

RNtrack – FTN Messages tracker
ver. 1.XX

Руководство пользователя

Оглавление

| | |
|--|--------------------|
| 1. Для чего эта штука может понадобиться..... | 6 |
| 2. Как она работает..... | 6 |
| 3. Запуск..... | 6 |
| 4. RNtrack.cfg..... | 6 |
| 4.1. Action: SomeAction { [Time] }..... | 7 |
| 4.2. Address: 2:5030/9999..... | 7 |
| 4.3. AddToNullPkt..... | 7 |
| 4.4. Aka: 7:1130/79 7:*/..... | 7 |
| 4.5. APktDir: "Dir" | 7 |
| 4.6. BadMessages: <Mode>..... | 8 |
| 4.7. BadPackets: <Mode>..... | 8 |
| 4.8. BodyMask: * * * | 8 |
| 4.9. CheckPoints: <Mode>..... | 9 |
| 4.10. CreateMissingBase..... | 9 |
| 4.11. Domain: "fidonet" 2:*/..... | 9 |
| 4.12. FileBox: "h:\fbox\alex" 2:5030/9999..... | 9 |
| 4.13. FileBoxDir: "h:\out\fboxes" | 10 |
| 4.14. FileBoxType: <Type>..... | 10 |
| 4.15. FileInbound: "h:\inb\prot" | 10 |
| 4.16. ForceINTL..... | 10 |
| 4.17. IgnoreAttachPath..... | 10 |
| 4.18. IgnoreBSY..... | 10 |
| 4.19. Include: "SomeFile" | 10 |
| 4.20. IndexFile: "o:\nodelist\RNtrack.ndl" | 11 |
| 4.21. KillRoutedMessage: <Mode>..... | 11 |
| 4.22. KludgeMask: * * * | 11 |
| 4.23. LogFile: "c:\log\RNtrack.log" | 12 |
| 4.24. LogLevel: 1..... | 12 |

| | | |
|-----------------------|---|--------------------|
| 4.25. | LoopStr: " 2:5030/9999 RNtrack" | 12 |
| 4.26. | Mask: * * * * * | 13 |
| 4.27. | MaxAge: 2 | 13 |
| 4.28. | MaxAttachSize: 2048 | 13 |
| 4.29. | MaxNodelistAge: 14 | 13 |
| 4.30. | MaxMsgSize: 2048 | 13 |
| 4.31. | MaxPktSize: 2048 | 14 |
| 4.32. | Nodelist: "o:\nodelist\nodelist.*" [Zone] | 14 |
| 4.33. | NodelistPath: "o:\nodelist" | 14 |
| 4.34. | NoLogIgnore | 14 |
| 4.35. | Origin: "My best origin" | 14 |
| 4.36. | Outbound: "h:\tst\outb" | 14 |
| 4.37. | Password: "Pwd" 2:5020/* | 14 |
| 4.38. | ScanDir: | 15 |
| 4.39. | Semaphore: Time "File" | 16 |
| 4.40. | SetViaAlways | 17 |
| 4.41. | ScriptFile: "File" | 17 |
| 4.42. | ScriptMask: "smask" | 17 |
| 4.43. | SkipHiddenFiles | 17 |
| 4.44. | SoftCheckInNodelists | 17 |
| 4.45. | SysopName: "Vladimir Lenin" | 17 |
| 4.46. | Tearline: "Your tear line" | 17 |
| 4.47. | TempMail: "h:\tst\temp" | 18 |
| 4.48. | TimeStampFile: "c:\ftrstamp.bin" | 18 |
| 4.49. | TrafficLog: "h:\log\RNtrack.trf" | 18 |
| 4.50. | TrafficLogTemplate: "File" | 18 |
| 4.51. | UseASO | 18 |
| 4.52. | UseBrake | 18 |
| 4.53. | UseFileBoxes | 18 |
| 4.54. | UseNewVia | 19 |
| 4.55. | UseOwnZone | 19 |

| | | |
|---------|--|----|
| 4.56. | UTC: +3..... | 19 |
| 5. | Параметры слова Mask:..... | 19 |
| 5.1. | FromName, ToName..... | 19 |
| 5.2. | FromAddr, ToAddr..... | 20 |
| 5.3. | Subject..... | 21 |
| 5.4. | Flags..... | 21 |
| 5.5. | Параметры слова 'ScriptMask':..... | 23 |
| 6. | Параметры слова 'Action':..... | 24 |
| 6.1. | Примеры..... | 27 |
| 6.2. | Макросы в шаблонах..... | 28 |
| 7. | Префиксы "*" и "+" для масок..... | 30 |
| 8. | Примеры использования 'Mask:' и 'Action:'..... | 31 |
| 9. | Особенности работы..... | 34 |
| 10. | Поддерживаемые типы баз..... | 34 |
| 10.1. | Squish..... | 35 |
| 10.2. | *PKT..... | 35 |
| 10.3. | *MSG..... | 36 |
| 10.4. | JAM..... | 36 |
| 11. | Работа с нодлистами..... | 36 |
| 11.1. | Адресная маска "#"..... | 37 |
| 11.2. | Адресная маска "&"..... | 38 |
| 12. | Работа со скриптами..... | 38 |
| 12.1. | Константы..... | 39 |
| 12.2. | Переменные..... | 39 |
| 12.3. | Функции..... | 40 |
| 12.3.1. | Log("text");..... | 40 |
| 12.3.2. | Update();..... | 40 |
| 12.3.3. | ExistsInNodelist("99:99/999");..... | 41 |
| 12.3.4. | FindHub("99:99/999");..... | 41 |
| 12.3.5. | NewMsg(...)..... | 41 |
| 13. | Гарантии..... | 41 |

| | | |
|---------------------|------------------------------|--------------------|
| 14. | Благодарности..... | 41 |
| 15. | Как найти разработчиков..... | 42 |

1. Для чего эта штука может понадобиться

Это "трекалка" нетмейла. Предназначена для обработки нетмейла на нетмейловых хабах, хотя может быть полезна и обычному узлу.

2. Как она работает

Берёт письмо, сравнивает его с маской и, если сравнение оказалось успешным, выполняет над данным письмом соответствующую операцию.

3. Запуск

Если запустить RNtrack без параметров, то он попытается найти в текущем каталоге файл `RNtrack.cfg` и обработать его.

Вы можете установить переменную окружения `RNTRACKCONFIG` (или использовать специальный ключ командной строки) для выбора другого пути.

Возможные ключи командной строки:

- c Имя файла конфигурации. Например, -c "d:\soft\fido\RNtrack.cfg". Пробел между '-c' и началом пути обязателен. Кавычки обязательны.
- u Указывает, что требуется распаковка аутбаунда. Перед тем, как выполнить основные функции, описанные в файле конфигурации, RNtrack распакует весь аутбаунд в каталог, указанный в ключевом слове `'TempMail:'`.
- t Выполнить все `'ScanDir:'`, для которых указано однократное выполнение, невзирая на то, должны они сейчас выполняться или нет.
- v Вывести на экран версию программы и список авторов.
- h Вывести на экран подсказку.

4. RNtrack.cfg

Все ключевые слова регистронезависимые. Двоеточие является составной частью ключевого слова. Поэтому нельзя писать "Address :" вместо "Address:". Комментарии могут начинаться с символа '`\`' (до конца строки) или '`(`' (до закрывающей скобки).

Везде, где в качестве параметров используется строка, можно использовать подстановку переменных окружения в виде `%VAR%`. Например:

```
Nodelist: "%NDL%\nodelist.*"
```

Если строку `%..%` не удалось разобрать как переменную окружения, то RNtrack будет

пытаться разобрать её как шестнадцатеричное число - код символа. Например, %22% будет воспринято как ' " '.

Для того, чтобы использовать символ '%' в строках, его надо писать дважды. Например:

```
Action: NewMsg "tpl\badfrom.tpl" "h:\fido\post\netmail\out" "Alex  
Bagmanov" @ "%%" % "Bad source address" +l
```

Ниже приведены все ключевые слова, отсортированные по алфавиту.

4.1. *Action: SomeAction { [Time] }*

Указывает операцию `SomeAction`, которая будет выполнена над письмом, совпавшим с маской. Подробнее об этой операции написано в главе [6](#). Может быть указано время, в которое будет срабатывать данная команда. Формат параметра `Time` аналогичен параметру [Time](#) в команде '`ScanDir:`' и подробно описан в описании этой команды. (см. [Time](#))

4.2. *Address: 2:5030/9999*

Адрес Вашей системы. Должен быть 3D (ну, или 4D, хотя я слабо себе представляю пойнта, который поставит себе RNtrack).

4.3. *AddToNullPkt*

Разрешает допаковывать в файлы ?UT нулевой длины. По умолчанию на такие ?UT RNtrack ругается.

4.4. *Aka: 7:1130/79 7:*/*.**

Ваши АКА. Первым параметром этого слова идёт ваше АКА, вторым -- маска. Маска указывает, при работе с какими адресами будет использован этот ака. Более подробно работа с АКА изложена в описании модификатора адреса '@' (см. [Модификатор '@'](#)) и в описании команды '[Action: Route](#)' (или '[Action: RouteFilebox](#)').

4.5. *APktDir: "Dir"*

Изменяет способ роутинга писем с приаттаченными файлами. По умолчанию данные письма роутятся как обычные (т.е. добавляются в общий ?UT, а приаттаченные файлы в соответствующий ?LO). Если указан `APktDir`, то письма с приаттаченными файлами будут класться в отдельные PKT (одно письмо в одном PKT) в указанном каталоге и прописываться в ?LO файл после приаттаченных файлов. За счёт этого гарантируется, что сначала будут переданы файлы, а уже потом - письмо, к которому эти файлы приаттачены. Каталог, указанный в '`APktDir:`', не должен совпадать с каталогом, указанным в '[Outbound:](#)'.

4.6. **BadMessages: <Mode>**

Указывает, что делать RNtrack-у, если он встретил "плохое" (нулевой длины, с нарушенной структурой, залоченное другим процессом и т.д.) письмо.

| | |
|-------------|---|
| <Mode> Skip | — пропустить "плохое" письмо. |
| Delete | — удалить "плохое" письмо. |
| Move "Dir" | — переместить "плохое" письмо в указанную базу. |
| Exit | — завершить работу Rntrack-а. |

По умолчанию отрабатывает конструкция Exit.

Учтите, что операции "Delete" и "Move" могут повести себя непредсказуемо, если возникла действительно сложная ситуация. Поэтому я рекомендую ими пользоваться только в том случае, если вы абсолютно уверены в том, что Вы делаете.

Дополнительные ограничения:

1. В качестве базы можно указывать базу только в формате *.MSG.
2. Для всех баз, кроме *.MSG операции Delete и Move работают как Skip. Это связано с тем, что неправильно читающиеся письма из этих баз являются первым признаком того, что данная база разрушена. И попытки удалить "плохие" письма могут привести к ещё более серьёзным разрушениям.

4.7. **BadPackets: <Mode>**

Указывает, что делать RNtrack-у, если он встретил "плохой" (нулевой длины, с нарушенной структурой, залоченный другим процессом и т.д.) пакет.

| | |
|-------------|--|
| <Mode> Skip | — пропустить "плохой" пакет. |
| Delete | — удалить "плохой" пакет. |
| Move "Dir" | — переместить "плохой" пакет в указанную базу. |
| Exit | — завершить работу Rntrack-а. |

По умолчанию отрабатывает конструкция Skip.

4.8. **BodyMask: * * ***

Маска, с которой сравнивается тело каждого письма из каталога, указанного '[ScanDir:](#)'.
Порядок параметров такой:

BodyMask: "StringInBody" <Lines> <Bytes>

- StringInBody** — Строка, которая будет искаться в теле письма. Подробнее о формате этой строки можно прочитать в главе 5, в пункте, описывающем маски **FromName** и **ToName** (см. [FromName](#)). Будьте внимательны, маска "Hi" (например) будет означать именно то, что Вы написали. А именно, письмо должно состоять из одной строки "Hi" (причём даже без перевода строки), для того, чтобы совпасть с этой маской. Поэтому правильной маской в этом случае будет "*Hi*".
- [!]Lines** — Сравнивается количество строк в теле письма с заданным. Без знака '!' сравнение идёт на "больше или равно". Со знаком '!' сравнение идёт на "меньше"
- [!]Bytes** — Сравнивается количество байт в теле письма с заданным. Без знака '!' сравнение идёт на "больше или равно". Со знаком '!' сравнение идёт на "меньше".

4.9. **CheckPoints: <Mode>**

Установка режима проверки наличия поинтов в поинтлисте.

- | | | |
|--------------|---|---------------------|
| <Mode> Never | — | Не проверять. |
| Hard | — | Проверять "жёстко". |
| Soft | — | Проверять "мягко". |

Подробнее о действии данного конфигурационного слова читайте в главах [11.1.](#) и [11.2.](#) Если данное конфигурационное слово пропущено, то считается 'Checkpoints: Never'.

4.10. **CreateMissingBase**

Если указанная база отсутствует, то разрешает создание новой. По умолчанию создание новых баз запрещено.

4.11. **Domain: "fidonet" 2:*/*.***

Определяет домен для адреса (адресов). Используется в The Brake!-файл-боксах. По умолчанию: "fidonet".

4.12. **FileBox: "h:\fbox\alex" 2:5030/9999**

Определяет персональный файл-бокс для указанного адреса.

4.13. **FileBoxDir: "h:\out\fboxes"**

Указывает каталог, в котором будут создаваться файл-боксы.

Формат файл-бокса: z.n.f.p, где z — зона, n — сеть, f — узел, p — поинт.

Пример: 2.5030.1157.0 — это файл-бокс узла 2:5030/1157.

4.14. **FileBoxType: <Type>**

Тип файл-боксов

| | | |
|--------|---------|--|
| <Type> | Long | — z.n.f.p (по умолчанию) |
| | Tmlong | — z.n.f.p; z.n.f.p.h если hold (T-Mail long box) |
| | Tmshort | — znf.p; znf.ph если hold (T-mail short box, представление по основанию 32) |
| | Brake | — domain.z.n.f.p.flavour (The Brake!, flavour = hold, normal, direct, crash, immediate). |

Здесь z — зона, n — сеть, f — узел, p — поинт.

4.15. **FileInbound: "h:\inb\prot"**

Указывает каталог, в котором будет выполнен поиск приаттаченных файлов, если в аттаче не указан полный путь. Может быть указан как глобально для всего конфига, так и локально для одного '[ScanDir:](#)'. Если эта команда опущена, то поиск файлов будет выполнен в текущем каталоге.

4.16. **ForceINTL**

Если в письме отсутствует кладж INTL, то этот кладж добавляется при всех перемещениях и изменениях этого письма, см. также '[UseOwnZone](#)'.

4.17. **IgnoreAttachPath**

Игнорировать путь во всех аттачах.

4.18. **IgnoreBSY**

Игнорировать .bsy флаги. При указании данного слова RNtrack не проверяет и не создаёт данные флаги. **Внимание!** Это потенциально опасная команда. Используйте её, только если Вы отчётливо понимаете, зачем и почему используются данные флаги.

4.19. **Include: "SomeFile"**

Подключение дополнительного файла конфигурации. Возможна любая вложенность

файлов. RNtrack начинает обрабатывать это ключевое слово сразу же, как только встретит его, и, обработав, продолжает просмотр основного конфига.

4.20. **IndexFile: "o:\nodelist\RNtrack.ndl"**

Указывает имя файла, в котором будет храниться скомпилированный нодлист. Если это слово опущено, то имя файла будет RNtrack.ndl, и создаваться он будет в текущем каталоге.

4.21. **KillRoutedMessage: <Mode>**

Указывает, что делать с письмом после выполнения команды [Action: Route](#) (или [Action: RouteFilebox](#))

| | | | |
|--------|--------|---|---|
| <Mode> | Always | — | После выполнения команды письмо всегда удаляется. |
| | Never | — | После выполнения команды письмо никогда не удаляется. |
| | Flag | — | Если на письме стоит флаг KillSent (k), то письмо будет удалено, иначе — нет. |

По умолчанию считается, что 'KillRoutedMessage: Always'. Удаление письма выполняется в базе писем, указанной в той команде '[ScanDir:](#)', к которой относится '[Action: Route](#)'. В то же время, это письмо остаётся в памяти и возможна его дальнейшая обработка.

4.22. **KludgeMask: * * ***

Маска, с которой сравниваются кладжи каждого письма из базы, указанной в '[ScanDir:](#)'. Идёт сравнение всех кладжей (и тех, что считаются Kludges, и тех, что считаются Vias). Порядок параметров такой:

KludgeMask: "KludgeName" "KludgeBody" <Times>

| | | |
|------------|---|---|
| KludgeName | — | Строка, с которой будут сравниваться имена кладжей. Подробнее о формате этой строки можно прочитать в главе 5, в пункте, описывающем маски FromName и ToName (см. FromName). |
| KludgeBody | — | Строка, с которой будут сравниваться тела кладжей. |
| [!]Times | — | Количество совпадений с KludgeName и KludgeBody в письме. Без знака '!' сравнение идёт на "больше или равно". Со знаком '!' сравнение идёт на "меньше". Для того, чтобы понять, как |

RNtrack разбирает тот или иной кладж, я рекомендую запустить RNtrack с отладочным журналом и посмотреть результат. Я думаю, что если Вы настолько сильны в FTN технологии, чтобы понять, как работает эта маска и вообще для чего она предназначена, то Вам не составит труда разобраться в отладочном журнале RNtrack-a.

4.23. *LogFile: "c:\log\RNtrack.log"*

Имя файла, в который пишется журнал. Учтите, что использование файла журнала начинается сразу, как только в конфигурационном файле встретится эта команда. Поэтому я рекомендую ставить её в самое начало файла конфигурации. Вы можете не указывать файл журнала.

4.24. *LogLevel: 1*

Уровень детализации журнала. Возможные значения от 0 до 5.

- Уровень 0 — Выдаются только сообщения о серьёзных ошибках и информация по ['Action: Display'](#).
- Уровень 1 — Предупреждения. Дополнительно к сообщениям об ошибках выдаются предупреждения о несуразностях в обрабатываемых письмах и файлах.
- Уровень 2 — Информация. Дополнительно к ошибкам и предупреждениям, выдаются сообщения о всех действиях, вносящих изменения в обрабатываемые объекты.
- Уровень 5 — Вывод отладочной информации. Я не рекомендую пользоваться подобным уровнем в обычных ситуациях, так как при этом файл журнала начинает расти с ужасающей скоростью, да и скорость работы несколько понижается.

4.25. *LoopStr: " 2:5030/9999 RNtrack"*

Указывает строчку, по которой будут ловиться зацикленные письма (лупы) — письма, которые мы отослали, а они, пройдя через другие узлы, снова вернулись к нам. Может быть указана как глобально для всего конфига, так и локально для одного ['ScanDir:.'](#) Эта строка используется как подстрока для поиска в Via. Так как сам RNtrack проставляет Via только при использовании ['Action: Route'](#) (или ['Action: RouteFilebox'](#)), то приходится ловить такие лупы и по чужим Via. И хотя существует стандарт FTS-4009, есть много старых программ, в которых каждый лепит, кто во что горазд. Поэтому для системы с адресом 2:5030/9999 нельзя использовать строку "2 : 5030/9999" как маску, ибо эта подстрока входит в адрес 2:5030/99999 (например). Да и мои собственные пойнты ... В

общем, я использую "2:5030/9999 @", потому что пакую нетмейл самим RNtrack-ом и использую новый формат Via-строки. Для тех, кто использует для этой цели Unimail, я могу порекомендовать использовать строку типа "2:5030/9999, " ибо Unimail ставит в Via строку вида:

UniMail/W32 0.43b 2:5030/9999, 02 Jun 1998 15:39:32 UTC +0400.

4.26. Mask: * * * * *

Маска, с которой сравнивается каждое письмо из каталога, указанного '[ScanDir:](#)'.

Порядок параметров такой:

Mask: FromName FromAddr ToName ToAddr Subject Flags

Подробнее о возможных значениях маски рассказано в главе [5.](#)

4.27. MaxAge: 2

Максимальный возраст письма. Используется совместно с флагом 'g' в '[Mask:](#)'. Может быть указан как глобально для всего конфига, так и локально для одного '[ScanDir:](#)'.

Подробнее об использовании этого параметра рассказано в главе 5 в описании флагов (см. [Flags](#)).

По умолчанию возраст письма, используемый при работе с флагом ' - g', рассчитывается следующим образом: от текущего времени отнимается время написания письма и получается возраст письма в днях. Если дату написания письма не удаётся получить, или оно явно "странное" (ну, например, если верить дате, то письмо написано через 10 лет), то считается что письмо написано только что.

4.28. MaxAttachSize: 2048

Максимальный размер файлов, приаттаченных к письму. Размер указывается в байтах. Используется совместно с флагом 'A' в '[Mask:](#)'. Может быть указан как глобально для всего конфига, так и локально для одного '[ScanDir:](#)'.

Подробнее об использовании этого параметра рассказано в главе 5 в описании флагов (см. [Flags](#)).

4.29. MaxNodeListAge: 14

Максимальный возраст нодлиста, при превышении которого отключается проверка на наличие адреса в нодлисте (будет считаться, что любой адрес присутствует в нодлисте).

Подробнее об использовании этого параметра рассказано в главе [11.](#)

4.30. MaxMsgSize: 2048

Максимальный размер входящего письма. Размер указывается в байтах. Письма в Squish и Jam базах не проверяются.

4.31. *MaxPktSize: 2048*

Максимальный размер входящего pkt файла. Размер указывается в байтах.

4.32. *Nodelist: "o:\nodelist\nodelist.*" [Zone]*

Имя нодлиста. Таких ключевых слов может быть несколько. Если в расширении имени используется символ '*', то берётся файл с наибольшим номером. Если Вы хотите использовать "усечённые" версии нодлиста, то Вы должны указать в данной команде зону (Zone), которая будет использоваться по умолчанию. Скомпилированный нодлист (RNtrack автоматически компилирует нодлист при необходимости) хранится в текущем каталоге или там, где указывает слово 'IndexFile'. Подробнее про работу RNtrack-а с нодлистами написано в главе [11.](#)

4.33. *NodelistPath: "o:\nodelist"*

После этого слова все последующие команды '[Nodelist:](#)' будут использовать указанный каталог как корень, за исключением случая, когда в команде '[Nodelist:](#)' указан полный путь к файлу. Например:

NodelistPath: "o:\nodelist"

Nodelist: "nodelist.*" эквивалентно "o:\nodelist\nodelist.*"

Nodelist: "z7\z7-list.*" эквивалентно "o:\nodelist\z7\z7-list.*"

Nodelist: "c:\addlist.*" эквивалентно "c:\addlist.*"

4.34. *NoLogIgnore*

Отключает вывод информации на экран и в журнал в случае отработывания '[Action: Ignore](#)'. По умолчанию вывод на экран и в журнал включён.

4.35. *Origin: "My best origin"*

Указывает ориджин, который будет использоваться в макросе [@Origin](#).

4.36. *Outbound: "h:\tst\outb"*

Указывает каталог, в котором хранится Binkley Style Outbound. Этот ключ может быть пропущен, если Вы не пользуетесь '[Action: Route](#)' (или '[Action: RouteFilebox](#)'), а маршрутизируете нетмейл другой программой.

4.37. *Password: "Pwd" 2:5020/*.**

Указывает пароль, который будет использоваться при упаковке пакетов на данный адрес. Может быть несколько таких команд. Команда '[Action: Route](#)' (или '[Action:](#)

[RouteFilebox](#)') проверяет адрес получателя пакета, и если он подпадает под маску, то в пакет записывается указанный пароль.

4.38. *ScanDir: ...*

Команда 'ScanDir:' может быть двух видов:

```
ScanDir: "o:\netmail" [Renumber] [Unpack] [Time] [Flag "[!|  
#]File"] [ScriptBefore "Func1"] [ScriptAfter "Func2"]
```

(в конфигурационном файле команда должна размещаться в одной строчке) или

```
ScanDir: @BeforeRoute|@AfterRoute [Flag "[!|#]File"]
```

Указывает базу писем, над которой будет производиться действие. Таких баз может быть несколько. Письма из базы последовательно считываются в память. Дальше каждое письмо можно сравнить с маской командой '[Mask:](#)' и при совпадении с маской можно выполнить над ним действия командами '[Action:](#)' (см. главу [6.](#)). Все команды, указанные ниже данной команды 'ScanDir:', относятся к ней, пока не встретится новая команда 'ScanDir:' или не закончится текст конфигурационного файла.

Если указано дополнительное слово **Renumber** (естественно, без квадратных скобок), то после того, как над базой будут выполнены все указанные действия, все сообщения в базе будут перенумерованы. Перенумерация писем в базе работает только для *.MSG формата.

Если указано **Unpack** — распаковать аутбаунд в указанную базу **перед** тем, как будут выполнены любые действия с базой.

Если указано **Flag "File"**, то сканирование данной базы будет произведено только при наличии соответствующего файла. Если перед именем файла идёт знак '!', то сканирование будет произведено только при **отсутствии** данного файла. Если перед именем файла идёт знак '#', то этот файл будет удалён после сканирования базы.

Формат **Time** следующий:

```
Daily [StartTime[-EndTime]] [P.PeriodTime] [...]
```

```
Weekly [StartDay[-EndDay]] [...]
```

Нумерация дней недели: 0 - воскресенье, 1 — понедельник, ..., 6 — суббота. Время должно быть указано в формате ЧЧ:ММ. Модификаторы времени работают только в том случае, если указан '[TimeStampFile](#)'.

Как это работает:

Daily

Если параметры опущены, то сканируем данную базу **один** раз каждый день (после 00:00).

Daily 17:00

Сканируем базу **один** раз в день, после 17:00.

Daily 17:00-18:00

Сканируем базу при **каждом** запуске в данном промежутке времени.

Daily 08:00 12:00 14:00 20:00-22:00

Сканируем базу **один** раз после 8 утра, **один** раз после 12 дня, **один** раз после 14 дня и **каждый** запуск в промежутке с 20:00 по 22:00.

Daily P.00:15

Сканируем базу **один** раз через каждые 15 минут. Эта запись — сокращённый вариант записи Daily 00:00 00:15 00:30 00:45 01:00... Первое событие при использовании данной формы записи всегда начинается в 00:00. Последнее событие — всегда в пределах 00:00-23:59. Например, при указании P.00:54 первое событие будет в 00:00, второе — в 00:54, последнее — в 23:24.

Weekly

Если параметры опущены, то сканируем данную базу **один** раз каждую неделю (после 00:00 воскресенья).

Weekly 1

Сканируем базу **один** раз в понедельник.

Weekly 2-3

Сканируем базу при **каждом** запуске со вторника по среду включительно.

Weekly 1 3 5

Сканируем базу **один** раз в понедельник, **один** раз в среду и **один** раз в пятницу.

Обратите внимание на то, когда база сканируется **один** раз (если указано только начало события) и когда при **каждом** запуске (если указан диапазон).

Если в качестве базы указано @BeforeRoute или @AfterRoute, то данный 'ScanDir:' будет срабатывать до или после выполнения команд '[Action: Route](#)' (или '[Action: RouteFilebox](#)') в последующих 'ScanDir:'. Использование модификаторов Renumber, Unpack и Time для 'ScanDir:' данного типа невозможно. Также запрещено использование '[Action: Route](#)'.

4.39. Semaphore: Time "File"

Указывает семафор активности RNtrack-a. Если файл "File" существует, значит активна другая копия RNtrack-a. Параметр Time указывает время, в течение которого RNtrack ждёт пропадания этого файла. Время указывается в минутах. Если в качестве времени указано 0, RNtrack ждёт до пропадания файла. Иначе, по истечении указанного времени RNtrack все равно начинает работу.

Пример:

Semaphore: 60 "c:\temp\RNtrack.bsy"

4.40. *SetViaAlways*

По умолчанию RNtrack добавляет собственную строку Via только при выполнении команды "Route". Если использовать данное слово, то Via будет добавляться и при выполнении команд "[Copy](#)" (будет добавлено к копии), "[Move](#)", "[AddNote](#)", "[Rewrite](#)" и других команд, изменяющих само письмо (команда "[Action: Ignore](#)", например, письмо не меняет, поэтому и Via в письмо не добавится). При этом, если Вы несколько раз перекинули письмо из базы в базу, то в письмо попадёт все равно только одна строка Via, и время в этой строке будет последнего переброса/копирования/изменения.

4.41. *ScriptFile: "File"*

Путь к файлу со скриптом. Для вызова функций из этого файла используют [Action: Script](#).

4.42. *ScriptMask: "smask"*

Здесь "smask" — имя функции в скрипте, которая будет вызвана при сравнении письма с данной маской. В зависимости от возвращаемого значения этой функции будет считаться, что письмо совпало или не совпало с маской.

4.43. *SkipHiddenFiles*

При проверке наличия файл-аттачей считать файлы с атрибутом Hidden как отсутствующие. По умолчанию такие файлы видимы.

4.44. *SoftCheckInNodeLists*

Установка "мягкой" проверки наличия узла в нодлисте. Более подробно о том, что делает эта команда можно прочитать в главах [11.1.](#) и [11.2.](#)

4.45. *SysopName: "Vladimir Lenin"*

Ваше имя. Используется в шаблонах как @Sysop. Кавычки обязательны. И вообще, везде, где кавычки стоят, они обязательны.

4.46. *Tearline: "Your tear line"*

Тирлайн, используемый в макросе @Tearline. Если не указан, будет использоваться " - - - Rntrack/XXX X.XX[.bX]".

4.47. TempMail: "h:\tst\temp"

Указывает каталог, в который распаковываются пакеты при использовании функции Unpack (ключ -u в командной строке). Этот ключ может быть пропущен, если Вы не пользуетесь перепакровкой аутбаунда.

4.48. TimeStampFile: "c:\ftrstamp.bin"

Указывает имя файла, который используется для корректной работы функций, срабатывающих в зависимости от времени. Если данного файла нет, он создаётся. Содержимое файла не изменяется и не используется. RNtrack изменяет у данного файла только "время изменения файла".

4.49. TrafficLog: "h:\log\RNtrack.trf"

Указывает файл, в который будет писаться журнал о всех письмах, обработанных командой '[Action: Route](#)' (или '[Action: RouteFilebox](#)'). Данное ключевое слово может быть пропущено, тогда этот журнал вестись не будет.

4.50. TrafficLogTemplate: "File"

Указывает шаблон для журнала трафика. Если данное ключевое слово пропущено, то будет использован внутренний шаблон (он соответствует шаблону, лежащему в примерах с именем traflog.tpl).

4.51. UseASO

Включает альтернативный формат outbound'a — так называемый AmigaDOS Style Outbound. Отличие его от традиционного BSO в том, что используется только одна директория для outbound'a, и файлы в ней имеют формат z.n.f.p.ext. Например:

2.5020.9999.0.flo

2.5020.999.888.hlo

99.333.1.0.out

4.52. UseBrake

Включает The Brake! аутбаунд.

4.53. UseFileBoxes

Использовать файл-боксы вместо *.?lo для приаттаченных файлов. Вы должны определить [FileBoxDir](#) до этой директивы. Сообщение (которое содержит аттач) тоже кладётся в файл-бокс.

4.54. UseNewVia

Использовать новый формат строки Via. Отличия форматов в следующем.

Без 'UseNewVia':

```
"Via 2:5030/9999 RNtrack 2.06/W32 08 Oct 1999 03:03:59 UTC+0400"
```

С использованием 'UseNewVia':

```
"Via 2:5030/9999 @19991007.230359.UTC RNtrack 2.06/W32"
```

Я настойчиво рекомендую использовать 'UseNewVia', потому что этот вариант соответствует стандарту.

Если Вы пользовались старой версией RNtrack, в которой нет этой возможности, и решили перейти на новый формат строки Via, то не забудьте проверить, чтобы у Вас была правильно прописана строка в команде 'LoopStr: ', если, конечно, Вы ей пользуетесь.

4.55. UseOwnZone

Если в письме отсутствует кладж INTL, то считать, что письмо написано в той же зоне, что и мы. При этом выбирается тот наш адрес, маска которого первой совпадёт с адресом отправителя.

4.56. UTC: +3

Конфигурационное слово UTC задаёт разницу (в часах) между UTC и местным временем в компьютере. Для московского времени эта величина +3 зимой и +4 летом. Используется в кладжах при выполнении команды '[Action: Route](#)' (или '[Action: RouteFilebox](#)'). В большинстве случаев RNtrack сам определяет эту разницу, но если в Вашей системе не настроено правильно время (один из признаков этого — компьютер не переводит сам часы на летнее/зимнее время), или Вы хотите по какой-либо причине использовать другую разницу во времени, Вы можете воспользоваться этой командой. Если это ключевое слово пропущено, и системное время не настроено, то в кладжи будет писаться текущее время.

5. Параметры слова Mask :

Ключевое слово 'Mask: ' указывает маску, с которой будет сравниваться письмо. Сейчас мы разберём это слово по косточкам.

Итак, синтаксис этого слова таков:

```
Mask: FromName FromAddr ToName ToAddr Subject Flags
```

5.1. FromName, ToName

Соответствующие текстовые поля из письма. Сравнение регистронезависимое. Можно использовать следующие дополнительные символы:

'*' — любое число любых символов;
'?' — любой символ;
'#' — любая цифра;
'\$' — любой печатный символ (т.е. все символы "больше" пробела);
 '[' и ']' ' — только указанные символы;
 '[' ! ' и ']' ' — только **не** указанные символы.

Если первым символом стоит '!', то проверяется на несовпадение.

Например:

"*Fix*" — проверить на нахождение данной подстроки. Например, "AreaFix", "FixIt" и "Somefixation" подпадут под эту маску.
"*@*. [rs]u" — "eddie@ironmaiden.com" и "some@where.su" подпадут под эту маску, а "some@where.net" — нет.
"!*. *" — соответственно, "aaa.bbb" и "с.с.с.с" **не** подпадут под эту маску, а "SomeText" — подпадёт.

Для того, чтобы использовать символы '*', '?', '[', ']', '!', ' ' в качестве просто символов, а не части маски, их надо предварить символом '\\'.
Например:

"*. \\. *" — под эту маску подпадёт только строка "*. *"

5.2. FromAddr, ToAddr

Соответствующие поля из письма. Могут содержать как маску (например, 2:*/* или просто *), так и символы:

'#' — проверить на наличие в нодлисте;
'H' — Узел есть в нодлисте и у него стоит флаг 'Hold';
'U' — Узел есть в нодлисте и у него стоит флаг 'Hub';
'D' — Узел есть в нодлисте и у него стоит флаг 'Down';
'P' — Узел есть в нодлисте и у него стоит флаг 'Pvt';
'@' — наш(и) адрес(а).

После точки ещё может стоять символ '&'. Например, "2:5030/9999.&". Эта запись означает "Узел, его пойнты, его подхабник (если это хаб), подхостник (если это хост) или весь регион (если это действительно регион)". Подробности работы с масками "#" и "&" читайте в главах [11.1.](#) и [11.2.](#)

Перед адресом может стоять символ '!', что означает '**не**'. Например, ! 2:5030/9999 — адрес **не** должен совпадать с 2:5030/9999.

5.3. Subject

Тема письма. Маски, описанные в полях FromName и ToName (см. [FromName](#)), работают и для этого поля.

5.4. Flags

Флаги письма Возможные значения

| | |
|---|--|
| a | Attach |
| A | Проверить размер аттачей: +A — больше, чем указано в ' MaxAttachSize: '; -A — меньше или равно, чем указано в ' MaxAttachSize: ' |
| b | ARQ |
| c | Crash |
| d | Direct |
| e | Empty (пустое тело письма. Кладжи не учитываются.) |
| E | Echo mail (письмо является эхомейлом) |
| f | File Request |
| g | Проверить возраст письма: +g — больше, чем указано в ' MaxAge: '; -g — меньше или равно, чем указано в ' MaxAge: ' |
| h | Hold |
| i | Forward/in transit |
| j | Kill File Sent |
| k | Kill |
| l | Local |
| L | Lock |
| m | immediate |
| M | Проверить размер сообщений: +M — больше, чем указано в ' MaxMsgSize: '; |

| | |
|-------|--|
| | -M — меньше или равно, чем указано в 'MaxMsgSize:' |
| n | CFM |
| o | Orphan |
| p | Private |
| q | return receipt request |
| r | Received |
| s | Sent |
| S | Archive/Sent |
| t | Truncate file sent |
| u | Update Request |
| v | Scanned (Только для эхо-баз в формате Squish) |
| x | Attach Exists |
| | +X — на письме стоит флаг 'а' и все перечисленные в аттаче файлы присутствуют; |
| | -X — на письме стоит флаг 'а' и один или несколько приаттаченных файлов отсутствуют |
| y | return receipt |
| 0 - 9 | Loop Count |

Если перед флагом стоит знак '+', значит для того, чтобы письмо совпало с маской, этот флаг должен быть установлен; если стоит знак '-', то флаг должен быть сброшен. Особая ситуация с Loop. Флаг +0 указывает, что ловится ситуация, когда письмо ещё ни разу не проходило через нашу станцию. Флаг -0 ловит ситуацию, когда письмо проходило через нашу станцию хотя бы один раз. Флаг +1...+9 ловит ситуацию, когда письмо проходило соответствующее число раз **или более**. Флаг -1...-9 — когда письмо проходило **менее** этого числа раз.
То есть +0 == -1, -0 == +1.

Примеры некоторых масок:

Mask: * * * * *

Под такую маску подпадает **любое** письмо.

Mask: * * "*Lenin*" * * *

Под такую маску подпадают те письма, которые написаны на имя, в котором есть слово Lenin.

Mask: * !# * !# * *

Под такую маску подпадают те письма, отправитель и получатель которых отсутствуют в нодлисте.

Mask: * * * * * +s

Под такую маску подпадают письма с установленным флагом Sent.

Масок может быть несколько подряд. Например:

Mask: * 2:5020/0 * * * *

Mask: * 2:5020/381 * * * *

Mask: * 2:5020/1381 * * * *

Под такую маску подпадают письма, написанные с адреса 2:5020/0, или 2:5020/381, или 2:5020/1381.

Модификатор '@' может быть использован в ситуации, когда надо указать в маске все наши адреса. Например:

Address: 2:5030/9999

Aka: 7:1130/79 7:*/*.*

Aka: 8:888/888 8:*/*.*

[...]

Mask * * * !@ * *

Под такую маску подпадут все письма, которые написаны не на 2:5030/9999, не на 7:1130/79 и не на 8:888/888.

5.5. Параметры слова '*ScriptMask:*'

Синтаксис:

ScriptMask: "smask"

Здесь "smask" - имя функции в скрипте, которая будет вызвана при сравнении письма с данной маской. В зависимости от возвращаемого значения этой функции будет считаться, что письмо совпало или не совпало с маской.

| код возврата | результат |
|--------------|-----------|
|--------------|-----------|

| | |
|---|---------------------------------|
| 0 | сообщение не совпадает с маской |
|---|---------------------------------|

1 сообщение совпадает с маской

Если функция возвращает 1, то считается, что письмо совпадает с маской. В противном случае письмо не совпадает с маской.

6. Параметры слова 'Action:'

Ключевое слово 'Action:' указывает, что мы будем делать с письмом, совпавшим с маской. Возможные варианты этого слова:

Action: AddNote "tpl"

Добавить в начало тела письма указанный шаблон.

Action: AddKludge "name" "value"

Добавить кладж в письмо.

Action: Delete

Удалить письмо. Письмо удаляется из базы, указанной в 'ScanDir:', к которой относится данная команда. В то же время, письмо остаётся в памяти и возможна его дальнейшая обработка другими командами 'Action:', относящимися к той же команде 'ScanDir:'.

Action: Move "o:\new\dir"

Переместить письмо в другую базу. После перемещения возможна обработка этого письма другими командами 'Action:', относящимися к той же команде 'ScanDir:'.

Action: Copy "o:\new\dir"

Скопировать письмо в другую базу.

Action: Rewrite * * * * *

Перезаписать письмо. Значения полей, в которых в строчке выше стоят звёздочки, те же, что и в [Mask:](#). Символ '*' означает, что данное поле надо оставить без изменений. Письмо из памяти записывается в базу писем, указанную в ['ScanDir:'](#). В поле Subject можно использовать макросы, используемые в шаблонах.

Action: Ignore

Проигнорировать это письмо.

Action: Display "Text"

Вывести строку на экран и в журнал. В поле "Text" можно использовать макросы, используемые в шаблонах.

Action: Flag "File"

Создать файл нулевой длины. Если каталог, в котором должен быть создан флаг, отсутствует, то он создаётся.

Action: DelFile "File"

Удалить указанный файл. Если файл уже отсутствует, то сообщение об ошибке не выдаётся.

Action: NewMsg "tpl" "dir" * * * * *

Создать новое письмо, используя шаблон. В поле Subject можно использовать макросы, используемые в шаблонах. "tpl" в этом слове указывает на шаблон, который будет использован при создании нового письма, а "dir" — это каталог, в котором это письмо будет создано. Остальные поля почти такие же, как в слове **'Mask:'** за исключением того, что здесь нельзя использовать символы '&', '!' и '#', но есть три других символа — это '\$', '%' и '@'. Символ '%' указывает, что в полях From или To надо использовать адрес или имя получателя, указанные в оригинальном письме. Символ '\$' указывает, что надо использовать адрес или имя отправителя. Символ '@' в адресе указывает, что надо использовать наш адрес (или AKA).

Action: WriteFile "tpl" "File"

Создать файл "File" и записать в него текст, созданный из шаблон "tpl".

Action: AppendToFile "tpl" "File"

Если файл "File" уже существует, то текст, созданный из шаблона "tpl", дописывается к концу этого файла, иначе — аналогично [WriteFile](#).

Action: Call "tpl" "File" "Prg"

Создать файл "File", используя шаблон "tpl" и затем вызвать программу "Prg". После её завершения удалить созданный файл. Параметры "tpl" и "File" могут отсутствовать. В поле "Prg" можно использовать макросы, используемые в шаблонах.

Action: Route <Mode> Address

Отправить это письмо через соответствующий адрес. Mode может быть Hold, Crash, Direct, Normal или Immediate. В качестве адреса могут быть использованы следующие символы:

% — отправить через самого получателя письма;

%.0 — отправить через босса получателя письма.

В качестве адреса отправителя пакета берётся тот Ваш адрес, который подпадает под соответствующую маску.

Action: RouteFilebox <Mode> Address

Отправить это письмо через соответствующий адрес. Mode может быть Hold, Crash, Direct, Normal или Immediate. В качестве адреса могут быть использованы следующие символы:

% — отправить через самого получателя письма;

`%.0` — отправить через босса получателя письма.

В качестве адреса отправителя пакета берётся тот Ваш адрес, который подпадает под соответствующую маску. Письмо кладётся в виде отдельного пакета в файл-бокс.

Action: RouteHub <Mode>

Отправить это письмо через хаб адреса назначения. Mode может быть Hold, Crash, Direct, Normal или Immediate. Отправляет на хост получателя если хаб не найден.

Action: Poll <Mode> Address

Сгенерировать Poll на указанный адрес. Аргументы Mode и Address аналогичны таким же аргументам в команде '[Action: Route](#)' (или '[Action: RouteFilebox](#)'). В качестве Poll-а создаётся соответствующий ?L0 файл нулевой длины. Если данный ?L0 файл уже существует, команда не выполняет никаких действий.

Action: DeleteAttach

Удаляет аттачи, то есть файлы, приложенные к письму. Само письмо не трогается. Если на письме нет флага FileAttach, то данная команда игнорируется.

Action: ChangePath "Dir"

Изменяет путь файлов в теме письма с приложенными файлами. Сами файлы никуда не перемещаются. Наличие такого каталога не проверяется. Если в качестве каталога указать " ", то путь просто будет удалён. Если сумма длин всех новых имён превышает 71 символ (максимальная длина поля темы), то выдаётся предупреждение, и письмо остаётся без изменений. Если на письме нет флага FileAttach, то данная команда игнорируется.

Action: MoveAttach "Dir"

Перемещает присоединённый(е) файл(ы) в указанный каталог. В теме письма записывается новый путь. Если на письме нет флага FileAttach, то данная команда игнорируется.

Action: MoveAttachFilebox <Mode> Address

Перемещает присоединённый(е) файл(ы) в файл-бокс для указанного адреса. Аргументы Mode и Address аналогичны таким же аргументам в команде '[Action: Route](#)' (или '[Action: RouteFilebox](#)' и т.д.). В теме письма записывается новый путь. Если на письме нет флага FileAttach, то данная команда игнорируется.

Action: CopyAttach "Dir"

Копирует присоединённый(е) файл(ы) в указанный каталог. Тема письма не изменяется. Если на письме нет флага FileAttach, то данная команда игнорируется.

Action: CopyAttachFilebox <Mode> Address

Копирует присоединённый(е) файл(ы) в файл-бокс для указанного адреса. Аргументы Mode и Address аналогичны таким же аргументам в команде '[Action: Route](#)' (или '[Action: RouteFilebox](#)'). Тема письма не изменяется. Если на письме нет флага FileAttach, то данная команда игнорируется.

Action: Script "Func"

Вызывает функцию "Func" из файла, определённого в переменной [ScriptFile](#).

Action: Split <Lines>

Разрезает указанное письмо на несколько, с количеством строк в теле письма не большим Lines. Само письмо после этой операции уничтожается. В начало каждого письма добавляется кладж "^aSPLIT: By Rntrack. Part N of M", где N — номер текущей части, а M — сколько всего частей получилось.

Action: ToLowerPath

Конвертирует в теме письма путь и имя присоединённого(ых) файла(ов) в нижний регистр.

Action: ToUpperPath

Конвертирует в теме письма путь и имя присоединённого(ых) файла(ов) в верхний регистр.

Action: Recode "Recode.tbl"

Перекодирует письмо в соответствии с таблицей перекодировки в файле "Recode.tbl". Формат файла перекодировки совместим с hpt. Примеры таких файлов смотрите в каталоге samples дистрибутива.

6.1. Примеры

Action: Copy "o:\new\dir"**Action: Delete**

Выполняет ту же операцию, что 'Action: Move "o:\new\dir"'

Action: Rewrite * * * * * -cdm

Сбрасывает у письма атрибуты Crash, Direct и Immediate.

Action: Route Direct 2:5020/1

Отправляет письмо через узел 2:5020/1 с флавором Direct.

**Action: NewMsg "tpl\badfrom.tpl" "o:\netmail\bout"
"Vladimir Lenin" @ "%%" % "Bad source address" +l**

(В конфигурационном файле это должно быть написано в одну строку, сейчас эта команда разбита на две строки просто для удобства чтения). Эта команда, используя

шаблон, находящийся в "tpl\badfrom.tpl", создаёт новое письмо в каталоге "o:\netmail\bout", написанное от имени "Vladimir Lenin" с адреса 2:5030/9999 на имя и адрес получателя оригинального письма с темой "Bad source address" и установленным флагом Local.

6.2. Макросы в шаблонах

В шаблонах можно использовать следующие макросы:

| | |
|---------------|--|
| @ToName | Имя получателя оригинального письма. |
| @ToAddr | Адрес получателя оригинального письма. |
| @InetToAddr | Адрес получателя оригинального письма (в стиле internet). |
| @ToZone | Зона получателя письма. |
| @ToNet | Сеть получателя письма. |
| @ToNode | Узел получателя письма. |
| @ToPoint | Пойнт получателя письма. |
| @FromName | Имя отправителя оригинального письма. |
| @FromAddr | Адрес отправителя оригинального письма. |
| @InetFromAddr | Адрес отправителя оригинального письма (в стиле internet). |
| @FromZone | Зона отправителя оригинального письма. |
| @FromNet | Сеть отправителя оригинального письма. |
| @FromNode | Узел отправителя оригинального письма. |
| @FromPoint | Пойнт отправителя оригинального письма. |
| @Subject | Тема оригинального письма. |
| @MsgDate | Дата написания оригинального письма. |
| @MsgAge | Возраст письма. |
| @MsgHeader | Заголовок оригинального письма. |
| @MsgKludges | Кладжи оригинального письма. |
| @MsgBody | Тело оригинального письма. |

| | |
|----------------|---|
| @MsgVias | Via и прочие технические строки в конце оригинального письма. |
| @MsgAttr | Атрибуты письма. |
| @MsgLoops | Loop Counter. |
| @MsgAreaName | Если письмо — эхомейл, то название эхообласти, из которой это письмо. |
| @MsgAttachSize | Размер всех приаттаченных файлов. |
| @MsgSize | Размер тела письма в байтах (без учёта кладжей). |
| @MsgID | MSGID письма. |
| @MsgLines | Количество строк тела письма (без учёта кладжей). |
| @MsgTearline | Тирлайн письма. Добавляется инвалидированным. |
| @MsgOrigin | Ориджин письма. Добавляется инвалидированным. |
| @MsgNTearline | Тирлайн письма. Добавляется не инвалидированным. Если в письме нет тирлайна, создаётся пустой. |
| @MsgNOrigin | Ориджин письма. Добавляется не инвалидированным. Если в письме нет ориджина, создаётся пустой. |
| @MsgRoutedVia | Если данный макрос используется в шаблоне, вызываемом внутри 'ScanDir: @AfterRoute @BeforeRoute ', то вместо него подставится адрес, на который роутится данное письмо. |
| @MsgReply | Содержимое кладжа 'Reply' письма. |
| @Sysop | Ваше имя, указанное в конфигурации. |
| @MyAddr | Ваш адрес, указанный в конфигурации. |
| @MaxAge | Значение ' MaxAge: '. |
| @MaxAttachSize | Значение ' MaxAttachSize: ' |
| @Date | Текущая дата. |
| @NodeLists | Список используемых нодлистов. |
| @Origin | Ориджин, указанный в конфигурации. Добавляется не инвалидированным. |

| | |
|-----------|--|
| @Tearline | Тирлайн, указанный в конфигурации. Добавляется не инвалидированным. Если в конфигурации не указан, то добавляется "- - - RNtrack" и версия. |
| @Version | Версия RNtrack. |
| @Pid | PID RNtrack (только под UNIX). |

7. Префиксы "*" и "+" для масок

Маски '[Mask:](#)', '[BodyMask:](#)', '[KludgeMask:](#)' и '[ScriptMask:](#)' могут быть с префиксом '*' или '+'. Например, '*Mask:' или '+BodyMask:'. Для чего они предназначены?

Префикс '+' используется для объединения масок. Например, запись:

```
Mask: "a" * * * * *
+Mask: * * "b" * * *
+Mask: * * * 2:5030/9999.0 * *
```

эквивалентна маске:

```
Mask: "a" * "b" 2:5030/9999.0 * *
```

Этот пример несколько надуман и вряд ли применим в реальном конфигурационном файле, но он хорошо демонстрирует работу данного префикса. А вот пример более жизненный:

```
Mask: * * "@*" 2:5020/400.0 * *
+BodyMask: * * 4096
```

С этой маской совпадут письма, имеющие в имени 'To:' знак '@', адрес 'To:' — 2:5020/400 и размер письма больше 4 килобайт.

Префикс '*' используется для указания RNtrack-у, что даже если письмо совпало с данной маской, всё равно требуется сравнение с последующими масками. Для объяснения работы данного префикса рассмотрим следующий пример:

```
Mask: * * * * *
Action: Display "Mask one"
Mask: * * * * *
Action: Display "Mask two"
```

При такой конфигурации мы никогда не увидим на экране строку "Mask two". Ибо первая маска будет перехватывать все письма. А вот если изменить конфигурационный файл так:

```
*Mask: * * * * *
Action: Display "Mask one"
```

```
Mask: * * * * *
```

```
Action: Display "Mask two"
```

мы увидим и первую строку, и вторую.

И последнее. Если в конфиге существует запись типа:

```
*Mask: ...
```

```
+Mask: ...
```

```
+Mask: ...
```

она означает, что письмо должно совпасть со всеми тремя масками, но при этом письмо будет обработано и другими масками.

Более развёрнутые примеры работы с данными префиксами Вы увидите в следующей главе.

8. Примеры использования 'Mask:' и 'Action:'

Сделать себе копию всех писем, снять с писем флаг Direct и письма, адресованные на гейт '2:5020/400' и превышающие размер в 4 килобайта, переадресовать на гейт '2:5020/204':

```
*Mask: * * * * *
```

```
Action: Copy "c:\netmail\archive"
```

```
*Mask: * * * * * +d
```

```
Action: Rewrite * * * * * -d
```

```
Mask: * * "@*" 2:5020/400.0 * *
```

```
+BodyMask: * * 4096
```

```
Action: Rewrite * * * 2:5020/204.0 * *
```

Удалить все письма, которые адресованы не мне и не моим пойнтам:

```
Mask: * * * 2:5030/9999.0 * *
```

```
Mask: * * * 2:5030/9999.1 * *
```

```
Mask: * * * 2:5030/9999.3 * *
```

```
Mask: * * * 2:5030/9999.5 * *
```

```
Action: Ignore
```

```
Mask: * * * * * *
```

```
Action: Delete
```

Роутить письма для 2:5030/9999.* с 00:00 по 11:59 на 2:5020/238, а с 12:00 по 23:59 — директом на 2:5030/9999.0:

```
Mask: * * * 2:5030/9999.* * *
```

```
Action: Route Crash 2:5020/238 [Daily 00:00-11:59]
```

Action: Route Crash 2:5030/9999 [Daily 12:00-23:59]

Удалить все письма, у которых и отправитель и получатель отсутствуют в нодлисте:

Mask: * !# * !# * *

Action: Delete

У всех писем, которые написаны на адрес 2:5030/9999.1, или которые написаны на адрес 2:5030/9999.0 и в имени To которых присутствует "Lenin", "Eddie" или "Moderator" сбросить флаги Crash, Direct, Immediate, переадресовать на адрес 2:5030/9999 на имя "Vladimir Lenin" и положить в каталог h:\netmail.

Mask: * * * 2:5030/9999.1 * *

Mask: * * "*Lenin*" 2:5030/9999 * *

Mask: * * "*Eddie*" 2:5030/9999 * *

Mask: * * "*Moderator*" 2:5030/9999 * *

Action: Rewrite * * "Vladimir Lenin" 2:5030/9999 * -cdm

Action: Move "h:\netmail"

Отправить все письма для сети 2:5030 через 2:5030/251

Mask: * * * 2:5030/*. * * *

Action: Route Crash 2:5030/251

Снять со всех писем флаг директ и отправить письма с узла 2:5020/1 на узел 2:5020/0

*Mask: * * * * * +d

Action: Rewrite * * * * * -d

Mask: * 2:5020/1 * * * *

Action: Route Direct 2:5020/0

Отправить письма для всех пойнтов узла 2:5030/9999 на них директом. Но не отправлять почту на сам этот узел. На самом деле это пример рассылки почты пойнтам на этом узле.

Mask * * * 2:5030/9999.0 * *

Action: Ignore

Mask: * * * 2:5030/9999.* * *

Action: Route Direct %

Отправить все письма для 2:5020/50.*, /78.*, /204.* и /1301.* на соответствующие **узлы**. (Так называемый "Босс-роутинг").

Mask * * * 2:5020/50.* * *

Mask * * * 2:5020/78.* * *

Mask * * * 2:5020/204.* * *


```
Mask * * * 2:5020/1301.* * *
```

```
Action: Route Direct %.0
```

Отправить весь подхабник 2:5020/50 на 2:5020/50, подхостник 2:5020/0 на 2:5020/1381, письма на 2:5030/9999.0 не трогать, а весь подхабник 2:5030/9999 и его пойнтов отослать на них директом.

```
Mask * * * 2:5020/50.& * *
```

```
Action: Route Direct 2:5020/50
```

```
Mask * * * 2:5020/0.& * *
```

```
Action: Route Direct 2:5020/1381
```

```
Mask * * * 2:5030/9999.0 * *
```

```
Action: Ignore
```

```
Mask * * * 2:5030/9999.& * *
```

```
Action: Route Direct %
```

Отправить весь регион 2:50 на 2:5020/204, весь регион 2:46 на 2:46/0, а все остальное — на 2:5020/278.

```
Mask * * * 2:50/0.& * *
```

```
Action: Route Direct 2:5020/204
```

```
Mask * * * 2:46/0.& * *
```

```
Action: Route Direct 2:46/0
```

```
Mask * * * * * *
```

```
Action: Route Direct 2:5020/278
```

Пример работы с лупами. Ловим весь нетмейл, который проходил через нашу станцию один и более раз и откладываем его в луполовку. Раз в сутки все письма из луполовки выпускаем наружу. Если-же письмо прошло через нашу станцию уже семь раз (а это значит, что за семь дней роутинг так и не выправился), то разворачиваем его отправителю.

```
ScanDir: "o:\netmail\loops" Daily 01:00
```

```
Mask: * * * * * *
```

```
Action: Move "0:\netmail\out"
```

```
...
```

```
ScanDir: "o:\netmail"
```

```
Mask: * * * * * +7
```

```
Action: NewMsg "tpl\Loop.tpl" "o:\netmail\out" "Vladimir  
Lenin" @ "$" $ "Loop detected" +l
```

```
Action: Delete
```

```
Mask: * * * * * +1
```

```
Action: Move "o:\netmail\loops"
```

```
...
```

```
Mask: * * * * *
```

```
Action: Move "o:\netmail\out"
```

Пример обработки флага +b (ARQ)

```
ScanDir: @AfterRoute
```

```
Mask: * * * * * +b
```

```
Action: NewMsg "Audit.tpl" "o:\netmail" "RNtrack" 2:5030/9999  
$ $ "Audit request receipt" +l
```

```
ScanDir: "o:\netmail"
```

```
Mask: * * * * *
```

```
Action: Route Crash 2:5030/9999
```

9. Особенности работы

При выполнении команд [Move](#), [Copy](#), [Rewrite](#), [ChangePath](#), [MoveAttach](#) результирующее письмо может иметь другой размер. Это связано с тем, что при выполнении данных команд письмо создаётся заново, и в него могут быть дописаны недостающие кладжи, переводы строк и тому подобное.

При работе с аттачами RNtrack воспринимает пробел в поле темы письма как разделитель между именами файлов. Если приаттаченный файл содержит пробел в имени, такое имя файла будет воспринято как два имени файла (до пробела и после). В результате файл не будет обработан.

Ещё одна особенность, при выполнении команды [Delete](#) письмо сразу удаляется, но при этом оно остаётся в памяти. Поэтому последовательность команд [Copy](#) [Delete](#) и [Delete](#) [Copy](#) сделают одно и тоже — [Move](#). Но при этом в первом случае сначала будет создано новое письмо, а потом удалено старое, а во втором случае сначала будет удалено старое письмо, а потом создано новое.

Да, имейте в виду, RNtrack сам не снимет и не поставит ни единого флага на письме. Он не удалит ни одного письма или файла без специальной на то просьбы. Так что, если вы хотите, чтобы на сканированных письмах появился флаг +s после отправки (флаг Sent), или чтобы письма с атрибутом KillSent удалялись, то вы должны сами об этом позаботиться.

10. Поддерживаемые типы баз

RNtrack на настоящий момент умеет работать со следующими типами баз:

1. Squish

2. *.PKT

3. *.MSG

4. JAM

10.1. *Squish*

Формат имени: "\$BaseName"

Первым символом в имени должен идти '\$'. В качестве имени базы указывается только имя файла, без расширения.

Примеры:

ScanDir: "\$p:\mymail"

Action: Move "\$p:\outmail"

Ограничения:

- Не может использоваться в качестве базы для BadMail.
- Операция Renumber не выполняет никаких действий.

10.2. **.PKT*

Формат имени: "#[#]Directory[\File][!FtnAddressMask]"

Первым символом в имени должен идти '#'. В качестве имени базы может указываться как каталог (тогда будут просматриваться все файлы совпадающие с маской *.pkt), так и имя конкретного файла. Если в имени базы указан второй знак '#', то пароли проверяться не будут, иначе при открытии каждого *.pkt проверяется совпадение пароля с указанным в команде '[Password:](#)'. Дополнительно можно указать FTN маску (через символ '!'), тогда будет дополнительно проверяться совпадение адреса отправителя пакета с указанной маской. Если пароль или адрес в пакете не совпадает с указанным, то данный пакет просто пропускается. Если в результате операций, указанных в '[ScanDir:](#)', пакет оказывается пустым, он удаляется.

Примеры:

ScanDir: "##o:\inbound"

ScanDir: "#o:\inbound!2:5030/9999"

ScanDir: "#o:\inbound\139c004f.pkt!2:5030/9999"

Ограничения:

- Не может использоваться в качестве базы для BadMail.
- Операция Renumber не выполняет никаких действий.
- Эта база поддерживает только операции чтения письма и удаления письма. То есть **любые** операции, которые изменяют содержимое письма, запрещены.

10.3. *.MSG

Формат имени: "Directory"

Указывается каталог, в котором лежат письма. Имя каталога должно быть без завершающего "\" за исключением ситуации, когда указывается корневой каталог.

Примеры:

```
ScanDir: "o:\netmail"
```

```
Action: Move "o:\netmail\out"
```

Ограничения:

- Если в базе, которая в данный момент сканируется, появились новые письма, RNtrack их не увидит до следующего запуска. Причём это касается и писем, созданных самим RNtrack-ом.

10.4. JAM

Формат имени: "@BaseName"

Первым символом в имени должен идти '@'. В качестве имени базы указывается только имя файла, без расширения.

Примеры:

```
ScanDir: "@p:\mymail"
```

```
Action: Move "@p:\outmail"
```

Ограничения:

- Не может использоваться в качестве базы для BadMail.
- Операция Renumber не выполняет никаких действий.

11. Работа с нодлистами

RNtrack пользуется нодлистами и пойнтлистами при выполнении сравнения с адресными масками "#" и "&". Сравнение происходит немного по-разному при использовании конфигурационных слов "[SoftCheckInNodelists](#)" и "[CheckPoints: <Mode>](#)". Как именно влияют эти слова, можно прочитать в главах [11.1.](#) и [11.2.](#)

RNtrack понимает как "полные" версии нодлиста (со всеми зонами и регионами), так и "усечённые" версии (региональные и сетевые). Единственное требование при использовании усечённых версий нодлиста — явно указывать зону в команде "[Nodelist:](#)", иначе RNtrack откажется компилировать данный нодлист.

Пойнтлист также считается полноправной версией нодлиста, и с ним RNtrack тоже работает. Пойнтлист должен быть следующего формата:

```
Boss,2:5030/9999
```

```
Point,1,PointOne,Moscow,Sysop,-Unpublished-,9600,XA,M0,ZYX,V34
```

```
Point,2,PointTwo,Moscow,Sysop2,-Unpublished-,2400,M0
```

```
...
```

Пойнтлисты должны подключаться после нодлистов. То есть вам сначала надо перечислить все нодлисты, а только потом пойнты. Если использовано конфигурационное слово "[MaxNodelistAge:](#)", то проверяется возраст всех используемых нодлистов (возраст проверяется по времени создания файла). И, если хотя бы один нодлист оказался старше, чем указано в данном слове, то отключаются любые проверки наличия узла в нодлисте. Считается, что любой узел и любой поинт всегда находится в нодлисте/пойнтлисте.

Если используется два или более "перекрывающихся" нодлиста (например полный нодлист и нодлист сети), то объединение этих нодлистов идёт по следующим правилам:

- Узел считается существующим, если он есть хотя-бы в одном нодлисте.
- Если в одном нодлисте узел хаб, а в другом нет, то считается, что он хаб.

11.1. Адресная маска "#"

Адресная маска "#" используется в слове "[Mask:](#)" и означает проверку на наличие (или отсутствие) узла отправителя и/или получателя в нодлисте.

Просто маска "#" - "жёсткая" проверка наличия узла в нодлисте. Узел считается присутствующим в нодлисте тогда и только тогда, когда он действительно там есть.

"#" + "[SoftCheckInNodelists](#)" — "мягкая" проверка наличия узла в нодлисте. Узел считается присутствующим в нодлисте тогда, когда он там действительно присутствует (странно, правда? :), или тогда, когда в нодлисте нет сети и/или зоны данного узла. То есть если Вы будете использовать сетевой сегмент нодлиста и маску "#" + "[SoftCheckInNodelists](#)", то наличие узла в Вашей сети будет проверяться жёстко, а узлы во всех остальных сетях проверяться не будут — считаем, что они существуют.

"[CheckPoints:](#)" вносит разнообразие в определение существования поинтов:

| | |
|----------------------|---|
| "CheckPoints: Never" | Не проверять наличие поинтов. Поинт считается существующим если существует его "босс". |
| "CheckPoints: Hard" | Проверять наличие поинтов по пойнтлисту. Если поинт отсутствует в пойнтлисте или отсутствует его "босс", то считается, что поинт не существует. |
| "CheckPoints: Soft" | Проверять наличие поинтов по пойнтлисту. Отличие от предыдущего варианта: если в пойнтлисте отсутствует |

сегмент "босса" пойнта, то считается, что пойнт существует. Наличие "босса" пойнта в нодлисте по-прежнему обязательно.

11.2. Адресная маска "&"

Адресная маска "&" используется в слове **Mask:** и означает проверку на вхождение адреса отправителя и/или получателя в "подбоссник" (если в качестве маски используется обычный узел), "подхабник" (если узел — хаб), "подхостник" (если узел является хостом сети) или в регион (если узел является региональной записью). Обратите внимание на различие масок "2:5020/0.*", "2:5020/*" и "2:5020/0.&". Первая маска означает "Все пойнты узла 2:5020/0". Вторая маска — "Все узлы сети 5020". Третья — "Все узлы, находящиеся в подхостнике". В случае с регионом 2:50 и масками "2:50/0.*", "2:50/*", "2:50/0.&" расшифровка такая: первая маска — "все пойнты узла 2:50/0", вторая маска — "все независимые узлы в 50 регионе", третья маска — "все узлы в 50 регионе". Учтите, что под адресную маску "&" не попадают узлы отсутствующие в нодлисте. То есть, если узла "2:5020/1079" нет в нодлисте, то он не совпадёт с маской "2:50/0.&".

Конфигурационное слово **SoftCheckInNodelists** не оказывает на данную маску никакого влияния.

Конфигурационное слово **CheckPoints:** оказывает следующее влияние:

| | |
|----------------------|--|
| "CheckPoints: Never" | Не проверять наличие пойнта в пойнтлисте. С маской "2:5030/9999.&" совпадут все пойнтовые адреса. |
| "CheckPoints: Hard" | Проверять наличие пойнтов по пойнтлисту. С маской "2:5030/9999.&" совпадут только те пойнтовые адреса, которые описаны в пойнтлисте. |
| "CheckPoints: Soft" | Проверять наличие пойнтов по пойнтлисту. С маской "2:5030/9999.&" совпадут только те пойнтовые адреса, которые описаны в пойнтлисте или все адреса, если сегмент 2:5030/9999 не описан в пойнтлисте. |

12. Работа со скриптами

RNtrack использует Perl в качестве скриптового языка. Если Вы компилируете RNtrack самостоятельно, для сборки с поддержкой скриптов запустите make (gmake, nmake) с параметром **ENABLE_SCRIPTS=1**. Если Вы взяли готовый скомпилированный RNTrack, поддерживает ли он встроенный Perl и с какой версией Perl он скомпилирован, можно увидеть в выводе программы, если запустить RNTrack с параметром -v.

Ниже приведены константы, переменные и функции, определённые в скриптах. Здесь

%FMSG — хэш, содержащий текущее сообщение.

12.1. Константы

| | |
|------------------------|--|
| \$ConfigFile | Путь к конфигурационному файлу |
| \$LogFile | Путь к журналу |
| \$LogLevel | Уровень детализации журнала |
| \$SysopName | Имя сисопа |
| \$TrafficLog | Путь к журналу трафика |
| \$Version | Версия RNtrack |
| \$FMSG{"InetFromAddr"} | Адрес отправителя оригинального письма (в стиле internet). |
| \$FMSG{"InetToAddr"} | Адрес получателя оригинального письма (в стиле internet). |
| \$FMSG{"Date"} | Дата написания оригинального письма. |
| \$FMSG{"Age"} | Возраст письма. |
| \$FMSG{"Lines"} | Количество строк тела письма (без учёта кладжей). |

%FKludges — кладжи текущего сообщения. Значение каждого ключа этого хэша — ссылка на массив. Для получения значения не повторяющегося кладжа (все кроме Via и SEEN-BY) используйте этот код:

```
$kludgevalue=$FKludges{"KludgeName"}[0];
```

Внимание! Вы должны использовать все спецсимволы в имени кладжа (0x01, ":" и т.д.).

12.2. Переменные

| | |
|--------------------|---|
| \$FMSG{"FromName"} | Имя отправителя оригинального письма. |
| \$FMSG{"FromAddr"} | Адрес отправителя оригинального письма. |
| \$FMSG{"ToName"} | Имя получателя оригинального письма. |
| \$FMSG{"ToAddr"} | Адрес получателя оригинального письма. |
| \$FMSG{"Subject"} | Тема оригинального письма. |
| \$FMSG{"Attr"} | Атрибуты письма. |

| | |
|----------------------------------|---|
| <code>\$FMSG{"Cost"}</code> | Cost |
| <code>\$FMSG{"TimesRead"}</code> | TimesRead |
| <code>\$FMSG{"ReplyTo"}</code> | ReplyTo |
| <code>\$FMSG{"NextReply"}</code> | NextReply |
| <code>\$FMSG{"AreaName"}</code> | Если письмо эхомейл, то название эхообласти, из которой это письмо. |
| <code>\$FMSG{"Body"}</code> | Тело оригинального письма. |

12.3. Функции

12.3.1. **Log("text");**

Вывод текста в журнал и на экран.

12.3.2. **Update();**

Обновление текущего сообщения после внесённых скриптом изменений. Вы можете изменить следующие ключи хэша %FMSG:

FromName, FromAddr, ToName, ToAddr, Subject, Body, AreaName, Cost, TimesRead, ReplyTo, NextReply.

Lines, InetFromAddr, InetToAddr не требуют изменения, поскольку их значение автоматически переопределяются при изменении Body, FromAddr и ToAddr.

Внимание! Изменённые данные становятся доступными только для следующих директив Action. По этой причине для изменения файла после ['Action: Script'](#) добавляется ['Action: Rewrite'](#).

Пример:

```
--- test.pl ---
sub maskfunc
{
    $FMSG{"Subject"}="My new subject!";
    Update();
}
--- test.pl ---
```

Mask: * * * * *

Action: Script "maskfunc"

\ После этого файл сообщения остался неизменным

Action: Rewrite * * * * *

\ Теперь в файле сообщения тема изменена на "My new subject!"

12.3.3. ExistsInNodelist("99:99/999");

Проверка 99:99/999 на наличие в нодлисте

12.3.4. FindHub("99:99/999");

Возвращает номер хаба для узла 99:99/999. Возвращает 0 если хаб не найден.

12.3.5. NewMsg(...)

NewMsg(\$MsgBase,\$FromName,\$FromAddr,\$ToName,\$ToAddr,\$Subject,\$Flags,\$Body) — создать сообщение и положить его в базу.

Пример

Следующая команда

```
NewMsg("msgs","Alex","2:5030/1666","User","2:5030/1157.1","msg  
test","", "It is My test");
```

создаст сообщение (типа msg) в директории 'msgs' от Alex@2:5030/1666 к User@2:5030/1157.1 с темой 'msg test' и телом 'It is My test'. Установка любых флагов и кладжей сейчас не поддерживается, так что значения аргументов \$Flags и \$Kludges будут игнорироваться.

Несколько примеров вы можете найти в файле test.pl.

13. Гарантии

Данный продукт является FreeWare, распространяется по принципу "как есть", и никаких гарантий никто не даёт. Разработчики обещают исправлять обнаруженные ошибки и выпускать новые версии.

14. Благодарности

Спасибо Fyodor Ustinov за создание FTrack.

Спасибо людям, которые в разное время переводили данный документ на английский, а именно: Kirill Arushanov и Dennis Sulimov.

Спасибо альфа-тестерам:

Alex Bagmanov 2:5014/4

Boris Gordon 2:5020/701

| | |
|-------------------|-------------|
| Kirill Arushanov | 1:2810/13 |
| Sergey Korowkin | 2:5033/27 |
| Serge Sapozhnikov | 2:4635/34 |
| Matt Bedynek | 1:106/1 |
| Eugene Palenock | 2:5020/2065 |

Спасибо авторам патчей:

| | |
|------------------|-------------|
| Serguei Revtov | 2:5021/19.1 |
| Andrey Slusar | 2:467/126 |
| Michael Kostylev | 2:5030/1530 |

Спасибо Scott Dudley за его Squish MSGAPI. Кстати, Squish это торговая марка SCI Communications.

This product uses the JAM(mbp) API - Copyright 1993 Joaquim Homrighausen, Andrew Milner, Mats Birch, Mats Wallin. ALL RIGHTS RESERVED.

15. Как найти разработчиков

Ваши пожелания и замечания:

- публикуйте в эхо-конференции RU.FTRACK
- публикуйте в баг-трекере на сайте <http://sourceforge.net/projects/ftrack-as>
- отправляйте разработчикам личной почтой.

Разработчики:

- с 2003 по 2006 г. - Alex Soukhotine
2:5030/1157@fidonet (ph0enix@users.sourceforge.net)
- с 2007 г. по настоящее время Stas Degteff (координатор проекта после Алекса)
2:5080/102@fidonet (stas_degteff@users.sourceforge.net)
- с 2009 г. по настоящее время Michael Dukelsky
2:5020/1042@fidonet (dukelsky@users.sourceforge.net)

Сайт проекта:

<http://sourceforge.net/projects/ftrack-as>