Rules Wizard (RWZ) File Format Business Case

Table of Contents

[1. Preamble 2](#_Toc71969048)

[2. Business Case 4](#_Toc71969049)

[2.1. Users want to know the format of RWZ files 4](#_Toc71969050)

[2.2. Users want to import and export rules between applications 4](#_Toc71969051)

[2.3. IT Pros want to automate and script tasks relating to rules 6](#_Toc71969052)

[2.4. Popular Outlook tools want to use the use the RWZ format 7](#_Toc71969053)

[2.5. Users want to parse, edit and view rules without Outlook 8](#_Toc71969054)

[2.6. Forensics companies want to recover and analyse rules 12](#_Toc71969055)

[2.7. Security companies want to detect rules exploits 13](#_Toc71969056)

[3. Documentation Options 14](#_Toc71969057)

[3.1. Office Developer Microsoft Docs 14](#_Toc71969058)

[3.2. Blog Post 14](#_Toc71969059)

[3.3. Test Cases/Dumping Tool 15](#_Toc71969060)

[3.4. Source Code Dump 15](#_Toc71969061)

[3.5. Publish Outlook Symbol Files 16](#_Toc71969062)

[3.6. Work with Open-Source projects on an ad-hoc basis 16](#_Toc71969063)

[3.7. Release an Open Specification 16](#_Toc71969064)

[3.8. Do Nothing 16](#_Toc71969065)

**Revision Summary**

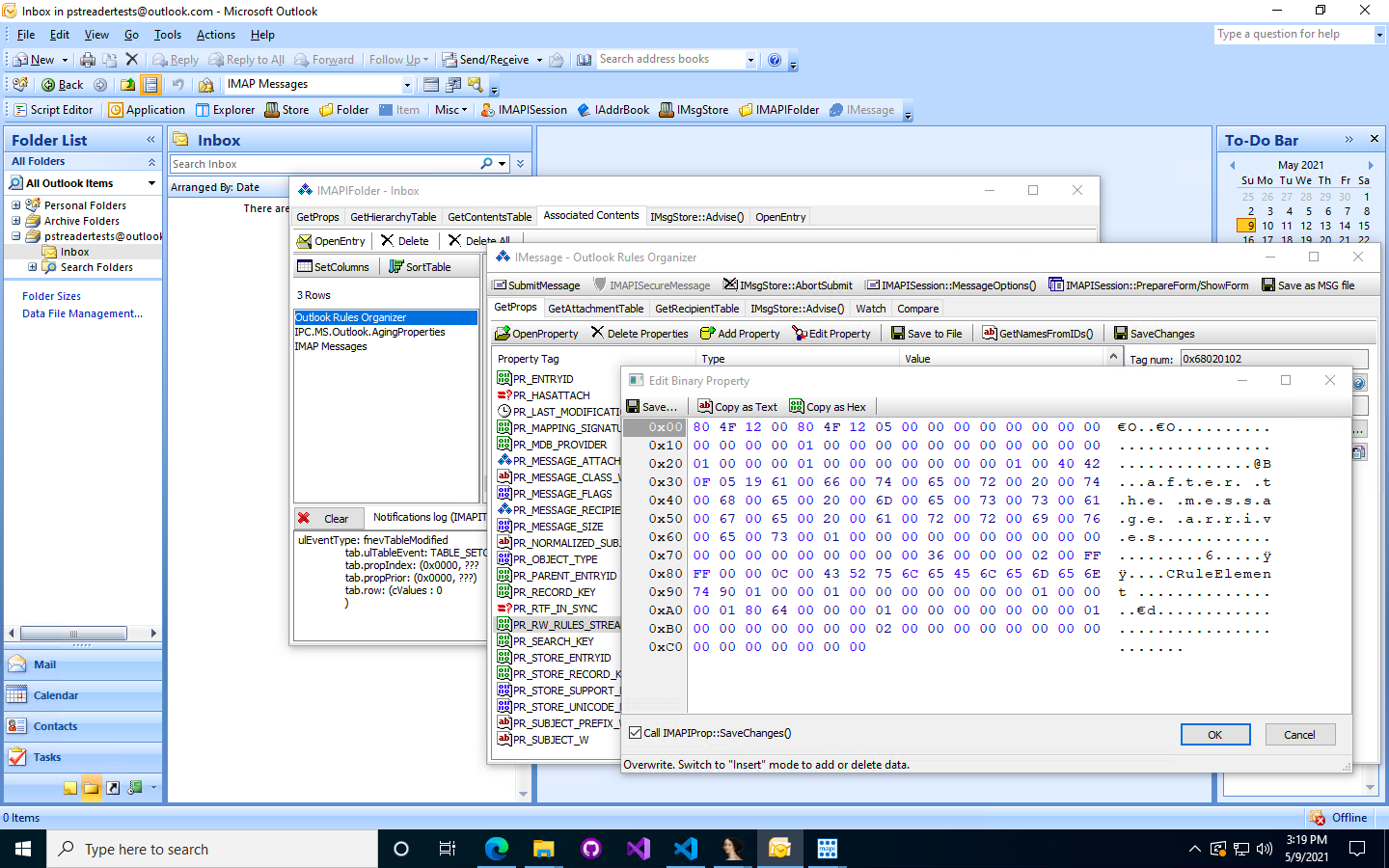
|  |  |
| --- | --- |
| **Date** | **Comments** |
| 15/05/21 | Added section on old security vulnerabilities in the RWZ format  Added symbol files as a documentation open  Fixed typos |
| 09/05/21 | Initial Release |

# Preamble

An Outlook Rule is a set of actions that Outlook performs automatically when certain conditions are met. For example, you can use rules to move, flag, and respond to email messages automatically. You can also use rules to play sounds, move messages to folders, or display new item alerts.[[1]](#footnote-1)

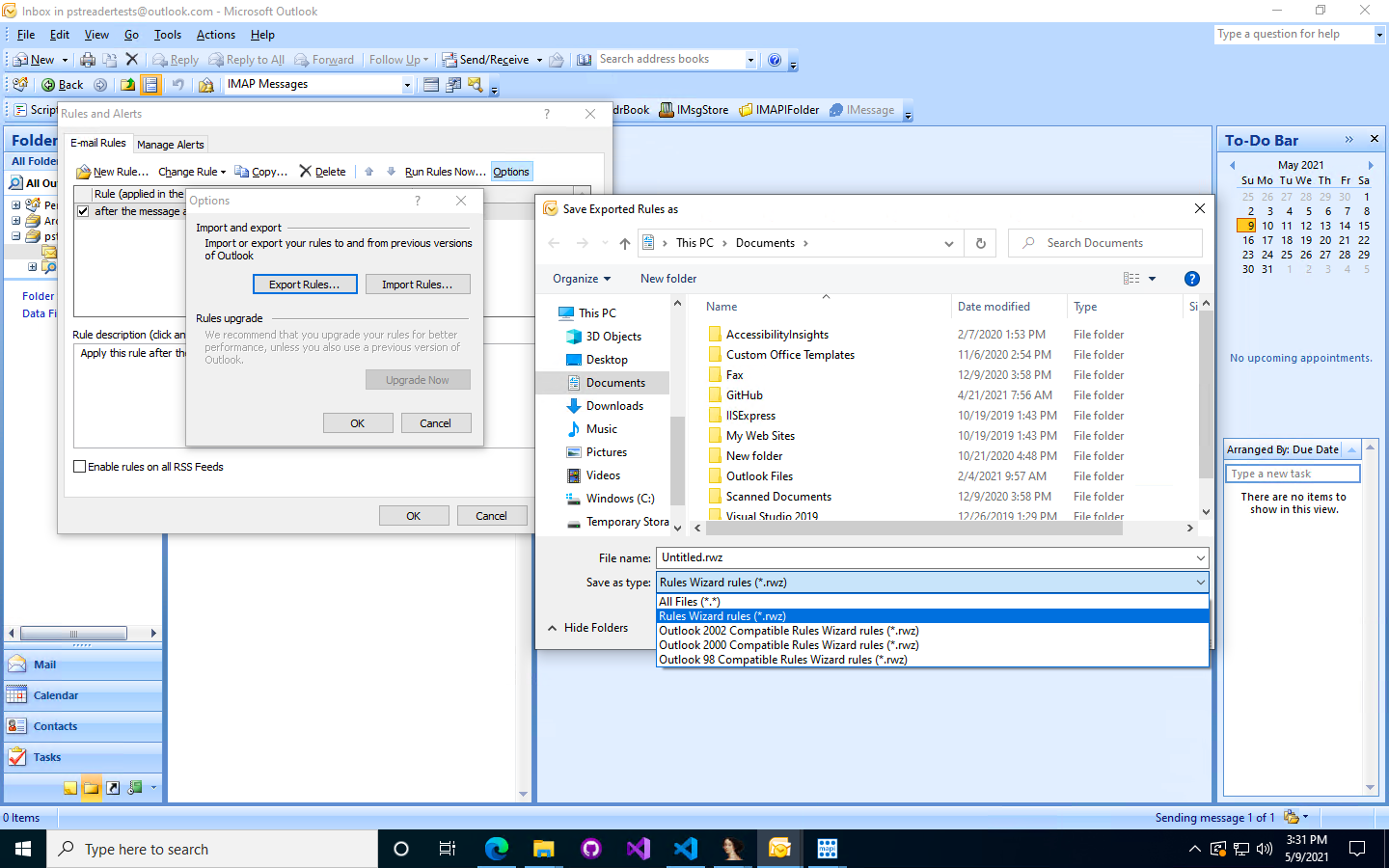
Outlook Rules began as an add-in to Outlook 97[[2]](#footnote-2) called *RulesWiz.dll* and were integrated in-box automatically in Outlook 98[[3]](#footnote-3).

Outlook stores Rules Wizard Files differently depending on the version of Outlook.[[4]](#footnote-4)

* Outlook 2000 and earlier: user-created rules and alerts are stored in their own file ending in .rwz, located in *C:\Documents and Settings\username\Application Data\Microsoft\Outlook*
* Outlook 2002 and later: user-created rules and alerts are stored in the *PR\_RW\_RULES\_STREAM/PidTagRwRulesStream (0x68020102)[[5]](#footnote-5)* property of a message named *Outlook Rules Organizer* of class *IPM.RuleOrganizer* inside the *Associated Contents Table*[[6]](#footnote-6) of the *Inbox* folder

Outlook allows users to import and export Rules to a rules file with the file extension .rwz[[7]](#footnote-7). The resulting file is known as a Rules Wizard File (shorted to RWZ file). Users can specify backwards compatibility to the latest version of Outlook, Outlook 2002, Outlook 2000 and Outlook 98.

The format of *.rwz* files and the corresponding *PR\_RW\_RULES\_STREAM* property is client-specific and opaque to the server[[8]](#footnote-8). Some values within the header are, for example different between the stream contained in PidTagRwRulesStream and the corresponding RWZ exported files. However, this document refers to RWZ files and the binary value of the PidTagRwRulesStream property interchangeably as, by-and-large, they have the same format.



There have been very few attempts to reverse engineer and document the RWZ file format and the documentation remains incomplete

* Kopano[[9]](#footnote-9), an open-source groupware application, contains a draft but incomplete specification of the file format[[10]](#footnote-10), but does not include an implementation
* OutlookRulesReader[[11]](#footnote-11), an open-source Swift library, contains a draft but incomplete specification of the file format[[12]](#footnote-12), and includes an implementation/API for reading and writing RWZ files

This document describes a business case for documenting the opaque RWZ file format. This document also describes the means by which the file format can be documented. Sources for this business case include online research and discussions/interviews with key stakeholders, including pro-users, MAPI experts and MVPs such as Dmitry Streblechenko (founder of Outlook Spy[[13]](#footnote-13)) and Stephen Griffin (founder of MFCMAPI[[14]](#footnote-14)) and forensics companies such as legal e-discovery firm GoldFynch[[15]](#footnote-15).

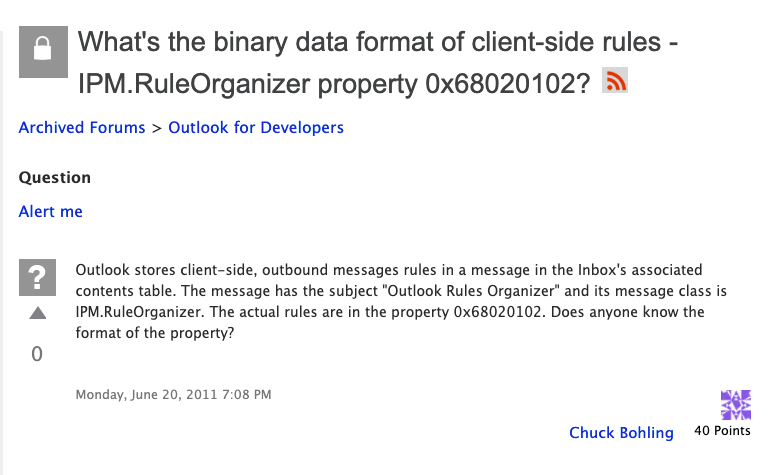
For any questions, email [hughbellars@gmail.com](mailto:hughbellars@gmail.com).

# Business Case

## Users want to know the format of RWZ files

Before detailing more real-world applications of the file format, there is clear interest from online discussions and conversations that users want to decipher this file format.

E.g. <https://social.msdn.microsoft.com/Forums/office/en-US/d1cbdc2e-b63d-4fe6-b062-d6ceb16b2abd/whats-the-binary-data-format-of-clientside-rules-ipmruleorganizer-property-0x68020102>



E.g. <https://stackoverflow.com/questions/127336/decipher-client-side-rules-binary-definition-for-microsoft-outlook>

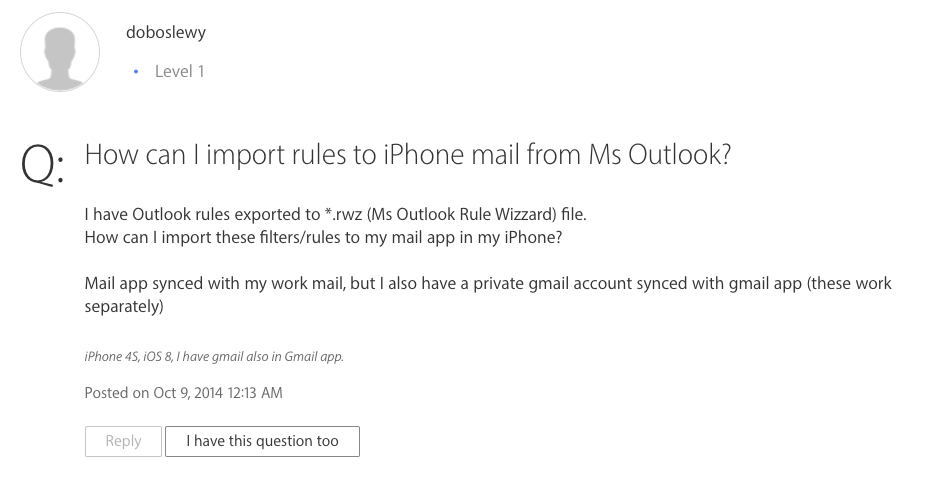


## Users want to import and export rules between applications

Documenting the RWZ file format would allow interoperability between mail applications. For example, if enabled, users could import rules from Outlook into email clients such as Gmail, iPhone mail, etc.

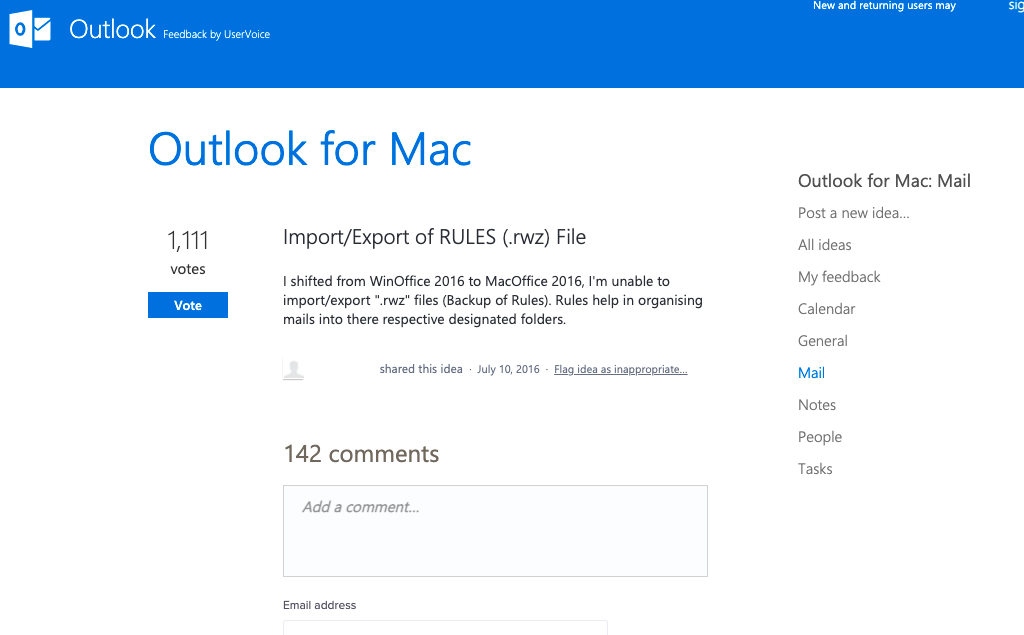
This is one of the reasons why open-source groupware Kopano has begun attempts to document the RWZ file format.

E.g. <https://discussions.apple.com/thread/6588002>

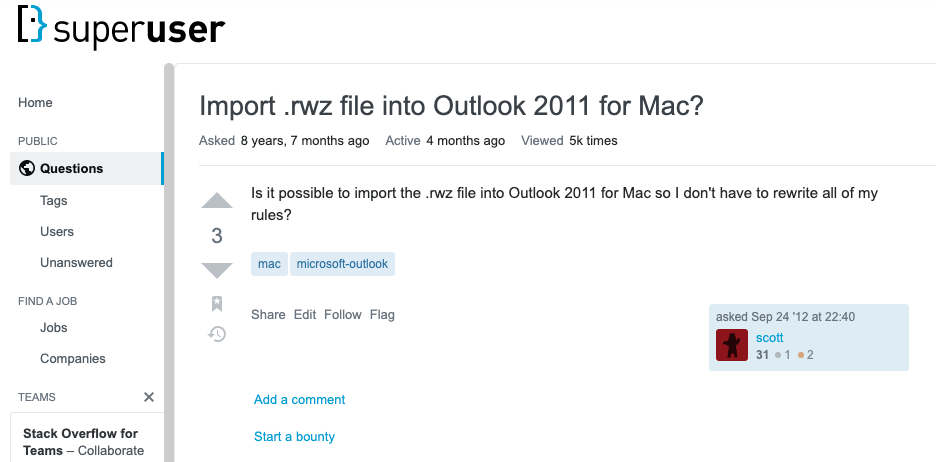


Benefits are not just limited to interoperability between Outlook and other non-Microsoft mail clients, but also between Outlook for Mac and Windows. For example, it may be possible to create an add-in for Outlook for Mac to allow importing RWZ files into Outlook for Mac and exporting from Outlook for Mac to Outlook for Windows.

E.g. <https://outlook.uservoice.com/forums/293343-outlook-for-mac/suggestions/15139845-import-export-of-rules-rwz-file>



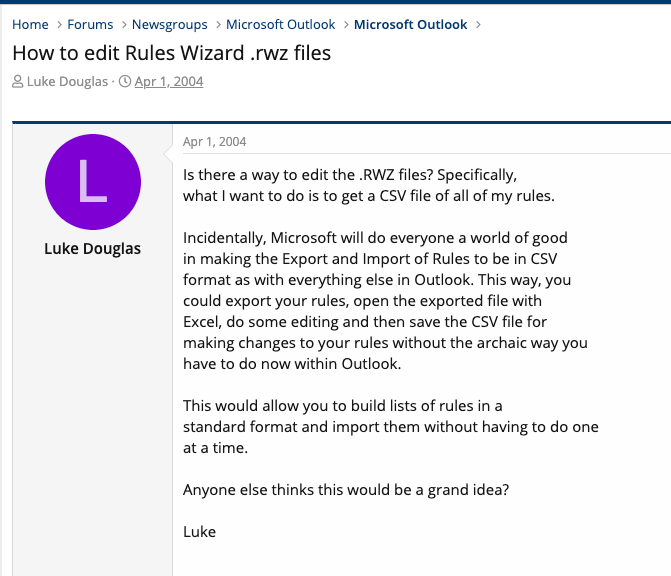
E.g. <https://superuser.com/questions/479154/import-rwz-file-into-outlook-2011-for-mac>



E.g. <https://social.technet.microsoft.com/Forums/office/en-US/2554d185-e015-4ff1-a07d-1140bbccf200/is-there-any-way-to-openedit-a-rwz-file-or-any-utility-to-import-inbox-rules-to-outlook-2013-from?forum=officeitpro>



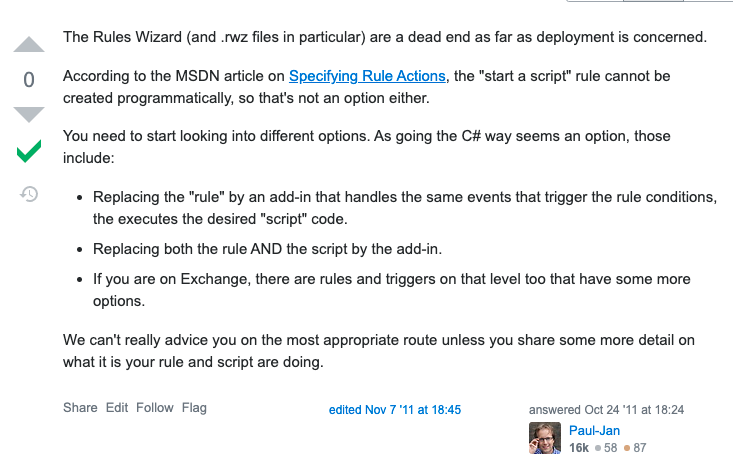
E.g. <https://www.pcreview.co.uk/threads/how-to-edit-rules-wizard-rwz-files.775604/>



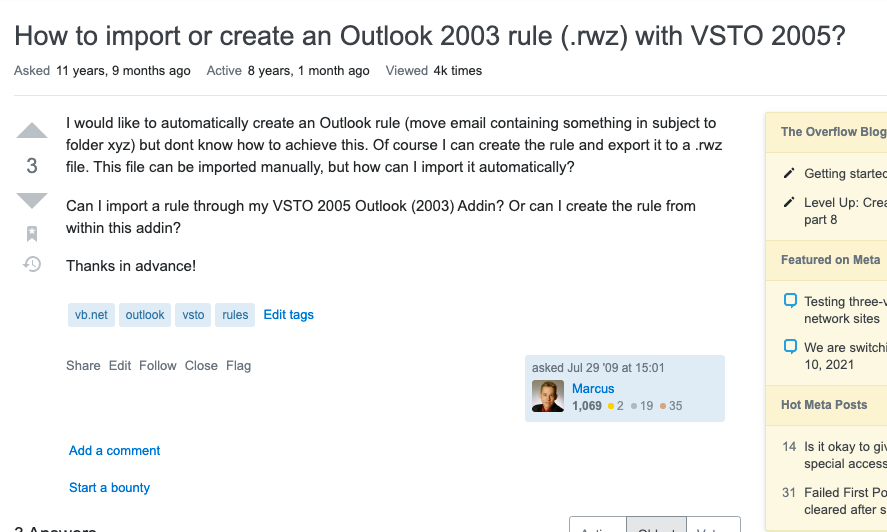
## IT Pros want to automate and script tasks relating to rules

Documenting the RWZ file format would allow Systems Managers and IT Pros to automate tasks relating to the creation and managing of Outlook Rules. For example, pros want to create templates that can be automatically imported.

E.g. <https://stackoverflow.com/questions/7839204/deploy-programmatically-create-outlook-rule-to-run-script>



E.g. <https://stackoverflow.com/questions/1200919/how-to-import-or-create-an-outlook-2003-rule-rwz-with-vsto-2005>

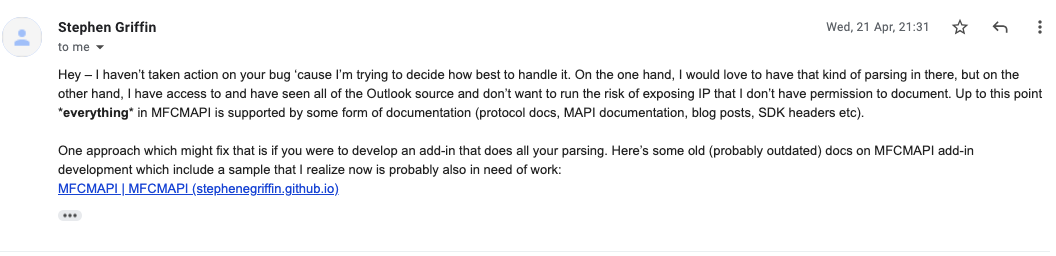


## Popular Outlook tools want to use the use the RWZ format

Outlook tools such as Redemption[[16]](#footnote-16), Outlook Spy[[17]](#footnote-17) and MFCMAPI[[18]](#footnote-18) do not have the capability to perform actions with the RWZ format.

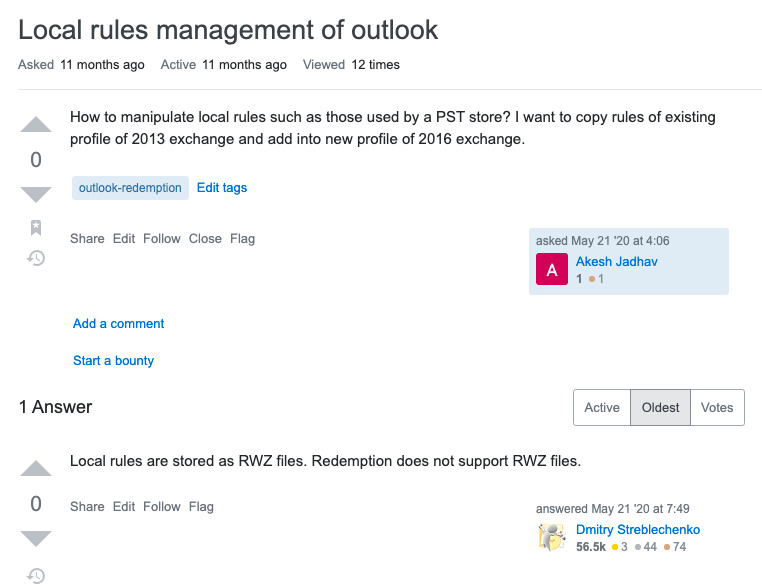
For example, there is demand for MFCMAPI to add parsing of PidTagRwRulesStream but this is not possible until the format is documented.

E.g. <https://github.com/stephenegriffin/mfcmapi/issues/465>



For example, users cannot manage their Outlook Rules using Redemption (although this feature may be supported in Redemption in the future, given). In addition to the following example, conversations with Dmitry Streblechenko (MVP, founder of Outlook Spy and Redemption), revealed that this was a common request of Redemption.

E.g. <https://stackoverflow.com/questions/61927073/local-rules-management-of-outlook>



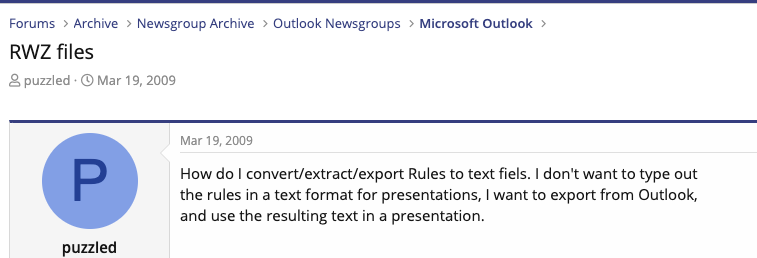
## Users want to parse, edit and view rules without Outlook

Users want to be able to view Outlook rules – for example to a text format or csv format for displaying outside of Outlook (e.g. text files, excel, presentations or printed). Documenting the format will allow the development of third-party tools that can enable users to do this

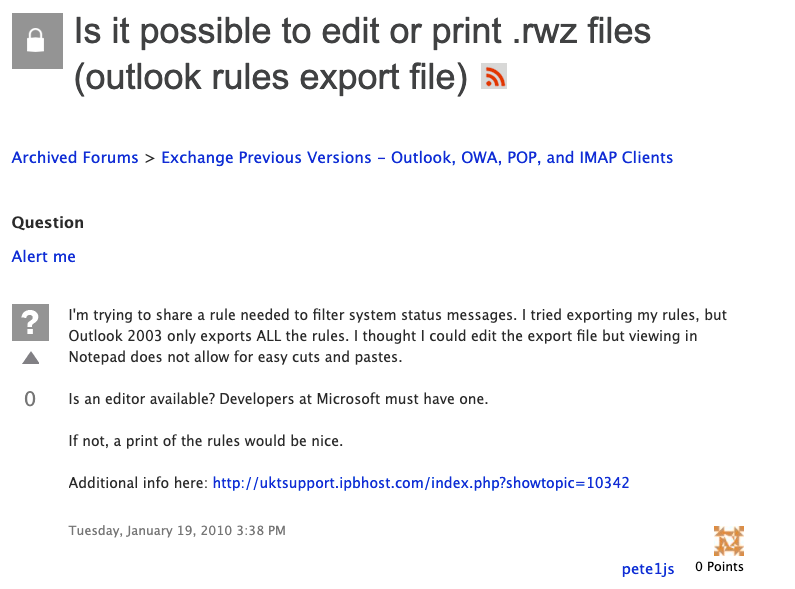
E.g. <https://superuser.com/questions/92197/export-outlook-2003-rules-to-text>



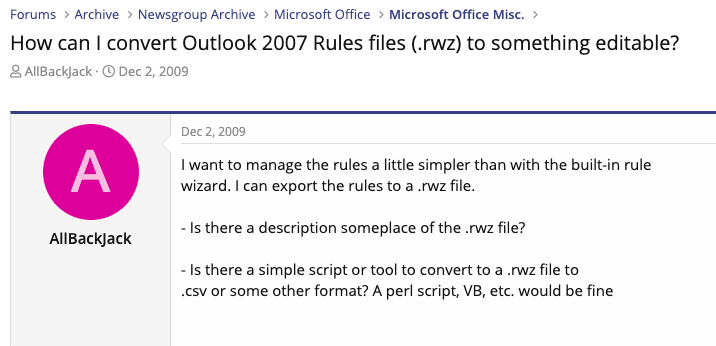
E.g. <https://www.office-forums.com/threads/rwz-files.1483435/>



E.g. <https://social.technet.microsoft.com/Forums/exchange/en-US/a78d158a-9278-45ee-aeff-a7f18fee41f6/is-it-possible-to-edit-or-print-rwz-files-outlook-rules-export-file?forum=exchangesvrclientslegacy>



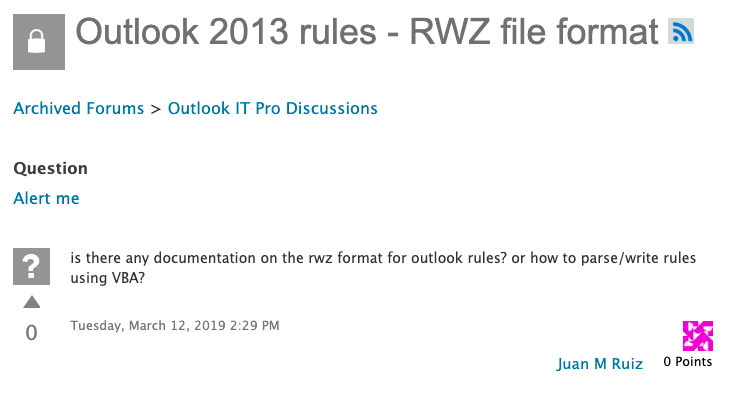
E.g. <https://www.office-forums.com/threads/how-can-i-convert-outlook-2007-rules-files-rwz-to-something-editable.2239888/> & <https://groups.google.com/g/microsoft.public.office.misc/c/D-_o46u_Hcs>



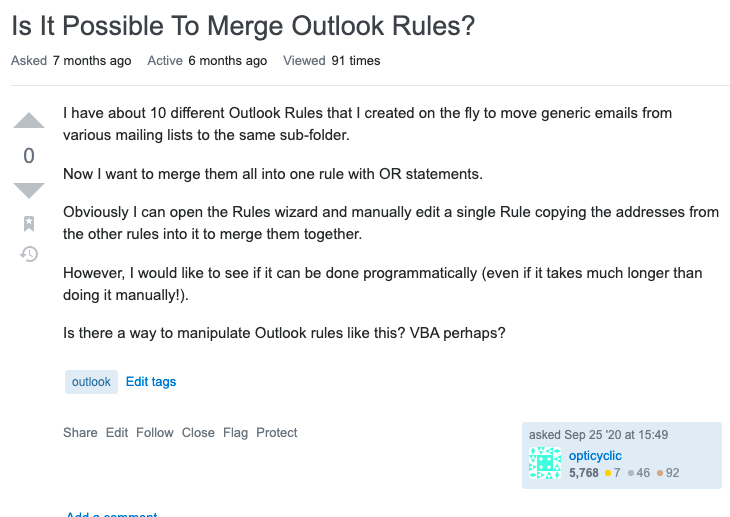
Users also want to overcome limitations of the Outlook rules editor. For example, they want to merge rules, sort rules, import & export only selected rules, etc. Documenting the file format would allow third parties to offer such functionality.

Note: although the Outlook object model enabled some functionality, there are some problems with this. Firstly, it requires Outlook to be installed (which excludes servers/Linux/installations without Windows) and the object model is not comprehensive (e.g. it does not allow for custom actions)

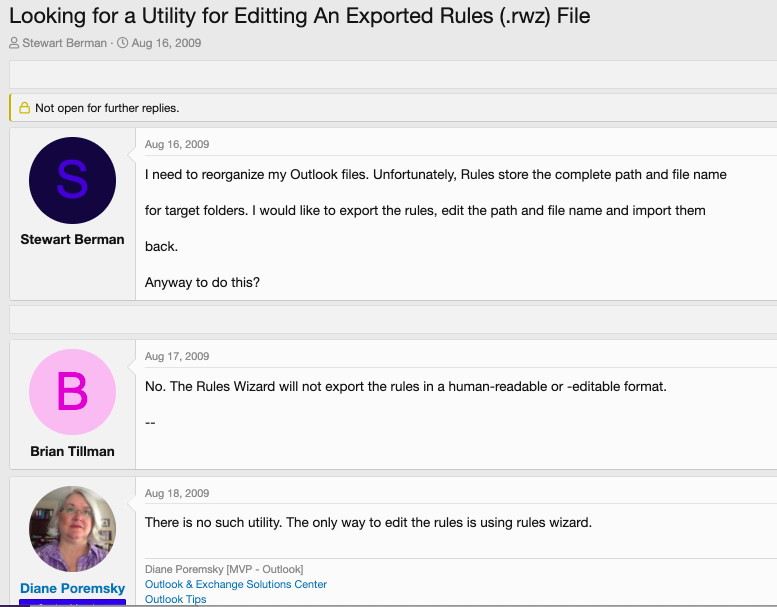
E.g. <https://social.technet.microsoft.com/Forums/en-US/0cb9d0d6-9c7d-4d47-852b-2a0b63438975/outlook-2013-rules-rwz-file-format?forum=outlook>



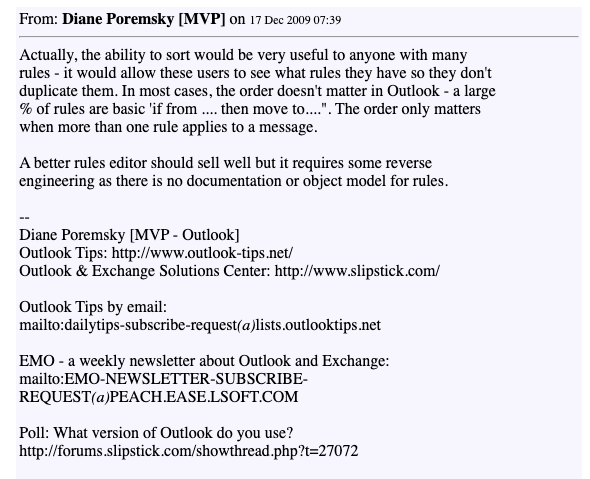
E.g. <https://stackoverflow.com/questions/64067406/is-it-possible-to-merge-outlook-rules>



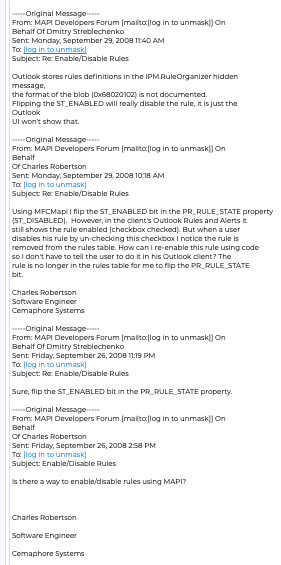
E.g. <https://forums.slipstick.com/threads/18711-looking-for-a-utility-for-editting-an-exported-rules-rwz-file/>



E.g. <http://www.adras.com/rwz-viewer-for-outlook-rules.t34546-14.html>



E.g. <https://peach.ease.lsoft.com/scripts/wa-PEACH.exe?A2=MAPI-L;e6463a34.0809&S=>



## Forensics companies want to recover and analyse rules

Interviews with stakeholders revealed that forensics companies want to be able to recover, read and analyse rules. For example, e-discovery firm GoldFynch[[19]](#footnote-19) has developed a solution for legal clients that allow lawyers to analyse PST dumps. Documenting the RWZ and PidTagRwRulesStream format will allow forensics companies the ability to get a better picture of their legal problems in question[[20]](#footnote-20), [[21]](#footnote-21).

For example, rules may explain why certain emails were not actioned or received (e.g. there was a rule to mark them as read, move them or even permanently delete them). Rules can reveal deliberate or accidental non-compliant or illegal behaviour such as forwarding sensitive or restricted information to personal accounts or to unauthorized individuals[[22]](#footnote-22). Rules can also help show “the big picture” within a company by providing a glimpse of relationships between employees.

## Security companies want to detect rules exploits

Business email compromise (BEC) attacks cost companies $3.5bn globally annually, according to 2019 FBI analysis[[23]](#footnote-23). This number will only increase as a result of the COVID-19 pandemic, increased online usage over time and the development of more sophisticated attacks that are more complex and harder to detect.

Outlook Rules exploits are well recognised. Rules and Custom Forms Injection attacks exist, whereby malicious attackers can setup forwarding rules and trigger custom actions[[24]](#footnote-24). Hidden rules attacks have been revealed in the past[[25]](#footnote-25), [[26]](#footnote-26). Malicious Outlook rules have been used to get shell access[[27]](#footnote-27), [[28]](#footnote-28), [[29]](#footnote-29), [[30]](#footnote-30).

Outlook Rules also form a key part of exploit chains involving Outlook & Exchange. According to Cypsis, a breach response and risk management cybersecurity firm, “Once threat actors have access to a mailbox, we commonly observe them leveraging the mailbox rule capabilities of either Inbox, Simple Mail Transfer Protocol (SMTP) forwarding, and Transport policy to forward emails to an external email address or to hide or delete emails from the original intended recipients.”[[31]](#footnote-31) Tools also exist, such as XRulz[[32]](#footnote-32). XRulz’s developers state that “Outlook rules can be used to achieve persistence on Windows hosts by creating a rule that executes a malicious payload. The rule can be setup to execute when the target receives an email with a specific keyword in the subject. An attacker can then drop shells on a target as and when they require by simply sending an email. In the past, this technique could only be done via the Outlook GUI. However, XRulez achieves this from the command line.”[[33]](#footnote-33)

Documenting the file format will allow cybersecurity and security penetration companies to develop general software for detecting and fixing common exploitation patterns and develop bespoke software for clients to ensure that certain policies are followed (e.g. there are no rules for printing or forwarding sensitive information etc.).

In 2018, researchers at FortiGuard Labs discovered and responsibly disclosed four remote code execution vulnerabilities[[34]](#footnote-34) (CVE-2018-8522[[35]](#footnote-35), CVE-2018-8524[[36]](#footnote-36), CVE-2018-8576[[37]](#footnote-37) and CVE-2018-8582) and a heap corruption vulnerability[[38]](#footnote-38) (CVE-2018-8587[[39]](#footnote-39)) in Outlook relating to importing malformed RWZ files. These are exploitable vulnerabilities that allowed malicious actors to execute code remotely in the context of the current user[[40]](#footnote-40).

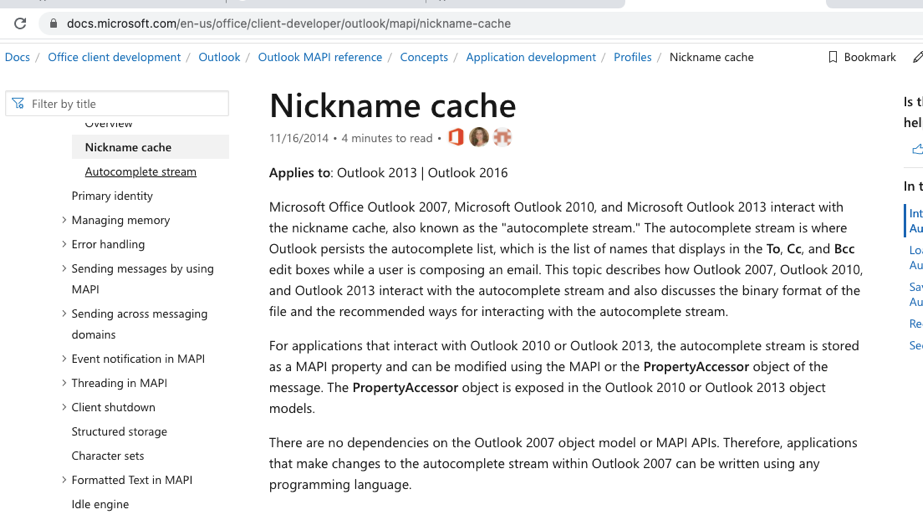
Security researchers may be able to use the documentation of the RWZ format to write fuzzers against Outlook, as Outlook Rules represent a surface area for attack. Transparency benefits the community – documenting the format may avoid future (or as yet unknown) attacks on the RWZ parser/surface area. The Rules Wizard code is old, is over 20 years old and appears to have been developed originally by a contracting software company, and not Microsoft. Code like this has caused problems in Office in the past, for example in the 17-year-old Equation Editor[[41]](#footnote-41).

# Documentation Options

There exist multiple options for documentation. The following list is not prescriptive and alternative options can be explored. This document does not suggest any one idea but does give examples of some options. It is acknowledged that there may exist some or significant developer time associated with any documentation option and some may not be feasible or in fact desirable. Any action taken should not disproportionately exceed the benefits of documenting the format listed in this document.

## Office Developer Microsoft Docs

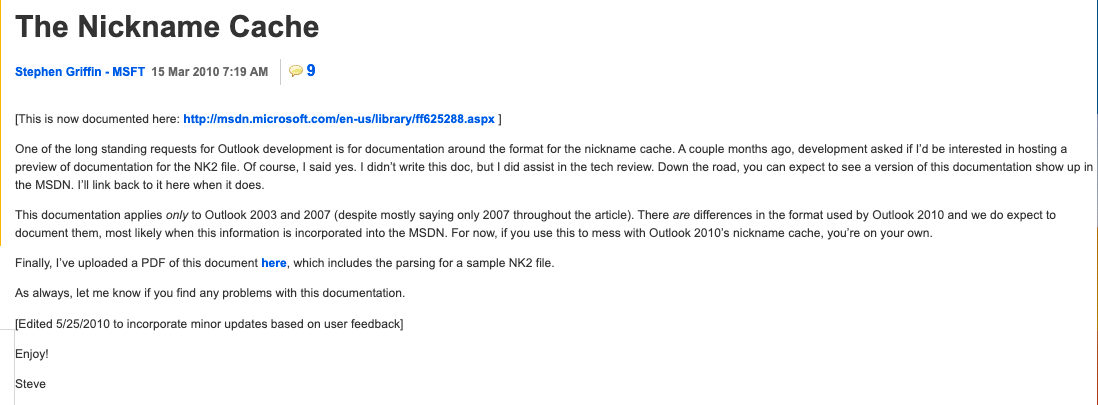
The format could be documented in the Office Developer Microsoft Docs. This has been used for the documentation of the Outlook Nickname Cache (NK2) file format[[42]](#footnote-42). The documentation should detail the format of the file header and footer, each rule’s header and each element contained within a rule.



This may require some time to document and there may not be sufficient demand within the product group.

## Blog Post

The format could be documented in a blog post. For example, MFCMAPI developer Stephen Griffin, posted a blog post describing the Outlook Nickname Cache (NK2) file format[[43]](#footnote-43). It is notable that this blog post led to increased demand for documentation, and accordingly the format was subsequently officially documented in the Office Developer Microsoft Docs. This path may be a good stopgap measure to fix ongoing incomplete implementations in current open-source projects.



This may require some time to document and there may not be sufficient demand within the product group.

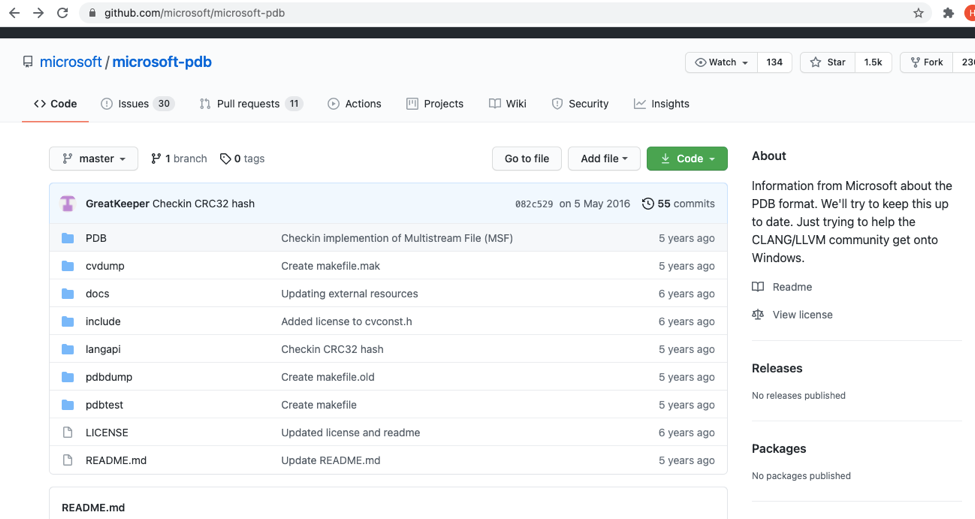
## Test Cases/Dumping Tool

The format could be documented through the release of some test cases and perhaps a tool that releases corresponding textual dump. For example, Microsoft has released tools for dumping PDB and PST files[[44]](#footnote-44).

There may exist no test cases or dumping tool for RWZ files.

## Source Code Dump

In the past, Microsoft have released source code dumps to assist parsing file formats that are requested by the community. For example, the microsoft-pdb project was released help the CLANG/LLVM community get onto Windows[[45]](#footnote-45). The following comment describes the aim of releasing the source code: “With this information we are now building the information for other compilers (and tools) to efficiently write a PDB. The PDB format has not been officially documented, presenting a challenge for other compilers and toolsets (such as Clang/LLVM) that want to work with Windows or the Visual Studio debugger. We want to help the Open-Source compilers to get onto the Windows platform.”



This may conflict with some source licensing restrictions. Furthermore, due to the Rule Wizard’s tight integration, release of source code may be impossible.

## Publish Outlook Symbol Files

Microsoft could release symbol (PDB) files for Office libraries and executables, for example outlook.exe and olmapi32.dll. This would allow reverse engineers to be able to understand the file format better as symbols can explain what various unknown functions do and give insights into the file format parser.

This may require approval within the product group as Office symbols are not currently released. Further, PDB files may need to be cleaned up to remove language inappropriate for public consumption and to exclude code from third-party vendors[[46]](#footnote-46).

## Work with Open-Source projects on an ad-hoc basis

As discussed previously, there exists some documentation of the RWZ file format by the Kopano developers[[47]](#footnote-47) and a more comprehensive specification and implementation in OutlookRulesReader[[48]](#footnote-48). Outlook developers could help certain open-source projects in answering specific questions, for example in revealing the meaning of unknown fields. Given their access to the source code, they may be able to answer some questions easily.

This may risk stratifying the community by creating a “preferred” project for reading/writing Outlook Rules. Furthermore, it is not in the spirit of open-source and does not convey the benefits to the open-source community as it would not be usable by tools such as MFCMAPI who only parse documented formats in their source base.

## Release an Open Specification

Microsoft could release an open specification for the RWZ file format, similar to the MSG[[49]](#footnote-49), PST (Personal Storage Table)[[50]](#footnote-50) and OAB (Offline Address Book)[[51]](#footnote-51) file format.

This may require some time to document and there may not be sufficient demand within the product group.

## Do Nothing

The OutlookRulesReader project already contains a ~80-90% reverse engineering effort of the RWZ file format. Considering the progress already made and the effort required by the Microsoft product group to document the format exceeds the benefits.

This may risk incomplete or incorrect documentation becoming the de-facto source of truth. Furthermore, it is not in the spirit of open-source and does not convey the benefits to the open-source community as it would not be usable by tools such as MFCMAPI who only parse documented formats in their source base.

1. <https://support.microsoft.com/en-us/office/manage-email-messages-by-using-rules-c24f5dea-9465-4df4-ad17-a50704d66c59> [↑](#footnote-ref-1)
2. <https://www.computerhope.com/download/updates.htm> [↑](#footnote-ref-2)
3. <https://www.slipstick.com/outlook/rules/rules-wizard/> [↑](#footnote-ref-3)
4. <https://www.itprotoday.com/email-and-calendaring/how-do-you-delete-and-recreate-corrupt-outlook-rules> [↑](#footnote-ref-4)
5. <https://docs.microsoft.com/en-us/openspecs/exchange_server_protocols/ms-oxprops/f1fbd19a-9b2c-4c16-a07b-2242b2c0dece> [↑](#footnote-ref-5)
6. <https://docs.microsoft.com/en-us/office/client-developer/outlook/mapi/contents-tables> [↑](#footnote-ref-6)
7. <https://support.microsoft.com/en-us/office/import-or-export-a-set-of-rules-f54b5bd2-40e0-426e-9f25-e51fa14eeb95> [↑](#footnote-ref-7)
8. <https://docs.microsoft.com/en-us/openspecs/exchange_server_protocols/ms-oxorule/ee0fd52b-1161-49b8-adab-6b8c2e31c3f3> [↑](#footnote-ref-8)
9. <https://github.com/Kopano-dev/kopano-core> [↑](#footnote-ref-9)
10. <https://github.com/Kopano-dev/kopano-core/blob/master/doc/ol_rule_spec.txt> [↑](#footnote-ref-10)
11. <https://github.com/hughbe/OutlookRulesReader> [↑](#footnote-ref-11)
12. <https://github.com/hughbe/OutlookRulesReader/blob/master/docs/RWZ%20Format.pdf> [↑](#footnote-ref-12)
13. <https://www.dimastr.com/outspy/home.htm> [↑](#footnote-ref-13)
14. <https://github.com/stephenegriffin/mfcmapi> [↑](#footnote-ref-14)
15. <https://goldfynch.com/> [↑](#footnote-ref-15)
16. <https://www.dimastr.com/redemption/home.htm> [↑](#footnote-ref-16)
17. <https://www.dimastr.com/outspy/> [↑](#footnote-ref-17)
18. <https://github.com/stephenegriffin/mfcmapi> [↑](#footnote-ref-18)
19. <https://goldfynch.com/> [↑](#footnote-ref-19)
20. <https://www.pstwalker.com/email-investigations-and-forensic-analysis-of-outlook-exchange-email-files.html> [↑](#footnote-ref-20)
21. <https://www.forensicfocus.com/forums/general/outlook-rule-settings-file/> [↑](#footnote-ref-21)
22. <https://www.crypsisgroup.com/insights/microsoft-365-inbox-and-forwarding-rule-forensics> [↑](#footnote-ref-22)
23. <https://www.fbi.gov/scams-and-safety/common-scams-and-crimes/business-email-compromise> [↑](#footnote-ref-23)
24. <https://docs.microsoft.com/en-us/microsoft-365/security/office-365-security/detect-and-remediate-outlook-rules-forms-attack?view=o365-worldwide> [↑](#footnote-ref-24)
25. <https://blog.compass-security.com/2018/09/hidden-inbox-rules-in-microsoft-exchange/> [↑](#footnote-ref-25)
26. <https://www.crypsisgroup.com/insights/discovering-hidden-rules-business-email-crypsis> [↑](#footnote-ref-26)
27. <https://www.blackhillsinfosec.com/malicious-outlook-rules-action/> [↑](#footnote-ref-27)
28. <http://www.cunaceocouncil.net/cuna/assets/files/155211_Email-Phishing.pdf> [↑](#footnote-ref-28)
29. <https://www.n00py.io/2017/03/from-osint-to-internal-gaining-access-from-the-outside-the-perimeter/> [↑](#footnote-ref-29)
30. <https://gist.github.com/monoxgas/7fec9ec0f3ab405773fc> [↑](#footnote-ref-30)
31. <https://www.crypsisgroup.com/insights/microsoft-365-inbox-and-forwarding-rule-forensics> [↑](#footnote-ref-31)
32. <https://labs.f-secure.com/archive/malicous-outlook-rules/> [↑](#footnote-ref-32)
33. <https://github.com/FSecureLABS/XRulez> [↑](#footnote-ref-33)
34. <https://www.fortinet.com/blog/threat-research/patch-your-microsoft-outlook--fortinet-discovered-four-outlook-r> [↑](#footnote-ref-34)
35. <https://msrc.microsoft.com/update-guide/en-us/vulnerability/CVE-2018-8522> [↑](#footnote-ref-35)
36. <https://msrc.microsoft.com/update-guide/en-us/vulnerability/CVE-2018-8524> [↑](#footnote-ref-36)
37. <https://msrc.microsoft.com/update-guide/en-us/vulnerability/CVE-2018-8576> [↑](#footnote-ref-37)
38. <https://www.fortinet.com/blog/threat-research/a-deep-analysis-of-the-microsoft-outlook-vulnerability-> [↑](#footnote-ref-38)
39. <https://msrc.microsoft.com/update-guide/vulnerability/CVE-2018-8587> [↑](#footnote-ref-39)
40. <https://security.stackexchange.com/questions/213840/microsoft-outlook-vulnerability-cve-2018-8587-how-likely-is-exploitation> [↑](#footnote-ref-40)
41. <https://securityboulevard.com/2018/01/microsoft-kills-old-office-equation-editor-due-to-new-flaw/> [↑](#footnote-ref-41)
42. <https://docs.microsoft.com/en-us/office/client-developer/outlook/mapi/nickname-cache> [↑](#footnote-ref-42)
43. <http://web.archive.org/web/20120325162100/http://blogs.msdn.com/b/stephen_griffin/archive/2010/03/15/the-nickname-cache.aspx> [↑](#footnote-ref-43)
44. <https://github.com/microsoft/microsoft-pdb/blob/master/pdbdump/pdbdump.cpp> [↑](#footnote-ref-44)
45. <https://github.com/microsoft/microsoft-pdb> [↑](#footnote-ref-45)
46. <https://docs.microsoft.com/en-us/archive/blogs/mpower/outlook-symbols> [↑](#footnote-ref-46)
47. <https://github.com/Kopano-dev/kopano-core/blob/e23230568e679c5879bf0a027754854576be5060/doc/ol_rule_spec.txt> [↑](#footnote-ref-47)
48. <https://github.com/hughbe/OutlookRulesReader> [↑](#footnote-ref-48)
49. <https://docs.microsoft.com/en-us/openspecs/exchange_server_protocols/ms-oxmsg/b046868c-9fbf-41ae-9ffb-8de2bd4eec82> [↑](#footnote-ref-49)
50. <https://docs.microsoft.com/en-us/openspecs/office_file_formats/ms-pst/141923d5-15ab-4ef1-a524-6dce75aae546> [↑](#footnote-ref-50)
51. <https://docs.microsoft.com/en-us/openspecs/exchange_server_protocols/ms-oxoab/b4750386-66ec-4e69-abb6-208dd131c7de> [↑](#footnote-ref-51)