарии в круглых скобках, напр.,

```
To: james@example.com (Jim Smith), mzjones@example.net (Mary Jones)
```



Интернационализованные доменные имена

Интернет-пользователи хотят использовать алфавит родного языка не только в теле электронных писем, но и в доменных именах. Хотя в коде системы доменных имен (DNS) используются восьмибитные байты и, в принципе, имеется возможность использовать имена из букв UTF-8, огромное количество программ, поддерживающих DNS, способно обрабатывать только ASCII. Кроме того, Unicode часто допускает несколько разных способов создания одного и того же символа, напр., написание символа с надстрочным знаком «é» либо в виде двух кодовых точек, обычной буквы «е» и отдельного символа надстрочного знака «´», либо в виде одной композитной кодовой точки — буквы с надстрочным знаком. Операции поиска по DNS выполняются только для точных совпадений. Таким образом, если имя хранится в одном виде, но ищется в другом, при поиске возникнет ошибка.

Сообщество DNS решило эту проблему путем создания системы, которая называется «Интернационализованные доменные имена в приложениях (IDNA)». В ней применяется ряд способов, обеспечивающих достаточно эффективное функционирование имен UTF-8. Для EAI используется самая последняя версия IDNA, IDNA2008.

Нормализация

Первый способ — нормализация за счет единообразного представления символов, которые в UTF-8 могут быть представлены несколькими способами, напр., как одна кодовая точка «é» или как «е» и надстрочный знак «´». Консорциум Unicode определил несколько форм нормализации. Для IDN-доменов используется [Форма нормализации С](#) (NFC), где буква С является сокращением слова «композиция». Грубо говоря, используются композитные символы, которые включают в себя диакритические знаки и другие модификаторы, а не отдельные кодовые точки букв и модификаторов.

Включение

Второй способ это *включение*. В соответствии с этим принципом конкретные символы обдуманно включаются в допустимый набор, вместо того чтобы начать с включения всех символов и в дальнейшем пытаться исключить те, которые создают проблемы. В [спецификациях IDNA](#) определены кодовые точки, разрешенные к применению в доменных именах. Как правило, это буквы и цифры, используемые в широком спектре языков, а также — в ряде случаев — дополнительные символы, необходимые для их объединения или форматирования. Все остальные кодовые точки Unicode, такие как знаки пунктуации и эмодзи, сознательно исключены. Строка UTF-8, соответствующая всем правилам IDNA, называется U-меткой.

Кодирование ACE

Третий способ — ASCII-совместимое кодирование (ACE). Поскольку значительная часть существующего программного обеспечения (не только программного обеспечения электронной почты) обрабатывает только символы ASCII, в спецификациях IDNA создан альтернативный вариант кодирования, совместимый с ASCII, который называется A-меткой. Каждая допустимая U-метка соответствует уникальной A-метке и наоборот. A-метка хранится в DNS, в то время как U-метка обычно вводится пользователями и отображается для них.



A-метка имеет вид *xn--текст*, где *xn--* — префикс, указывающий на то, что это A-метка, а текст — закодированный с использованием схемы Punycode ASCII-совместимый вариант U-метки. IDNA широко поддерживается веб-браузерами, которые перед поиском преобразуют имена UTF-8 в своих адресных строках в A-метки.

Алфавиты и LGR

Четвертый способ заключается в использовании алфавитов и правил генерирования меток (LGR). Алфавит — это совокупность символов, используемых в языке для письма. Взаимосвязь между алфавитами и языками может быть сложной. Некоторые алфавиты, например кириллица, используются в нескольких языках, в то время как в некоторых языках, например в японском, для письма используется несколько алфавитов. Правила генерирования меток определяют допустимый для конкретного языка набор символов Unicode. Основная цель LGR — избежать смешения алфавитов в именах, чтобы не вводить пользователей в заблуждение (напр., избежать смешения латинских букв и кириллицы), и определить вариантные символы, отличающиеся внешне, но одинаковые по значению.

LGR официально не являются частью IDNA2008, но стандарты IETF рекомендуют их использовать.

У большинства регистратур доменов верхнего уровня есть LGR для всех языков, на которых они разрешают регистрацию доменных имен. IANA поддерживает репозиторий этих правил по адресу: <https://www.iana.org/domains/idn-tables>.

В спецификации UTS#39, Механизмы безопасности Unicode, даются рекомендации по ограничению смешения алфавитов. «Очень высокий» уровень ограничения, определение которого дано в этой спецификации, является консервативным и широко используемым правилом создания идентификаторов.

Часть адреса электронной почты после символа @ представляет собой доменное имя, поэтому здесь допускается использование A-меток. Однако часть адреса электронной почты до символа @, локальная часть, не является доменным именем, и по многим причинам локальную часть UTF-8 невозможно закодировать с помощью Punycode или в виде A-метки.² Для надлежащей интернационализации электронной почты требуется расширение почтовой системы, способное обрабатывать UTF-8 в локальной части адреса, а также в других уголках и закоулках, не охваченных расширениями MIME.

² Поскольку почтовые программы принимают и отображают локальные части как есть, пользователи вместо правильного адреса UTF-8 увидели бы лишенный смысла Punycode или A-метку. Хуже того, при наборе адреса пользователям пришлось бы вводить символы в формате Punycode или A-метки.



Почта MIME

MIME (Multipurpose Internet Mail Extensions — многоцелевые расширения интернет-почты), первая крупная веха в истории развития почты, появились в 1992 году. До MIME тело почтового сообщения было просто блоком неструктурированного ASCII-текста. MIME позволили рассматривать тело почтового сообщения как группу структурированных блоков данных, часть которых может быть текстом, в то время как другие — изображениями, документами или любым другим видом данных, которые можно хранить в файле. MIME обеспечивают наличие стандартных способов кодирования любого вида данных. Когда в состав почтового сообщения включаются вложения, изображения или другие файлы, для этого используются MIME.

MIME выполняют наложение структуры частей сообщения на блок неструктурированного текста. Сообщение может состоять из одной части или нескольких частей, связанных друг с другом различными способами. При этом у каждой части может быть определенный тип. Характерные примеры типов: `text/plain` и `text/html` для обычного текста и текста в формате HTML, или `application/pdf` для вложенного PDF-файла. Сообщение из нескольких частей может иметь тип `multipart/alternative`, различные варианты данных одного и того же вида (как правило, неформатированный и форматированный текст), или `multipart/related`, как правило, текст и вложения.

Для каждой текстовой части MIME может быть указан набор символов, например UTF-8, что позволяет даже в почте прежней версии использовать в теле сообщения UTF-8.

Кроме того, MIME добавили функцию 8BITMIME, позволяющую передавать сообщения с 8-битными символами в теле, если тело сообщения все еще является текстовыми строками с символами CR и LF в конце. Это дает возможность использовать текст UTF-8, поскольку все символы ASCII, в том числе CR и LF, также являются символами UTF-8. В принципе, поддержка 8BITMIME почтовыми серверами осуществляется опционально, но на практике все современные почтовые серверы поддерживают данную функцию. В сообщениях также можно использовать кодирование *quoted-printable* или *base64*, для отображения любых данных, в том числе UTF-8, в сообщениях прежней версии.

MIME также позволяют указывать закодированные наборы символов в неформатированных частях заголовков сообщений, таких как строка Subject: (Тема:), используя те же наборы символов и кодировок, которые разрешены в теле сообщений. Они называются закодированными словами.

Например, эта строка темы — закодированное слово в формате UTF-8. В этом примере используется шестнадцатеричный символ UTF-8 C2 B0, приподнятая строчная буква «o».

Subject: =?utf-8?Q?Your_reservation_N=C2=B0_F39-04XS?=

Тема: Your reservation N° F39-04XS

Закодированные слова MIME можно использовать в большинстве почтовых заголовков, которые видны пользователям, включая Subject: (Тема:) и предназначенные для комментариев части заголовков адресов, таких как To: (Кому:) и From: (От:), но нельзя использовать в адресах электронной почты прежней версии, которые по-прежнему должны иметь кодировку ASCII.



Следует подчеркнуть этот последний момент: *за исключением адресов, практически везде в почтовом сообщении уже можно использовать UTF-8 через MIME.*

Знакомство с почтой EAI

Технические различия между прежней версией и почтой EAI весьма невелики. Наиболее важное отличие в том, что адреса электронной почты (локальная часть и/или доменное имя) могут содержать UTF-8 наряду с ASCII. Локальная часть адреса может содержать любые печатные символы UTF-8. В доменном имени могут использоваться только те символы, которые разрешены IDNA2008.

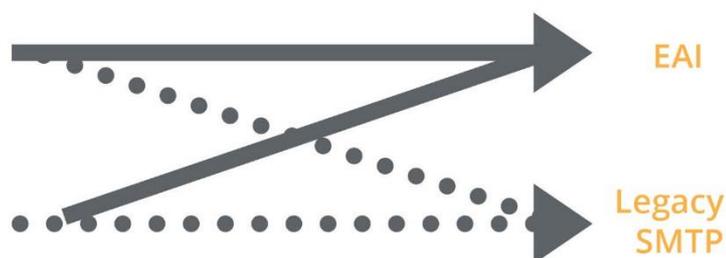
Другое отличие состоит в том, что большинство частей сообщения может быть в кодировке UTF-8, а не ASCII. Содержание полей заголовка, таких как To: (Кому:), From: (От:) и Subject: (Тема:) может иметь кодировку UTF-8, хотя в именах полей по-прежнему должны использоваться символы ASCII. Содержание сообщения может быть в кодировке UTF-8 без необходимости прибегать к особым приемам кодирования.

Есть еще несколько отличий, но эти самые важные. Поскольку почтовая система прежней версии не может обрабатывать адреса UTF-8 в конверте сообщения или адреса UTF-8 в почтовых заголовках, в процессе EAI потребовалось изменить и SMTP, и форматы сообщений для обеспечения возможности обработки новых видов адресов.

Транспортировка почты EAI

EAI создает концептуально параллельный почтовый поток для почты EAI. Когда один компьютер передает сообщение EAI другому компьютеру, например, когда MUA отправляет сообщение MSA или MTA передает сообщение другому MTA, необходимо убедиться, что принимающая система сможет обработать это сообщение. (Мы говорим «концептуально параллельный», так как любой почтовый сервер EAI может поддерживать почту прежней версии.)

Для стандартной отправки MUA→MTA и MTA→MTA в протоколе SMTP определено функциональное расширение SMTPUTF8. Если почтовый сервер поддерживает SMTPUTF8, он может обрабатывать почту EAI; если нет, то не может. Когда отправляющий компьютер устанавливает связь с принимающим, последний отправляет список поддерживаемых им расширений (см. EHLO в Приложении В ниже), и отправляющий компьютер проверяет, есть ли в этом списке SMTPUTF8.



На этом рисунке показаны два концептуально разделенных почтовых потока, EAI и SMTP прежней версии. Сплошная диагональная линия указывает на то, что любой почтовый сервер EAI также является сервером почты прежней версии, то есть любой отправитель почты прежней версии может отправить ее через сервер EAI на адрес прежней версии. Пунктирная диагональная линия указывает на то, что в некоторых (немногочисленных) случаях, почту EAI можно преобразовать в почту прежней версии.

Распознавание сообщений EAI

Поскольку параллельные почтовые потоки EAI и почты прежней версии скорее всего будут существовать еще довольно долго, иногда потребуется различать сообщения EAI и сообщения прежней версии. Это можно сделать двумя способами.

Первым является анализ сообщения, которое нужно идентифицировать как EAI. Если символы UTF-8 есть в каком-либо из адресов конверта или в каком-либо из заголовков сообщения, то это сообщение EAI.³ Кроме того, конец заголовка обозначен пустой строкой, закодированной как четыре символа: CR LF CR LF. Такой процесс идентификации мог бы оказаться медленным при наличии у сообщения больших заголовков или при необходимости считывания сообщения с дискового файла.

Другой способ различить сообщение EAI — хранить его статус EAI в составе метаданных сообщения. В состав метаданных уже входят адреса в конверте и — во многих системах — другие данные, например, время получения и сведения о том, прочитано ли сообщение, отправлен ли ответ и признано ли сообщение нежелательным. В некоторых случаях можно было бы без труда включить еще один индикатор статуса EAI.

³ Его легко реализовать в программном обеспечении, поскольку коды всех символов UTF-8 больше 0x80, в то время как коды ASCII меньше 0x80.



Технические вопросы почты EAI

Большинство изменений относится к сеансу SMTP и заголовкам сообщения, но также есть несколько изменений в пользовательских почтовых программах, и ряд новых опций для тела сообщения.

Назначение и интерпретация почтовых адресов EAI

Многие соглашения для адресов ASCII сохраняют силу для почтовых адресов EAI: локальные части адресов EAI могут содержать любые печатные символы UTF-8, в то время как доменные части остаются доменными именами, но при этом имена IDN-доменов можно вводить как U-метки UTF-8.

Любой адрес с UTF-8 в локальной части, доменной части или в обеих частях — это адрес EAI. Хотя в стандартах электронной почты указано, что локальная часть может содержать любую последовательность символов UTF-8 длиной до 64 байтов, одни локальные части пользователям намного проще запомнить и ввести, чем другие.

Рабочая группа IETF PRECIS⁴ (Preparation and Comparison of Internationalized Strings — подготовка и сравнение интернационализированных строк) разработала стандартные способы обработки ввода строковых данных, используемых в приложениях. Для различных областей применения определены классы строк. Класс *Identifier* (*Идентификатор*) предназначен, как предполагает его название, для строк, используемых в качестве идентификаторов. У этого класса есть профиль *UsernameCaseMapped* который достаточно хорошо сопоставляет имена почтовых ящиков, в том числе нечувствительные к регистру буквы и цифры, но не пробелы или знаки пунктуации. Группа PRECIS также сформулировала правила подготовки для предварительной обработки пользовательского ввода, чтобы разобраться с несколькими вариантами символов, такими как буквы верхнего и нижнего регистров или полной и половинной ширины.

Дополнительное преимущество использования профиля Username (Имя пользователя) заключается в том, что интернационализированные имена пользователей для серверов POP и IMAP (описание представлено ниже) должны быть именами пользователей PRECIS, что позволяет использовать пользовательский почтовый ящик как имя пользователя POP/IMAP. См. список RFC в Приложении Г, где представлены ссылки на подробные спецификации PRECIS.

Консорциум Unicode дает аналогичные рекомендации относительно адресов электронной почты в разделе 3.3 документа [UTS#39 Механизмы безопасности Unicode](#).

В некоторых случаях может принести пользу присвоение почтовому ящику одновременно адреса EAI и адреса прежней версии. (См. раздел «Возврат к прежней версии» ниже.) В некоторых случаях возможна прямая транслитерация, например борис@domain в boris@domain или 李伟@domain в liwei@domain. В других случаях

⁴ Произносится как французское *précis*, примерно как «преси».



естественный способ транслитерации может отсутствовать, и между двумя именами не будет очевидной связи.

Стандарты электронной почты требуют, чтобы системы относились к адресам почтовых ящиков в других системах как к непрозрачным идентификаторам, то есть передавали их без изменений, не пытаясь интерпретировать или изменить. Это означает, что системы не должны даже пытаться нормализовать текст согласно NFC, поскольку нет гарантии того, что принимающая система использует именно этот алгоритм нормализации или вообще использует нормализацию.

Однако стандарты предоставляют системе доставки широкую свободу действий в плане обработки почты, адресованной ее собственным пользователям. Что касается адресов прежней версии, символы ASCII верхнего и нижнего регистров, как правило, считаются одинаковыми, и часто системы принимают варианты адреса с незначительными изменениями, например, адреса с точками или без точек, с апострофами или другими знаками пунктуации (к примеру, почта, отправленная на адреса J.O'Brian@, J.obrian@ и jobrian@ может быть доставлена в один и тот же ящик).

В почтовых системах EAI возможна аналогичная гибкость в отношении адресов UTF-8. Рекомендации по подготовке и нормализации вводимых строковых данных см. в документах PRECIS.

Хранение почты EAI

Традиционная структура папок, используемая для почты прежней версии, не меняется при EAI, однако теперь имена папок могут иметь кодировку UTF-8, и метаданные, связанные с сообщениями, например адреса To: (Кому:) и From: (От:) как и заголовки Subject: (Тема:), тоже могут иметь кодировку UTF-8.

Класс PRECIS *FreeFormText*, определенный в стандарте [RFC 8264](#), предназначен для строк произвольной формы, таких как «удобные для восприятия псевдонимы», которые могут быть полезны в качестве имен папок. Этот класс допускает использование более широкого спектра текстовых символов по сравнению с классом Identifier, в том числе пробелов и некоторых знаков пунктуации.

Агенты MUA и почта EAI

Необходимые для обработки почты EAI изменения агентов MUA невелики.

Везде, где пользователь может ввести почтовый адрес прежней версии, например в заголовках To: (Кому:) и Cc: (Копия:) и в записях адресной книги, любая проверка адресов во время ввода должна допускать возможность использования адресов EAI.

Адреса EAI могут содержать текст, имеющий направление *слева-направо (LTR)*, *справа-налево (RTL)* или *двунаправленный (bidi)*, например, если у адреса есть локальная часть RTL и доменное имя ASCII. Агенты MUA должны быть готовы к отображению таких адресов, когда они встречаются в полях заголовка или в теле сообщения. Рассмотрение наилучших способов отображения двунаправленного текста выходит за рамки настоящего документа. Справочную информацию см. в Приложении Г.

Когда к введенному пользователем тексту сообщений агентом MUA добавляются ссылки, в процессе создания ссылок все доменные имена и все адреса электронной почты должны обрабатываться одинаково.



Адресные книги

Адресные книги дают возможность указать, готов ли определенный адрес к получению почты EAI. Хотя любые адреса EAI позволяют получать почту EAI, адреса прежней версии могут как предоставлять, так и не предоставлять такую возможность. При получении пользователем сообщения EAI можно указать, что отправитель такого сообщения готов к получению EAI.

Если у контакта в адресной книге есть и адрес EAI, и адрес прежней версии, скорее всего почту EAI желательно отправлять на адрес EAI, хотя у пользователя должна быть возможность окончательного выбора.

Почтовые адреса в сети

Многие сайты принимают адреса электронной почты через веб-формы, чтобы использовать эти адреса для отправки подтверждений, добавления в листы рассылки и для других целей. Проверка адресов, как правило, выполняется при помощи сценария Javascript на веб-странице или программы на сервере, куда отправляется форма. Javascript обычно использует регулярные выражения для проверки синтаксиса адреса. Достаточно полный набор выражений для адресов представлен в интернет-проекте IETF [draft-seantek-mail-regexen](#).

При получении адреса сервер должен выполнить несколько проверок помимо тех, которые могут быть выполнены Javascript. Если доменная часть адреса имеет кодировку UTF-8, необходимо нормализовать ее в соответствии с правилами IDNA2008 и затем убедиться, что домены являются допустимыми U-метками, преобразовав их в A-метки. Если при преобразовании возникает ошибка, то адрес недопустимый и пользователь должен его исправить. Когда у сервера будет вариант домена в виде A-метки, он сможет выполнить поиск домена в DNS, чтобы убедиться в его существовании и наличии у этого домена DNS-записей MX, A или AAAA. Если домен не существует или у него нет записей MX, A или AAAA, он непригоден для отправки почты.

Нет иного способа проверки локальной части адреса кроме отправки письма на этот адрес и выяснения того, получил ли пользователь это сообщение. (В частности, организация неполного сеанса SMTP для оценки успешности выполнения команды RCPT TO:, не является надежной проверкой.) Отправка подтверждающего сообщения для проверки работоспособности адреса и личности адресата — передовая практика работы с электронной почтой, независимо от того, является ли адрес EAI.⁵

⁵ См. [M³AAWG Распространенная передовая практика отправителей сообщений, версия 3.0](#).



Почтовые адреса могут встречаться в поле `mailto:` URI в форматированном тексте. Если в поле `mailto:` URI находится адрес EAI, то в соответствии с требованиями спецификаций URI⁶ все символы URI, не входящие в набор ASCII, должны быть закодированы с использованием знака процента. В случае адреса 猫王@普遍接受-测试.世界 поле `mailto:` URI имело бы следующий вид:

```
mailto:%e7%8c%ab%e7%8e%8b@%e6%99%ae%e9%81%8d%e6%8e%a5%e5%8f%97%2d%e6%b5%8b%e8%af%95.%e4%b8%96%e7%95%8c
```

Китайские иероглифы закодированы с использованием знака процента, а точка и коммерческое «эт» ASCII нет. Доменное имя может быть U-меткой или эквивалентной A-меткой, напр.:

```
mailto:%e7%8c%ab%e7%8e%8b@xn----f38am99bqvcd5liy1cxsg.xn--rhqv96g
```

Отправка, доставка и сбор почты EAI

В нижеследующем разделе предполагается, что читатель уже знаком с принципами работы протоколов SMTP и SUBMIT. См. обзор SMTP в [Приложении В](#).

Когда MTA использует SMTP для отправки сообщения другому MTA, или когда MUA использует SUBMIT для отправки сообщения MSA, в протоколы SMTP и SUBMIT вносятся несколько изменений. EAI вносит аналогичные изменения в SMTP и SUBMIT, поэтому такие изменения будут рассмотрены далее совместно.

Первое отличие состоит в том, что принимающий компьютер включает ключевое слово `SMTPUTF8` в список ключевых слов расширений, чтобы сообщить о том, что он может обрабатывать почту EAI:

```
S: <connect>
R: 220 receive.net ESMTP
S: EHLO sender.org
R: 250-8BITMIME
R: 250-SMTPUTF8
R: 250 PIPELINING
```

Порядок ключевых слов не имеет значения, но слово `SMTPUTF8` должно присутствовать в списке. (Каждый почтовый сервер EAI также должен поддерживать `8BITMIME`, но в любом случае все современные сервера обеспечивают такую поддержку.)

Помимо этого, при отправке обратного адреса команда `MAIL FROM:` содержит ключевое слово `SMTPUTF8`, которое указывает, что это сообщение EAI. Адреса UTF-8 отправляются без изменений.

```
S: MAIL FROM:<猫王@普遍接受-测试.世界> SMTPUTF8
R: 250 Sender accepted
```

⁶ Определены в RFC 3987.



Если тело сообщения содержит 8-битные данные, такие как UTF-8, отправитель может указать это в команде MAIL FROM, хотя это не обязательно.

```
S: MAIL FROM: <猫王@普遍接受-测试.世界> SMTPUTF8 BODY=8BITMIME
```

```
R: 250 Sender accepted
```

Остальная часть почтового сеанса не меняется по сравнению с почтой прежней версии, за исключением того, что в адресах получателей в конверте и в заголовках сообщения разрешено использовать UTF-8.

Если отправляющий компьютер пытается отправить сообщение EAI, а в полученном от сервера ответе EHLO отсутствует SMTPUTF8, этот сервер не может обрабатывать почту EAI и возникает ошибка доставки. Аналогичным образом, если принимающий компьютер отклоняет команду MAIL FROM, которая содержит ключевое слово SMTPUTF8, возникает ошибка доставки.

Агенты МТА и адреса входящих сообщений вида «А-метка» и «U-метка»

МТА обычно обрабатывает почту для нескольких конкретных доменов.⁷ Для тех доменов, чья почта принимается, почтовый сервер EAI должен принимать сообщения и с А-меткой, и с U-меткой. Хотя отправителям EAI следует использовать U-метку, отправители прежней версии будут использовать А-метку. Кроме того, вполне возможно, что имеющие неправильную конфигурацию отправители EAI тоже будут использовать А-метку. После того как сообщение получено, к адресу должен применяться одинаковый подход, независимо от того, был ли этот адрес А-меткой или U-меткой. Простым способом сделать это является нормализация адресов входящих сообщений, чтобы перед последующей обработкой все они становились А-метками или U-метками.

POP и IMAP с почтой EAI

Серверы POP и IMAP, поддерживающие EAI, нуждаются в двух видах изменений.

Первое изменение разрешает использование UTF-8 в именах пользователей и паролях, а также в именах папок IMAP и строках поиска. В каждом случае необходимо, чтобы сервер мог сообщить клиентам о том, что поддерживает эти новые функции.

⁷ Некоторые серверы одновременно являются МТА и MSA. Если выполняется аутентификация отправителя перед отправкой почты, то сервер используется как MSA. Таким образом, он будет принимать и ретранслировать почту на адреса в любых доменах. МТА, который принимает и ретранслирует почту для любого домена без необходимости аутентификации, называется *открытым ретранслятором* и скорее всего станет объектом злоупотреблений со стороны спаммеров и будет включен в черный список.



Второе изменение позволяет клиентам POP получать сообщения EAI, а клиентам IMAP получать сообщения и вновь сохранять их на сервере. Для этого необходимо, чтобы клиент сообщил серверу, что может обрабатывать сообщения EAI.

Оба протокола, POP и IMAP, поддерживают множество способов отправки учетных данных для входа. В каждом протоколе предусмотрена команда отправки имени пользователя и пароля в виде незашифрованного текста, а также более сложная команда аутентификации, позволяющая использовать многообразные схемы обеспечения безопасности, в совокупности именуемые SASL (Simple Authentication and Security Layer — простой уровень аутентификации и безопасности).⁸

Поддержка EAI в POP3

Протокол POP3 дает возможность клиентским MUA получать почту с почтовых серверов и должен обеспечивать простоту реализации как на серверах, так и на клиентских устройствах.

В протоколе POP3 используется последовательность четырехбуквенных команд и простых ответов. Клиенты используют команду CAPA, чтобы получить список доступных функциональных возможностей и — для некоторых возможностей — опций.

Принципиальная возможность применения UTF8 указывает способность сервера обрабатывать почту EAI.

Наличие функциональной возможности «UTF8 user» (то есть UTF8 с опцией «пользователь») означает, что сервер может также обрабатывать имена пользователей и пароли для входа в кодировке UTF-8.

Каждое имя пользователя и пароль в кодировке UTF-8 необходимо подготовить с помощью профиля имени пользователя PRECIS (или его предшественника SASLprep) и отправить, воспользовавшись командой AUTH, которая задействует механизм SASL.

Сразу после входа клиент отправляет команду UTF8, чтобы сообщить серверу, что он (клиент) способен обрабатывать сообщения EAI. После этого сеанс протекает так же, как и для клиентов прежней версии. Если клиент не отправит команду UTF8, но на сервере есть сообщения EAI, сервер либо исключает сообщения EAI из списка доступных сообщений, либо преобразует их в сообщения прежней версии. (Более подробную информацию о возврате к прежней версии см. ниже в разделе «Обратная совместимость».) POP3 обрабатывает только один почтовый ящик на каждое соединение, поэтому не приходится иметь дело с именами папок.

⁸ SASL определен в [RFC 4422](#), а различные механизмы перечислены в созданном IANA реестре механизмов SASL.



Поддержка EAI в IMAP

IMAP имеет комплексную структуру, которая позволяет клиентским MUA управлять несколькими почтовыми ящиками на удаленных серверах. Для управления большими почтовыми ящиками на сервере клиент может использовать сложные команды, такие как SEARCH для поиска сообщений по различным критериям, SORT и THREAD для регулирования порядка возврата сервером сообщений клиенту. На протяжении ряда лет несколько раз предпринимались попытки добавить в протокол IMAP функции интернационализации, которым трудно сосуществовать.

Как и POP3, протокол IMAP использует список возможностей, чтобы указать, какие опциональные функции поддерживаются сервером. При подключении клиента сервер IMAP отправляет в приветственном сообщении список первоначальных возможностей, а после входа клиента — еще один список. (Многие функциональные возможности становятся доступны только после входа клиента.) Затем клиент отправляет команду ENABLE с именем функции, чтобы сообщить серверу, что желает использовать эту функцию.

В протоколе IMAP уже давно есть функции, позволяющие в некоторых ситуациях использовать текст с символами, не входящими в набор ASCII, и различные упорядочивающие последовательности для команд SORT и THREAD, но они неудобны в использовании и часто замедляют обработку, поэтому клиенты редко их применяют.

В 2013 году в IMAP была добавлена почти полная поддержка UTF-8 за счет включения в протокол функций UTF-8=ACCEPT и UTF-8=ONLY. Если сервер предлагает UTF-8=ACCEPT, а клиент включает эту функцию, то клиент и сервер могут отправлять друг другу данные UTF-8 в большинстве полей, где до этого требовалось использовать ASCII. Сюда относятся имена почтовых ящиков и почтовые сообщения, отправляемые сервером клиенту, а также имена почтовых ящиков, строки поиска и почтовые сообщения, отправляемые клиентом серверу.

Поскольку клиент не может включить UTF-8=ACCEPT до входа, в команде IMAP LOGIN: по-прежнему используется только ASCII. Команда IMAP AUTHENTICATE позволяет использовать методы аутентификации SASL, поддерживающие обработку UTF-8. Таким образом, клиент EAI может использовать для входа AUTHENTICATE:, а затем включить функцию UTF-8=ACCEPT.

Если сервер предлагает функцию UTF-8=ONLY, это говорит о том, что он предоставляет *только* совместимую с EAI службу. Клиент должен включить UTF-8=ACCEPT или разорвать связь. Поскольку ASCII является подмножеством UTF-8, клиент может получить доступ к папкам и сообщениям как ASCII, так и EAI.

Если сервер предлагает UTF-8=ACCEPT, но клиент не включает эту функцию, у сервера есть возможность преобразовать имена почтовых ящиков UTF-8 и сообщения EAI в формат прежней версии. См. раздел «Обратная совместимость» ниже.

Заголовки сообщений EAI

Изменения заголовков сообщения в почте EAI варьируются от их полного отсутствия до очень масштабных изменений, в зависимости от заголовка. Рекомендации касательно конкретных заголовков см. в Приложении А.



Заголовки с адресами

К заголовкам с адресами относятся From: (От:), Sender: (Отправитель:), To: (Кому:), Cc: (Копия:), Bcc: (Скрытая копия:), Reply-To: (Адрес для ответа:), Resent-From: (Переслано-От:), Resent-To: (Переслано-Кому:), Resent-Cc: (Переслано-Копия) и Resent-Bcc: (Переслано-Скрытая копия:). Всех они имеют одинаковый синтаксис: разделенный запятыми список адресов и необязательных комментариев. У адресов сообщений EAI может быть любой печатный текст UTF-8 в локальной части, U-метки наряду с A-метками и именами хостов ASCII в доменной части. U-метки предпочтительнее A-меток, поскольку пользователи могут их прочитать.

Текст UTF-8 в комментариях может быть закодирован с помощью MIME, как в почте прежней версии, но это не является требованием. Поскольку текст в кодировке UTF-8 короче, он предпочтительнее.

Поля заголовков, которые видны пользователям

Некоторые поля заголовка, такие как Subject: (Тема:) и Organization: (Организация:) содержат произвольный текст, предназначенный для прочтения пользователями. В этих полях может находиться любой печатный текст UTF-8. Этот текст должен иметь кодировку UTF-8, но (как и в сообщениях прежней версии) также может представлять собой закодированные слова MIME.

Поля заголовков с данными для компьютеров

Некоторые поля заголовка, например Date: (Дата:) и Message-ID: (Идентификатор сообщения:) предназначены для компьютеров, а не для людей. Как правило, использование в этих заголовках символов, не входящих в набор ASCII, не приносит пользы и затрудняет отправку почты системам прежней версии.

Несмотря на то, что содержимое заголовка Date: (Дата:) имеет знакомый англоговорящим пользователям вид, его формат жестко зафиксирован и определяется стандартом [RFC 5322](#), который необходимо строго соблюдать. Агенты MUA часто выполняют синтаксический анализ заголовков и отображают даты в удобном для восприятия локализованном виде после преобразования из фиксированного формата, в котором они находятся в сообщении. Другие заголовки с датами, такие как Resent-Date (Дата пересылки), обрабатываются аналогичным образом.

Заголовок Message-ID: (Идентификатор сообщения:) содержит уникальную строку идентифицирующую сообщение. Программное обеспечение электронной почты может сравнить Message-ID: (Идентификатор сообщения:) в двух сообщениях, чтобы определить, являются ли эти сообщения дубликатами. В ответных сообщениях первоначальный Message-ID: (Идентификатор сообщения:) указывается в заголовках References: (Ссылки:) и In-Reply-To: (В ответ на:), что позволяет агентам MUA группировать связанные сообщения. Синтаксически Message-ID: (Идентификатор сообщения:) похож на адрес электронной почты в скобках, например <text@text>. Большинство почтовых систем создают идентификаторы сообщений, указывая справа от символа @ собственное доменное имя, а слева — псевдослучайный текст. Хотя в идентификаторах сообщений допускается использование символов UTF-8, это нецелесообразно, поскольку в сообщении прежней версии невозможно будет сослаться на сообщение EAI в заголовке



In-Reply-To: (В ответ на:) или References: (Ссылки:). Имена IDN-доменов в идентификаторах сообщений должны быть представлены как A-метки ASCII, а не U-метки.

Примечание: Хотя идентификаторы сообщений напоминают адреса электронной почты, на самом деле это просто строки, не являющиеся адресами. Невозможно вернуть идентификатор сообщения в формат прежней версии; при замене U-метки в идентификаторе сообщения на A-метку будет получен совершенно другой идентификатор сообщения, не связанный с первоначальным.

Заголовки Received

Каждый раз, когда сообщение передается по SMTP с одного компьютера на другой, принимающий компьютер помещает в начало сообщения заголовок Received: (Получено:). Этот заголовок идентифицирует отправляющий и принимающий компьютеры («from» и «by»), протокол передачи («with»), содержит временную метку и часто другую информацию. Вот типичный пример сообщения, отправленного по SMTP с использованием TLS-шифрования:

```
Received: from mail.mit.edu ([128.31.0.31])
  by mail1.iecc.com ([64.57.183.56]) with ESMTPS
  via TCP port 40060/25 id 586470169; 15 Feb 2018 02:28:32 -0000
```

Ключевое слово «with» указывает, какой транспортный протокол использовался для передачи сообщения. В настоящее время наиболее часто используются протоколы, которые перечислены ниже. Когда для сообщения используется транспортный протокол EAI — то есть, с тегом SMTPUTF8 в предваряющей сообщении команде MAIL FROM: — значение «with» меняется, чтобы указать на использование транспортного протокола EAI.

Транспортный протокол прежней версии	Транспортный протокол EAI	Значение
ESMTP	UTF8SMTP	Сеанс SMTP, начинающийся с EHLO
ESMTPS	UTF8SMTPS	ESMTP с TLS-шифрованием
ESMTPA	UTF8SMTPA	ESMTP от аутентифицированного отправителя (обычно для SUBMIT)
ESMTPSA	UTF8SMTPSA	ESMTPA с TLS-шифрованием

Так, к примеру, сообщение, отправленное с использованием EAI SMTP и TLS, имело бы другой заголовок Received: (Получено:).



Received: from mail.mit.edu ([128.31.0.31])
by mail1.iecc.com ([64.57.183.56]) with UTF8SMTPS
via TCP port 40060/25 id 586470169; 15 Feb 2018 02:28:32 -0000

Хотя пользователи обычно не видят заголовков Received, эти заголовки являются бесценным подспорьем при отслеживании транспортировки и анализе проблем с доставкой. Значения «with» упрощают задачу определения, для каких транспортных протоколов использовалась EAI, а для каких нет. Если кажется странным, что ключевым словом расширения SMTP является SMTPUTF8, но в заголовке Received того же документа тегом «with» является UTF8SMTP, да, это именно так.

Заголовки MIME

Большинство сообщений состоит из нескольких частей, таких как альтернативные версии в формате обычного текста и HTML или вложенные файлы. Между частями сообщения находится разделительная строка, которая определена в поле Content-Type: заголовок. Например, эти строки могут находиться в заголовке сообщения:

```
MIME-Version: 1.0  
Content-Type: multipart/alternative;  
  boundary="b1_d9ef97d65ef88703f84b0a36e52aa4a2"
```

А эти — в теле сообщения:

```
--b1_d9ef97d65ef88703f84b0a36e52aa4a2  
Content-Type: text/plain; charset=utf-8
```

... текстовая версия сообщения ...

```
--b1_d9ef97d65ef88703f84b0a36e52aa4a2  
Content-Type: text/html; charset=utf-8
```

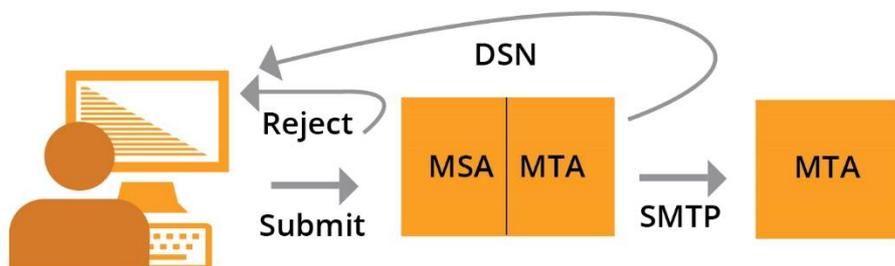
... HTML-версия сообщения ...

```
--b1_d9ef97d65ef88703f84b0a36e52aa4a2--
```

Каждая часть начинается с разделительной строки. За ней следует заголовок Content-Type: (Тип содержимого:), который определяет характер и кодировку данных в этой части и иногда содержит другую информацию, например предлагаемое имя вложенного файла. Дополнительный заголовок Content-Description: (Описание содержимого:) содержит произвольный текст с описанием части MIME, а Content-ID: (Идентификатор содержимого:) — идентификатор, синтаксис которого совпадает с синтаксисом заголовка Message-ID: (Идентификатор сообщения:), позволяющий одной части MIME ссылаться на другую.

EAI практически не меняет заголовки MIME, и в сообщениях EAI следует по-прежнему использовать для разделительных строк символы ASCII. Текст в кодировке UTF-8 может встречаться в дополнительных полях Content-Type: (Тип содержимого:), например в предлагаемом имени вложенного файла, и Content-Description: (Описание содержимого:).

Устранение ошибок доставки и информирование об ошибках доставки



Всякий раз, когда один компьютер отправляет другому сообщение по протоколу SUBMIT или SMTP, могут возникнуть ошибки. При получении команд от отправляющего компьютера принимающий компьютер отправляет ответ, в котором указывает, была ли каждая команда успешно выполнена или нет. В большинстве случаев, если при выполнении любой из команд возникает ошибка, сообщение не удастся отправить.⁹

Если сообщение отклонено в процессе обработки, во время сеанса SMTP или SUBMIT, конкретные коды ошибок позволяют определить причину отклонения. EAI несколько новых кодов отклонения, указывающих на то, что сообщение было отклонено, поскольку получатель не принимает сообщений EAI. Эти коды будут разными, в зависимости от того, какая команда SMTP стала причиной отклонения:

- 550 — не принимается отправитель EAI 5.6.7 (в ответ на MAIL FROM)
- 553 — не принимается получатель EAI 5.6.7 (в ответ на RCPT TO)
- 554 — некоторые получатели не могут принять EAI 5.6.9 (по окончании DATA)

После принятия сообщения принимающий компьютер должен либо доставить это сообщение, либо отправить Уведомление о статусе доставки (DSN) на адрес, который был указан в команде MAIL FROM:. Хотя почтовые системы стараются отклонить почту, которую невозможно доставить, во время сеанса SMTP или SUBMIT, некоторые уведомления DSN неизбежны.

⁹ Если у сообщения несколько получателей, отправитель может послать это сообщение, даже если некоторые адреса получателей признаны недопустимыми, при условии, что принят хотя бы один адрес. Решение о том, выполнять ли отправку в любом случае или прекратить ее и обратиться к пользователю с предложением исправить адреса, определяется качеством реализации.



Предположим, что пользователь с помощью MUA создает сообщение EAI и отправляет его собственной почтовой системе. MTA этой системы использует SMTP для ретрансляции сообщения в систему получателя. Если окажется, что система получателя не принимает почту EAI, сеанс SMTP завершится ошибкой, после чего системе пользователя придется отправить агенту MUA пользователя уведомление DSN.

Формат уведомлений DSN для сообщений прежней версии определен стандартом [RFC 3464](#), который был дополнен стандартом [RFC 6533](#), чтобы охватить сообщения EAI. DSN — это сообщение электронной почты особого формата, имеющее структуру отчета MIME из трех частей: текстовое описание для получателя; пригодный для машинного считывания раздел фиксированного формата с описанием ошибки; копия недоставленного сообщения или, по крайней мере, его заголовки.

DSN может быть получено всего через несколько секунд после отправки первоначального сообщения, но оно также может поступить и через несколько часов, а иногда даже дней, если система получателя недоступна и промежуточная система предпринимает несколько попыток доставки, прежде чем отказаться от этой операции.

Обратная совместимость

В обозримом будущем почта EAI придется сосуществовать с почтой прежней версии. Хотя почтовая система EAI способна принимать любые сообщения от почтовых систем прежней версии, системы прежней версии могут принимать только почтовые сообщения прежней версии.

Получатели с адресами EAI и адресами прежней версии

Когда пользователь отправляет сообщение получателю с адресом EAI или нескольким получателям с адресами EAI, безусловно, все они могут принять почту EAI. Если у получателя адрес ASCII, который занесен в адресную книгу пользователя, то этот адрес может быть помечен как EAI-совместимый или нет. В отсутствие такой информации один из возможных вариантов действий — проявить консервативный подход и отправить сообщение прежней версии, в то время как второй вариант — отправить сообщение EAI в надежде, что оно будет доставлено, и сообщить об ошибке, если доставка не состоится.

Как указано в предыдущем разделе, если сообщение невозможно доставить, отправляющая система может сразу получить код ошибки или сообщение об ошибке может поступить позже в DSN, в зависимости от архитектуры системы.

Если у сообщения несколько получателей, возможно, что некоторые из них не могут принимать сообщения EAI. В таком случае отправляющая система может попытаться создать сообщение прежней версии и отправить его получателям, не принимающим почту EAI. Обратите внимание, что SMTP не делает различий между получателями сообщения, указанными в полях To: (Кому:), Cc: (Копия:) или Bcc: (Скрытая копия:). Процесс возврата к прежней версии одинаков, независимо от того, где в заголовке сообщения указан получатель, не принимающий почту EAI, если таковой имеется. Следовательно, процесс возврата к прежней версии и повторной отправки зависит только от наличия получателей, не принимающих почту EAI.



Целесообразность возврата к прежней версии

Приведенные в нижеследующем разделе рекомендации по возврату к прежней версии носят исключительно опциональный характер. EAI-совместимая система вообще не нуждается в каком-либо возврате к прежней версии.

Хотя стандарты EAI допускают возможность преобразования отправителями сообщений в формат прежней версии, они не поощряют и не ограничивают этого. Доводом в пользу возврата к прежней версии служит то, что это позволяет адресатам с системами прежней версии получить примерную копию сообщений, которые в ином случае они вообще не смогли бы получить. Однако доводом *против* возврата к прежней версии служит то, что при возврате к сообщениям прежней версии часть информации теряется, и получатели подобных сообщений могут счесть их непонятными или раздражающими. Кроме того, нельзя достоверно определить, есть ли у получателя с адресом ASCII возможность принимать почту EAI, поэтому система, выполняющая возврат к прежней версии, скорее всего иногда будет ошибаться и отправлять сообщения прежней версии получателям, которые могли бы принять оригинальное сообщение EAI.

Еще одна проблема в том, что при возврате к прежней версии из заголовков удаляются адреса EAI. В некоторых случаях их можно заменить альтернативными адресами, но гораздо чаще адрес просто удаляется. В отсутствие альтернативного адреса получатели не смогут ответить на сообщение, занести отправителя в адресную книгу или выполнить какие-либо другие операции, которые они обычно выполняют с адресами входящих сообщений. Сообщения прежней версии могут неправильно обрабатываться почтовыми системами, не готовыми к тому, что в полях заголовка From: (От:) не будет адреса. При автоматическом преобразований сообщений, отправитель скорее всего не узнает, что часть адресатов или все они получили сообщение прежней версии. При этом пользователи могут счесть, что получение разными адресатами разных версий сообщения, создает путаницу.

Где возможен возврат к прежней версии почты

Попытка MTA отправить сообщение получателю может завершиться ошибкой из-за того, что MTA получателя не поддерживает SMTPUTF8 или отклоняет сообщение с кодом состояния, указывающим, что получатель не принимает почту EAI. В этот момент MTA может попытаться создать и отправить получателю сообщение в формате прежней версии. Теоретически, это может сделать любой MTA в канале передачи сообщения, однако полезный результат с большей вероятностью будет получен, если MTA удастся найти альтернативный адрес для замены адресов EAI в сообщении, преобразованном в формат прежней версии. Вероятность успеха при возврате к прежней версии выше всего в системе веб-почты, где MUA, MSA и MTA для исходящих сообщений интегрированы в единый пакет.

Еще одним целесообразным местом преобразования сообщений в формат прежней версии является программное обеспечение диспетчера листов рассылки. Если некоторые подписчики на лист рассылки способны принимать почту EAI, а некоторые нет, программное обеспечение диспетчера листов рассылки могло бы создавать сообщения прежней версии для соответствующих получателей.



Общие способы возврата к прежней версии

В стандартах EAI описаны два способа преобразования полученных сообщений EAI в формат прежней версии на серверах POP или IMAP. Эти два подхода также, с незначительными изменениями, пригодны для преобразования отправителями сообщений, адресованных получателям, не поддерживающим почту EAI.

В стандарте [RFC 6857](#), *Преобразование интернационализованных сообщений электронной почты в формат прежней версии после доставки*, описана комплексная схема, позволяющая создавать такие заголовки, как Downgraded-To: (Преобразовано в:) и Downgraded-From: (Преобразовано из:), предназначенные для включения всех данных, которые были в сообщении, для получения возможности впоследствии отменить внесенные изменения. В стандарте [RFC 6858](#), *Упрощенная схема преобразования интернационализованных сообщений электронной почты в формат прежней версии для протоколов POP и IMAP*, описана гораздо более простая схема, в рамках которой информация EAI сохраняется, когда это удобно, но нет стремления сохранить все детали. Мы рекомендуем использовать более простой подход RFC 6858.

Заголовки и тела сообщений

В большинстве случаев, чтобы сообщение стало приемлемым для почтовых систем, не поддерживающих EAI, необходимо преобразовать в формат прежней версии только его заголовки. Теоретически могли бы существовать системы, принимающие исключительно почту в 7-битной кодировке ASCII, но на практике все почтовые системы сейчас поддерживают 8BITMIME и, следовательно, могут принимать UTF-8 в теле сообщения.

Возврат к прежней версии заголовков с адресами

Заголовки с адресами, такие как From: (От:), To: (Кому:) и Resent-Сс: (Переслано-Копия:) содержат адреса и необязательные комментарии. Поскольку комментарии могут быть закодированными словами, использование которых допускается в сообщениях прежней версии, при возврате к прежней версии только адреса нуждаются в специальной обработке. Преобразование адресов UTF-8 в формат прежней версии выполняется в два этапа: определите, существует ли альтернативный адрес ASCII, и затем надлежащим образом измените заголовок.

Если адрес находится под управлением локальной системы (как правило, это адреса в заголовке From: (От:)), может существовать список локально эквивалентных адресов EAI и ASCII. В таком случае найдите эквивалентный адрес ASCII в этом списке. Кроме того, если у пользователя есть адресная книга, доступная в момент возврата к прежней версии, альтернативный адрес для замены адреса EAI, возможно, есть в этой книге.

Помимо этого, есть один редкий случай, когда можно преобразовать адрес в альтернативный адрес ASCII механически. Если адрес имеет вид <ascii@U-метка> — то есть, локальная часть ASCII и доменная часть UTF-8 — допускается замена U-меток доменов на A-метки, в результате которой будет получен адрес ASCII. Это *единственная* ситуация, в которой возможен механический возврат к прежней версии. В частности, невозможно выполнить механическое преобразование локальной части UTF-8 в формат прежней версии.



Если существует альтернативный адрес, замените первоначальный адрес на альтернативный адрес ASCII.

Если альтернативного адреса нет, при возврате к прежней версии первоначальный адрес удаляется из заголовка и заменяется пустой группой адреса, состоящей из двоеточия и точки с запятой : ; .¹⁰ Можно скопировать первоначальный адрес EAI в комментарий к адресу, чтобы получатель его увидел, однако получатели с системами прежней версии не смогут ответить на это сообщение.

Рассмотрите следующий пример, где прописные буквы — это текст UTF-8, а подчеркнутый текст означает закодированное слово MIME.

До: From: "CCCCC" <MMMM@DDDD.DDD>
После: From: "CCCCC MMMM@DDDD.DDD" ; ;

Комментарий (CCCCC) и адрес EAI (MMMM@DDDD.DDD) перемещены и преобразованы в комментарий в виде закодированного слова, а адрес заменен пустой группой для сохранения синтаксической правильности заголовка. Подробная информация о закодированном слове комментария не представляет важности, потому что это комментарий. Обратите внимание, что в преобразованной форме больше нет адреса, на который получатель мог бы отправить ответ.

В этом примере рассматривается адрес в поле From: (От:), однако такой же алгоритм преобразования можно использовать для всех заголовков с адресами. Если в заголовке несколько адресов, обрабатывайте каждый по-отдельности, преобразуя адреса EAI и не затрагивая адреса прежней версии.

Возврат к прежней версии других заголовков, которые видны пользователям

В заголовках, которые видны пользователям, можно использовать закодированные слова MIME, представляющие текст UTF-8. Замените весь текст UTF-8 в заголовке сообщения на закодированные слова. Сюда относится текст в заголовках Subject: (Тема:), Comment: (Комментарий:) и Keyword: (Ключевое слово:), а также в многочисленных полустандартных заголовках, таких как Organization: (Организация:).

¹⁰ Изначально стандарты электронной почты допускали использование пустых групп только в конкретных заголовках, однако [RFC 6854](#) смягчил правила, разрешив использовать их во всех полях, где может находиться адрес.



Возврат к прежней версии заголовков, которые не видны пользователям

Как упоминалось выше, в заголовках Message-ID: (Идентификатор сообщения:) не следует использовать символы, не входящие в набор ASCII. Если идентификатор сообщения тем не менее содержит символы, не входящие в набор ASCII, удалите его и создайте новый. (Поскольку этот идентификатор — непрозрачная строка, нет смысла стараться создать идентификатор прежней версии, который выглядит как первоначальный.) Идентификаторы сообщений также встречаются в заголовках References: (Ссылки:) и In-Reply-To: (В ответ на:). Если такие заголовки содержат идентификаторы с символами, не входящими в набор ASCII, удалите эти идентификаторы, а если в заголовке больше не осталось идентификаторов, удалите сам заголовок.

Если символы UTF-8 есть в каких-либо других заголовках, которые не видны пользователям, удалите эти заголовки, поскольку их невозможно преобразовать в формат прежней версии и они не влияют на то, что видит пользователь.

Если заголовок MIME Content-Type: (Тип содержимого:) — в заголовках сообщения либо в заголовке MIME части тела сообщения — содержит поле с текстом, символы которого не входят в набор ASCII, удалите такое поле. Все обязательные поля в этих заголовках содержат только символы в кодировке ASCII. Если в заголовке больше не осталось полей, удалите заголовок.

Возврат к прежней версии в POP и IMAP

Если в почтовом ящике POP или IMAP есть сообщения EAI, но клиент MUA не поддерживает EAI, у сервера POP или IMAP есть возможность преобразовать сообщения EAI в формат прежней версии, чтобы пользователь смог увидеть хотя бы приблизительный вариант отправленного сообщения. Процесс возврата к прежней версии аналогичен описанному выше, за исключением того, что у серверов POP и IMAP вряд ли есть альтернативные адреса. Этот вопрос подробнее рассматривается в [RFC 6858](#).

Примечания касательно возврата к прежней версии

Если у сообщения, преобразованного в формат прежней версии, есть подпись DKIM, процесс преобразования сделает такую подпись действительной, поэтому удалите ее. Иногда есть возможность подписать сообщение снова. См. следующий раздел.

Вместо того, чтобы пытаться отправить каждое сообщение EAI на адрес, не являющийся адресом EAI, для проверки возможности доставки, в случае недоставки сообщения, если адрес получателя есть в адресной книге, присвойте записи тег, указывающий, что это адрес прежней версии. (Кроме того, обеспечьте возможность удаления этого тега после модернизации системы получателя.)

Сообщения об отсутствии на работе, уведомления о прочтении и другие автоответчики

Почтовые системы часто позволяют отправлять автоматические ответы в случае отсутствия на работе и аналогичные уведомления.

С точки зрения EAI почта автоответчика ничем не отличается от любой другой. Если у получателя ответа (то есть у отправителя первоначального сообщения) есть адрес EAI или из других источников известно, что он принимает почту EAI, автоответчик может отправить сообщение EAI. В ином случае нужно отправить сообщение прежней версии, вновь созданное или преобразованное из сообщения EAI.



Проверка подлинности почты и фильтры спама

Поскольку проблема спама все более усугубляется, в почтовые системы был добавлен широкий спектр функций для борьбы с ним. К наиболее эффективным из них относится аутентификация, которая подтверждает личность предполагаемого отправителя сообщения.

Для получения дополнительных сведений об аутентификации почты в контексте EAI, помимо представленных ниже, см. интернет-проект <https://datatracker.ietf.org/doc/draft-levine-appsarea-eaiauth/>.

Аутентификация SPF

SPF (Sender Policy Framework — инфраструктура политики отправителей) проверяет наличие IP-адреса, с которого отправлено сообщение, в списке официальных IP-адресов домена. Проверяется домен, который указан в обратном адресе MAIL FROM:, или, если это поле адреса пустое, в параметре EHLO/HELO.

Домен публикует свои официальные адреса в виде записи TXT с достаточно сложным синтаксисом. Типовая запись SPF для icann.org:

```
icann.org IN TXT "v=spf1 ip4:192.0.32.0/20 ip4:199.91.192.0/21
ip4:64.78.40.0/27 ip4:162.216.194.0/27 ip6:2620:0:2d0::0/48
ip6:2620:0:2830::0/48 ip6:2620:0:2ed0::0/48
include:salesforce.icann.org -all"
```

В этом списке несколько адресных блоков IPv4 и IPv6 и все блоки, опубликованные на salesforce.icann.org.

Процесс использования SPF в почте EAI очень простой. Поскольку эти данные опубликованы в DNS, модуль проверки подлинности выполняет типовую операцию поиска домена IDNA, преобразуя U-метки в A-метки, а затем отправляет запрос DNS. Сами записи SPF не содержат UTF-8, поэтому процедура проверки того, соответствует ли IP-адрес записи SPF, не меняется.

Аутентификация DKIM

Агенты МТА, выполняющие отправку и ретрансляцию, используют DKIM для добавления к почтовым сообщениям криптографических подписей. Принимающая система может проверить эту подпись. Если подпись действительна, система убеждается, что проверенное сообщение идентично подписанному отправителем. Подпись помещается в заголовок DKIM-Signature, аналогичный заголовку этого сообщения от Microsoft:

```
DKIM-Signature: v=1; a=rsa-sha256; c=relaxed/relaxed; d=microsoft.com;
s=rm1; h=From:Date:Subject:Message-ID:Content-Type:MIME-Version;
i=billg@microsoft.com; bh=SwUgUaGo9gVX7cc6F206JZvUfMea/5YYkAej9uuVF6Q=;
b=EdBiodGIf4bCmbGEZBQFUa+MCfoPhm138ox5x6iq4FCmITu522v1Q/9t/A+E=
```

Подпись — это список пунктов тег=значение. Тегом является подписавший домен, а s= тег — это «селектор», используемый для поиска проверочного ключа в DNS.



Необязательный пункт `i=` тег — лицо, важное для подписавшего.¹¹ Пункт `bh=` тег — хэш тела сообщения, а `b=` тег — подписанный хэш отдельных заголовков сообщения вместе с самим заголовком DKIM-Signature. Принимающая система проверяет значения хэшей на соответствие полученному сообщению и проверочному ключу в DNS, чтобы убедиться в том, что подпись действительна.

Проверочный ключ хранится в DNS по адресу `<селектор>._domainkey.<домен>`, то есть в данном случае, поскольку `s=` селектор — `rm1`, а `d=` домен — `microsoft.com`, это был бы адрес (опуская некоторые интересные подробности):

```
rm1._domainkey.microsoft.com. TXT "v=DKIM1; k=rsa;
p=MIGfMA0GCSqGSIb3DQEBAQUAA4GNADCBiQKBgQCkHq3ztGIm1R8a1D+7oZiaG5mT;"
```

EAI вносит незначительные изменения в подписи DKIM. Значениями `s=` селектор и доменами в тегах `d=` и `i=` могут быть U-метки; строка `i=` тоже может иметь кодировку UTF-8.¹² Это не должно влиять на создание подписей DKIM, поскольку символы в доменах обрабатываются так же, как и другие символы в сообщениях. При проверке подписи DKIM все U-метки в теге `d=` домен или `s=` селектор должны быть преобразованы в A-метки для поиска в DNS.

Если система преобразует сообщение в формат прежней версии, в результате такого преобразования сообщение меняется настолько сильно, что существующие подписи DKIM становятся недействительными. Если возврат к прежней версии осуществляется в MUA, то сообщение, как правило, будет подписано снова при прохождении через MSA. Если возврат к прежней версии осуществляется в MSA или MTA, сообщение следует подписать снова, если у MSA или MTA есть ключ для подписания.

Аутентификация DMARC

DMARC — антифишинговая схема, которая использует результаты аутентификации SPF и DKIM для проверки подлинности домена в заголовке `From: (От:)`. Если домен публикует в DNS политику DMARC, получатели проверяют, есть ли у сообщения достоверный результат SPF или достоверная подпись DKIM для этого домена. Если да, сообщение проходит. Это не значит, что оно не является спамом, но это значит, что оно действительно получено от предполагаемого отправителя.

Предположим, к примеру, что в заголовке `From: (От:)` указано следующее:

```
From: Bob Smith <bob@example.com>
```

¹¹ Хотя пункт `i=` тег имеет такой же синтаксис, что и адрес электронной почты, он не обязательно должен быть адресом лица, отправившего сообщение, или какого-либо другого.

¹² В документе RFC 6376 сказано, что все домены в подписях должны быть A-метками, однако это требование, вероятно, изменится.



Модуль проверки подлинности DMARC оценивает, подтверждают ли результаты проверки SPF, что сообщение отправлено с example.com (то есть, что адрес MAIL FROM: находился в домене example.com), или, что у сообщения есть действительная подпись DKIM в d=example.com. Если хотя бы в одном случае результат положительный, проверка DMARC считается успешно пройденной.

Как только в системе EAI будут реализованы SPF и DKIM, для внедрения DMARC останется сделать совсем немного. При проверке домена в заголовке From: (От:), если это IDN-домен, то SPF домена или тегу DKIM d= домен может соответствовать А-метка или U-метка IDN-домена.

Фильтрация спама

EAI мало влияет на методы фильтрации спама. Поскольку все в сообщении, кроме адресов в заголовках To:/From:/Cc: (Кому:/От:/Копия:),¹³ уже может иметь кодировку UTF-8, фильтры уже занимаются обработкой интернационализованного текста. Если в системе фильтрации предусмотрены белый и черный списки адресов отправителей, такие списки должны обрабатывать адреса EAI, а при проверке входящей почты должны обрабатывать как U-метки, так и А-метки доменов.

Одним из возможных источников проблем являются пустые группы адреса в сообщениях, преобразованных в формат прежней версии. Такие группы разрешено использовать в заголовках To: (Кому:) и Cc: (Копия:) с 1970-х годов, однако использовать их в заголовках From: (От:) было разрешено только после вступления в силу стандарта RFC 6854 в 2013 году. Еще существуют почтовые системы, которые считают группы в поле From: (От:) недопустимыми, в некоторых случаях они отображают их неверно, а иногда все сообщение признается спамом.

¹³ У входящих сообщений обычно отсутствуют заголовки Bcc (Скрытая копия), так как подразумевается, что они должны изыматься при отправке сообщения.



Приложение А. Обработка полей заголовка сообщения

Фраза «Заменить или удалить адрес EAI» в этой таблице означает, что нужно заменить адрес на адрес прежней версии, если это возможно, или удалить адрес, как описано в разделе «Возврат к прежней версии заголовков с адресами» выше. Фраза «Разрешен текст UTF-8» означает использование буквенного текста вместо закодированных слов MIME.

Имя поля заголовка	Изменения при EAI	Действия для возврата к прежней версии
Authentication-Results	Разрешены IDN-домены в предназначенных для доменов полях	Удалить, если содержит UTF-8
Всс	Разрешены адреса EAI	Заменить или удалить адреса EAI, удалить заголовок, если не осталось адресов
Сс	Разрешены адреса EAI	Заменить или удалить адреса EAI
Comments	Разрешен текст UTF-8	Использовать вместо текста UTF-8 закодированные слова MIME
Content-Description	Разрешен текст UTF-8	Использовать вместо текста UTF-8 закодированные слова MIME
Content-ID	Отсутствуют	Удалить, если содержит UTF-8
Content-Transfer-Encoding	Отсутствуют	Отсутствуют



Content-Type	Обычно отсутствуют, возможен UTF-8 в необязательных тегах имен файлов	Удалить все теги со значениями UTF-8
Date	Отсутствуют	Отсутствуют
DKIM-Signature	Разрешены IDN-домены в тегах d= i= s=, разрешен UTF-8 в i= локальной части	Удалить и подписать снова
Errors-To	Могут быть разрешены адреса EAI	Заменить адреса EAI или удалить заголовок
From	Разрешены адреса EAI	Заменить или удалить адреса EAI
In-Reply-To	Отсутствуют	Удалить идентификаторы Message-ID в формате UTF-8, удалить заголовок, если не осталось идентификаторов Message-ID.
Keywords	Разрешен текст UTF-8	Использовать вместо текста UTF-8 закодированные слова MIME
List-Archive	Mailto: URI может содержать адреса EAI	Заменить или удалить адреса EAI, удалить поле заголовка, если не осталось URI
List-Help	Mailto: URI может содержать адреса EAI	Заменить или удалить адреса EAI, удалить поле заголовка, если не осталось URI



List-ID	Отсутствуют (Текст UTF-8 не рекомендуется.)	Удалить, если содержит текст UTF-8
List-Owner	Mailto: URI может содержать адреса EAI	Заменить или удалить адреса EAI, удалить заголовок, если не осталось URI
List-Post	Mailto: URI может содержать адреса EAI	Заменить или удалить адреса EAI, удалить заголовок, если не осталось URI
List-Subscribe	Mailto: URI может содержать адреса EAI	Заменить или удалить адреса EAI, удалить заголовок, если не осталось URI
List-Unsubscribe	Mailto: URI может содержать адреса EAI	Заменить или удалить адреса EAI, удалить заголовок, если не осталось URI
Message-ID	Отсутствуют	Должны отсутствовать, удалить и заменить все заголовки Message-ID в формате UTF-8
MIME-Version	Отсутствуют	Отсутствуют
Organization	Разрешен текст UTF-8	Использовать вместо текста UTF-8 закодированные слова MIME
Received	Разрешены IDN-домены и почтовые ящики EAI	Удалить, если содержит текст UTF-8



References	Отсутствуют	Удалить идентификаторы Message-ID в формате UTF-8, удалить заголовок, если не осталось идентификаторов Message-ID.
Reply-To	Разрешены адреса EAI	Заменить или удалить адреса EAI, удалить поле заголовка, если не осталось адресов
Resent-Bcc	Разрешены адреса EAI	Заменить или удалить адреса EAI, удалить поле заголовка, если не осталось адресов
Resent-Cc	Разрешены адреса EAI	Заменить или удалить адреса EAI
Resent-Date	Отсутствуют	Отсутствуют
Resent-From	Разрешены адреса EAI	Заменить или удалить адреса EAI
Resent-Message-ID	Отсутствуют	Должны отсутствовать, удалить и заменить все ID в формате UTF-8
Resent-Sender	Разрешены адреса EAI	Заменить адрес EAI или удалить поле заголовка
Resent-To	Разрешены адреса EAI	Заменить или удалить адреса EAI



Return-Path	Разрешены адреса EAI	Удалить, если содержит UTF-8
Отправитель	Разрешены адреса EAI	Заменить адрес EAI или удалить поле заголовка
Тема	Разрешен текст UTF-8	Использовать вместо текста UTF-8 закодированные слова MIME
Кому	Разрешены адреса EAI	Заменить или удалить адреса EAI

Приложение Б. Перечень шагов, необходимых для EAI

Ниже приведено краткое описание изменений в программном обеспечении, позволяющих добавить возможности EAI в различные виды почтовых программ.

Изменения агентов MTA

- Разделить сообщения EAI и сообщения прежней версии либо при помощи внешнего тега, либо путем сканирования на предмет наличия UTF-8 в заголовках
- На стороне сервера: объявить ключевое слово EHLO SMTPUTF8, принимать ключевое слово SMTPUTF8 в MAIL FROM:, указывать UTF8SMTP в заголовке Received: (Получено:)
- На стороне клиента: искать ключевое слово EHLO SMTPUTF8, отправлять ключевое слово SMTPUTF8 в MAIL FROM:
- Если отправка сообщения EAI приводит к ошибке, преобразовать сообщение в формат прежней версии и отправить вновь, но только в том случае, если у MTA есть необходимая информация из адресной книги и ключи для подписания DKIM

Изменения агентов MSA

- Объявить ключевое слово EHLO SMTPUTF8
- Разделить сообщения EAI и сообщения прежней версии при помощи ключевого слова MAIL FROM:



- Если отправка сообщения EAI приводит к ошибке, преобразовать сообщение в формат прежней версии и отправить вновь, но только в том случае, если у MSA есть необходимая информация из адресной книги и ключи для подписания DKIM

Изменения агентов MUA

- Разрешить UTF-8 в адресах сообщений
- Разрешить UTF-8 в адресах адресных книг, разрешить в записях использование как адресов EAI, так и адресов прежней версии
- Искать ключевое слово ENLO SMTPUTF8 в данных MSA, добавлять ключевое слово SMTPUTF8 в MAIL FROM:, не отправлять сообщения EAI тем MSA, которые не могут их обработать
- Если отправка сообщения EAI приводит к ошибке, преобразовать сообщение в формат прежней версии и отправить вновь, но только в том случае, если у MUA есть необходимая информация из адресной книги

Серверы IMAP

- Поддерживать имена папок UTF-8
- Помечать тегами сообщения EAI и сообщения прежней версии либо при создании сообщения, либо после сканирования заголовков
- Добавить функцию IMAP UTF-8=ACCEPT
- (Необязательно) По мере необходимости выполнять преобразование сообщений EAI в формат прежней версии для клиентов прежней версии

Серверы POP

- Помечать тегами сообщения EAI и сообщения прежней версии либо при создании сообщения, либо после сканирования заголовков
- Добавить поддержку UTF8 и команду UTF8:
- (Необязательно) По мере необходимости выполнять преобразование сообщений EAI в формат прежней версии для клиентов прежней версии

Приложение В. Как работает SMTP?

В протоколе SMTP используется серия команд и ответов для пересылки сообщения с одного почтового сервера на другой. Все команды и ответы состоят из символов ASCII (кроме адресов EAI). Ответы начинаются с трехразрядного числа, определяющего содержание ответа, за которым следует текст, обычно являющийся комментарием.

В приведенном ниже примере пользователь sam@sender.org отправляет письмо на адрес ray@receive.net. Для сохранения простоты отправляющий хост — sender.org, а хост получателя — receive.net.



Прежде всего, отправляющий компьютер инициирует соединение, а принимающий компьютер отправляет первоначальный ответ. Отправляющий компьютер посылает команду EHLO (расширенное приглашение), чтобы идентифицировать себя и запросить список функций, поддерживаемых принимающим компьютером:

```
S: <connect>
R: 220 receive.net ESMTP
S: EHLO sender.org
R: 250-8BITMIME
R: 250 PIPELINING
```

Кодовые числа вида 2xx указывают на успешное выполнение команды. В этом примере хост получателя поддерживает две функции: 8BITMIME и PIPELINING. Их поддерживают почти все хосты. (8BITMIME позволяет использовать текст с символами, не входящими в набор ASCII, в телах сообщений, PIPELINING позволяет отправителю посылать несколько команд, не дожидаясь подтверждения получения каждой из них.)

Отправляющий компьютер посылает команду MAIL FROM: , указывая адрес отправителя, на который можно отправлять сообщения о состоянии и отчеты о доставке, и опционально другую информацию из конверта.

```
S: MAIL FROM: <sam@sender.org>
R: 250 Sender accepted
```

Отправляющий компьютер посылает команды RCPT TO: , в каждой из которых указан адрес получателя сообщения.

```
S: RCPT TO: <ray@receive.net>
R: 250 Recipient accepted
```

(При наличии нескольких получателей, для каждого из них была бы отправлена отдельная команда RCPT TO: и получен ответ.)

Отправляющий компьютер посылает команду DATA: и после получения ответа от принимающего компьютера отправляет сообщение как единый блок текста, за заголовком следует все сообщение целиком, включая заголовок и тело, после этого идет строка с единственной точкой, указывающей на конец сообщения.

```
S: DATA
R: 354 Send your message
S: From: sam@sender.org
S: To: ray@receive.net
S: Subject: lunch
S:
S: How about lunch at 12:30?
S: .
R: 250 Message accepted 409343fg34
```

На этом передача сообщения завершена, поэтому отправитель посылает команду QUIT для завершения сеанса.

```
S: QUIT
R: 221 Sayonara
```



Добавление функций UTF-8 в пример использования SMTP

EAI вносит незначительные изменения в протокол. Ответ сервера на команду EHLO содержит ключевое слово UTF8SMTP, которое указывает на поддержку сервером EAI:

```
S: <connect>
R: 220 receive.net ESMTP
S: EHLO sender.org
R: 250-8BITMIME
R: 250-UTF8SMTP
R: 250 PIPELINING
```

Команда клиента MAIL FROM: содержит ключевое слово SMTPUTF8, которое указывает, что это сообщение EAI.

```
S: MAIL FROM: <猫王@普遍接受-测试.世界> SMTPUTF8
R: 250 Sender accepted
```

Остальная часть сеанса SMTP не меняется, за исключением того, что в заголовках сообщения разрешено использовать данные UTF-8:

```
S:RCPT TO:<ray@receive.net>
R:250 Recipient accepted
S:DATA
R:354 Send your message
S:From: 猫王 <猫王@普遍接受-测试.世界>
S:To: ray@receive.net
S:Subject: 我们要吃午饭吗?
S:
S:How about lunch at 12:30?
S:.
R:250 Message accepted 389dck343fg34
S:QUIT
R:221 Sayonara
```



Приложение Г. RFC, относящиеся к рассматриваемой теме

Большинство спецификаций интернет-служб разрабатывается и поддерживается Инженерной проектной группой интернета (IETF), которая аффилирована с Обществом интернета. В силу исторических причин основная серия документов, публикуемых IETF, называется «Запросы комментариев» или RFC. Все RFC пронумерованы, начиная с RFC 1 в 1969 году и примерно до RFC 8300 на настоящий момент. Некоторые RFC являются стандартами, в то время как в других описаны современные передовые методы или представлена другая информация, которая важна для технических аспектов функционирования интернета. Иногда IETF издает новые стандарты RFC для обновления или замены предыдущих стандартов. При этом группа, как правило, старается избежать изменений, несовместимых с действующими стандартами.

Все RFC находятся в открытом и бесплатном доступе на сайте издательства RFC по адресу <https://www.rfc-editor.org/> и во многих других местах в интернете.

Важные RFC, относящиеся к почте

Ниже перечислены некоторые более важные RFC. Более длинный перечень см. в документе [UASG 006](#),

Для SMTP:

- [RFC 5321](#) — Протокол SMTP, транспортировка почты от одного MTA к другому (ранее RFC 821 и 2821)
- [RFC 6609](#) — Отправка почты, от MTA к MSA (ранее RFC 2476 и 4409)

Для форматов почтовых сообщений:

- [RFC 5322](#) — Формат интернет-сообщений (ранее RFC 822 и 2822)
- [RFC 2045](#) — Многоцелевые расширения интернет-почты (MIME), первая часть: Формат тела интернет-сообщений
- [RFC 2046](#) — Многоцелевые расширения интернет-почты (MIME), вторая часть: Типы носителей
- [RFC 2047](#) — Многоцелевые расширения интернет-почты (MIME), третья часть: Расширения для заголовков сообщений с текстом, символы которого не входят в набор ASCII
- [RFC 2048](#) — Многоцелевые расширения интернет-почты (MIME), четвертая часть: Процедуры регистрации
- [RFC 2049](#) — Многоцелевые расширения интернет-почты (MIME), пятая часть: Критерии соответствия требованиям и примеры
- [RFC 3464](#) — Расширяемый формат сообщений для уведомлений о доставке
- [RFC 6854](#) — Обновление формата интернет-сообщений, позволяющее использовать синтаксис групп в полях заголовка «From:» («От:») и «Sender:» («Отправитель:»)



Для POP и IMAP:

- [RFC 1939](#) — Протокол POP, версия 3 (POP3)
- [RFC 3501](#) — Протокол доступа к сообщениям в интернете, версия 4, ред. 1 (IMAP4)

Для IDN:

- [RFC 3629](#): UTF-8, формат преобразования ISO 10646
- [RFC 5890](#), [5891](#), [5892](#), [5893](#), [5894](#) и [5895](#) — Интернационализованные доменные имена в приложениях (IDNA)

Для почты EAI:

- [RFC 6530](#) — Обзор и концепция интернационализованной электронной почты
- [RFC 6531](#) — Расширение SMTP для интернационализованной электронной почты
- [RFC 6532](#) — Интернационализованные заголовки сообщений электронной почты
- [RFC 6533](#) — Интернационализованные уведомления о доставке и состоянии сообщения
- [RFC 6855](#) — Поддержка UTF-8 в IMAP
- [RFC 6856](#) — Поддержка UTF-8 в протоколе POP версии 3 (POP3)
- [RFC 6857](#) — Преобразование интернационализованных сообщений электронной почты в формат прежней версии после доставки
- [RFC 6858](#) — Упрощенная схема преобразования интернационализованных сообщений электронной почты в формат прежней версии для протоколов POP и IMAP

Для PRECIS и сопоставления строк:

- [RFC 8264](#) — Платформа PRECIS: Подготовка, реализация и сравнение интернационализованных строк в протоколах приложений
- [RFC 8265](#) — Подготовка, реализация и сравнение интернационализованных строк, являющихся именами пользователей и паролями
- [RFC 8266](#) — Подготовка, реализация и сравнение интернационализованных строк, являющихся псевдонимами
- [RFC 7790](#) — Преобразование символов для классов подготовки, реализации и сравнения интернационализованных строк (PRECIS)
- [RFC 6885](#) — Пересмотр Stringprep и описание проблемы подготовки и сравнения интернационализованных строк (PRECIS)
- [RFC 6943](#) — Проблемы сравнения идентификаторов в целях безопасности
- [UTS#39](#) — Механизмы безопасности Unicode, разделы 3.3 «Профили безопасности электронной почты для идентификаторов» и 5.2 «Определение уровня ограничений»



Для SASL (вход и аутентификация для POP, IMAP и отправки):

- [RFC 4422](#) — Простой уровень аутентификации и безопасности (SASL)
- [RFC 4616](#) — Простой уровень аутентификации и безопасности (SASL): механизм PLAIN
- Механизм SASL LOGIN [Интернет-проект](#)
- IANA [Реестр методов SASL](#)

Для URL и URI:

- [RFC 3986](#) — Унифицированный идентификатор ресурса (URI): типовой синтаксис
- [RFC 3897](#) — Интернационализованные идентификаторы ресурсов (IRI)

Для аутентификации сообщений:

- [RFC 6376](#) — Подписи DomainKeys Identified Mail (DKIM)
- [RFC 7208](#) — Инфраструктура политики отправителей (SPF) для авторизации использования доменов в электронной почте, версия 1
- [RFC 7489](#) — Идентификация сообщений, создание отчетов и определение соответствия по доменному имени (DMARC)
- [Проект Eaiauth](#) — Аутентификация интернационализованной электронной почты

Для ввода и отображения RTL и разнонаправленного текста:

- [RFC 5893](#) — Алфавиты с направлением письма справа налево в интернационализованных доменных именах в приложениях (IDNA)



Приложение Д. Другие источники рекомендаций

- Документация UASG: <https://uasg.tech/documents/>
- Стандарты, регламентирующие формат и спецификации электронной почты: см. RFC, относящиеся к электронной почте, список которых приведен выше
- Стандарты, регламентирующие HTML и другие данные в сети: см. сайт консорциума W3C <https://www.w3.org/standards/>
- Общие вопросы программирования электронной почты: см. сайт Stack Overflow <https://www.stackoverflow.com>
- Листы рассылки для обсуждения электронной почты. Для большинства листов рассылки поддерживаются интернет-архивы; ознакомьтесь с этими архивами. Возможно, там уже есть ответ на вопрос, который вы хотите задать.
 - Postfix <http://www.postfix.org/lists.html>
 - Qmail <http://www.gmail.org/top.html#patches> и <https://lists.gt.net/qmail/users/>
 - Sendmail <https://groups.google.com/forum/#!forum/comp.mail.sendmail>
 - Cyrus IMAP <https://cyrusimap.org/imap/support/feedback-mailing-lists.html>
 - Dovecot <https://dovecot.org/maillinglists.html>
- Дискуссионные листы рассылки рабочих групп IETF. Их полезно использовать для вопросов о стандартах RFC и тонкостях их реализации, но не для вопросов общего характера об электронной почте или программировании.
 - Лист рассылки DKIM: ietf-dkim@ietf.org
 - Лист рассылки для обсуждения вопросов функционирования служб DKIM: dkim-ops@mipassoc.org
 - Лист рассылки DMARC: dmarc-discuss@dmarc.org
 - Лист рассылки SMTP: ietf-smtp@ietf.org



Глоссарий

ASCII-совместимое кодирование (ACE): Формат, используемый в А-метке, имеющий вид хп-*-наполнение*, где *наполнение* — Punycode-версия UTF-8.

А-метка: Вариант метки домена UTF-8 в кодировке ASCII, имеющий вид хп-*-наполнение*, где *наполнение* — Punycode-версия UTF-8. Сравните с U-меткой.

IDNA: Интернационализированные доменные имена в приложениях. Правила, которые определяют допустимые символы и их сочетания в доменных именах UTF-8. Текущая версия — IDNA2008, определена в стандартах RFC 5891–5895.

IMAP: Internet Message Access Protocol — протокол доступа к сообщениям в интернете, стандарт управления почтовыми сообщениями и папками на удаленных серверах. Текущая версия — IMAP4. Сравните с POP.

MIME: Multipurpose Internet Mail Extensions — многоцелевые расширения интернет-почты, спецификация кодирования различных видов данных, в том числе текста с символами, не входящими в набор ASCII, и включения нескольких логических частей в состав одного сообщения.

MX: Адрес почтового шлюза, запись DNS домена, которая идентифицирует хост для получения почты.

POP: Post Office Protocol — почтовый протокол, стандарт загрузки сообщений с удаленного сервера. Текущая версия — POP3. Сравните с IMAP.

PRECIS: Preparation and Comparison of Internationalized Strings — подготовка и сравнение интернационализированных строк, комплект документов IETF, определяющих принципы создания пользовательских имен и псевдонимов в формате UTF-8.

Procmail: MDA, получивший широкое распространение в системах Unix и Linux. [Статья в Википедии.](#)

Punycode: Кодирование UTF-8 в виде последовательности символов ASCII, применяемое в А-метках ACE ([RFC 3492](#)).

SASL: Simple Authentication and Security Layer — простой уровень аутентификации и безопасности, группа стандартов IETF, определяющих схемы аутентификации по имени пользователя и паролю, применяемые для входа на сервера SUBMIT, IMAP и POP.

SIEVE: Язык для фильтрации и сортировки входящих почтовых сообщений ([RFC 5228](#)).

SMTP: Simple Mail Transport Protocol — простой протокол передачи почты, способ обмена почтой между компьютерами в интернете.

SUBMIT или отправка: Передача недавно созданного сообщения для ретрансляции и/или доставки ([RFC 6409](#))

U-метка: Метка домена в формате UTF-8. Сравните с А-меткой.

Агент доставки почты (MDA): Серверная программа, которая обрабатывает входящую почту и обычно сохраняет ее в почтовом ящике или папке.

Агент отправки почты (MSA): Серверная программа, которая получает почту от MUA и подготавливает ее к транспортировке и доставке.



Адрес прежней версии: Адрес электронной почты, который состоит только из символов ASCII. Сравните с интернационализированным адресом.

Адресная книга: Список адресов электронной почты и соответствующих контактных данных.

Алфавит: Совокупность кодовых точек, используемых в языке для письма. В некоторых языках, например в английском, для письма используется один алфавит, в то время как в других, например в японском, для письма используется несколько алфавитов.

Аутентификация: Проверка подлинности источника или другой информации о происхождении сообщения.

Вариантные символы: Символы алфавита с одинаковым значением, но разным визуальным представлением.

Добавление ссылок: Автоматическое форматирование текста для добавления в текст активируемых щелчком мыши ссылок на адреса HTML URL.

Заголовок почтового сообщения: Группа структурированных полей в начале почтового сообщения, предшествующая телу сообщения. Каждое поле начинается с имени поля и двоеточия и состоит из одной или нескольких строк.

Закодированные слова: Формат MIME для отображения текста UTF-8 (помимо адресов) в кодировке ASCII в строках заголовка сообщения. Имеет следующий вид: `=?utf-8?Q?наполнение?=`.

Имена полей: Имена полей в заголовке почтового сообщения, например From: (От:), To: (Кому:) и Subject: (Тема:).

Имя хоста: Запись DNS, которая идентифицирует компьютер в интернете. В именах хостов используются только символы, допустимые в соответствии с правилом LDH.

Композитный символ: Символ, представленный как одна кодовая точка вместо отдельного базового символа, диакритических знаков и других модификаторов.

Конверт: Информация, которая сопровождает сообщение при передаче, в том числе адрес(а) отправки и обратный адрес для сообщений об ошибках и сбоях.

Метаданные: Информация о почтовом сообщении, которая отделена от самого сообщения. В состав метаданных может входить время отправки или получения сообщения, а также сведения о том, открывал ли получатель это сообщение.

Нормализация: Приведение текста к стандартному виду, напр., объединение отдельных символов и диакритических знаков в композитные кодовые точки.

Подготовка строки: Преобразование строки UTF-8 в стандартную форму для повышения удобства обработки.

Пользовательский почтовый агент (MUA): Клиентская программа, с помощью которой пользователь осуществляет отправку, получение и управление почтой. К популярным MUA относятся Outlook и Thunderbird.

Почта прежней версии: Сообщения только с адресами прежней версии. Сравните с почтой EAI.

Почтовая папка: Поименованная группа сообщений. Почтовый ящик может содержать несколько папок.



Почтовый ящик: Место хранения сообщений, связанных с определенным почтовым адресом. Почтовый ящик может быть одной папкой или группой папок.

Правило LDH: Правило, по которому имена хостов DNS могут содержать только буквы, цифры и дефисы. Дополнительные правила запрещают дефис в начале и в конце имени, а также имена с дефисами в третьей и четвертой позициях.

Правило двунаправленности: Запрет на объединение текста, направленного слева-направо и справа-налево в одной метке доменного имени. Сформулировано в RFC 5893, раздел 2.

Тело сообщения: Содержание сообщения, которое следует за заголовками. Может быть неформатированным текстом или одной или несколькими форматированными или закодированными в соответствии с MIME частями.

Транспортный почтовый агент (MTA): Серверная программа, которая отправляет почту на другие хосты и получает почту от других хостов интернета. MTA может получать почту от MSA и/или доставлять почту MDA.

Форма нормализации C (NFC): Профиль Unicode в котором выполняется предварительная композиция символов.