Click Here

 2010 Jul 30;11:405. doi: 10.1186/1471-2105-11-405 Today, data evaluation has become a bottleneck in chromatographic science. Analytical instruments equipped with automated samplers yield large amounts of measurement data, which needs to be verified and analyzed. Since nearly every GC/MS instrument vendor offers its own data format and software tools, the consequences are problems with data exchange and a lack of comparability between the analytical results. To challenge this situation a number of either commercial or non-profit software applications have been developed. These applications provide functionalities to import and analyze several data formats but have shortcomings in terms of the transparency of the implemented analytical algorithms and/or are restricted to a specific computer platform. This work describes a native approach to handle chromatographic data files. The approach can be extended in its functionality such as facilities to detect baselines, to detect, integrate and identify peaks and to compare mass spectra, as well as the ability to internationalize the application. Additionally, filters can be applied on the chromatographic data to enhance its quality, for example to remove background and noise. Extended operations like do, undo and redo are supported. OpenChrom is a software application to edit and analyze mass spectrometric chromatographic data. It is extensible in many different ways, depending on the demands of the users or the analytical procedures and algorithms. It offers a customizable graphical user interface. The software is independent of the operating system, due to the fact that the Rich Client Platform is written in Java. OpenChrom is released under the Eclipse Public License 1.0 (EPL). There are no license constraints regarding extensions. They can be published using open source as well as proprietary licenses. OpenChrom is available free of charge at . Software has become an integral part of analysis techniques. Especially in the area of gas chromatography/mass spectrometry, automatic samplers enable high throughput analyses. Software assists handling large amounts of data generated by automated and fast operating analytical instruments. Modern computer systems are inexpensive, powerful and allow analysis techniques that could not have been applied in the past. Deconvolution, a chromatographic quality enhancing technique, demonstrates for instance that increasing processor power makes new analysis techniques applicable. The technique of deconvolution has been described by Biller and Biemann [1,2], Dromey et al. [3], Colby [4], Hindmarch et al. [5], Halket et al. [6], Kong et al. [7], Taylor et al. [8], Pool et al. [9,10] and Davies [11] in various ways. Stein [12] published an enhanced deconvolution algorithm that has been implemented in the software AMDIS (Automated Mass Spectral Deconvolution and Identification System) [13]. AMDIS is available free of charge from the National Institute of Standards and Technology (NIST). Windig et al. [14,15] described another approach to enhance chromatographic quality by a deconvolution method called CODA (Component Detection Algorithm). The commercially available software ACD/MS Manager [16] offers an implementation of this approach. Increasing computational power enables new applications, but there is still a lack of interoperability. Instrument vendors, such as Agilent Technologies, Shimadzu, Thermo Fisher Scientific and Waters Corporation have created their own software and data format. Usually, the mass spectral data formats are binary and can only be accessed by the instrument vendors' proprietary software. Some commercial tools exist to convert the mass spectral data files into other formats, such as MASS Transit from PALISADE Corporation [17]. To avoid these limitations, some efforts have been made to design and implement interoperable data formats and software libraries as for example NetCDF [18] or mzXML [19,20]. But even if it is possible to convert the data files to other formats, there are drawbacks in data processing as each software implements specific functions, has its own graphical user interface and is in most cases commercially available only, as for example the applicable software of ChemStation, Xcalibur or MassLynx. Hence, the users are forced to become familiar with different software systems, user interfaces and methods. Moreover, the software tools primarily target only specific operating systems, such as Microsoft Windows and Mac OSX. The number of software applications that are independent of the operating system and can also be run under Unix or Linux is limited. Linux systems are open source, available at no cost and their usage increases in scientific research (see Scientific Linux [21]), as well as in the public sector [22,23]. Software applications, such as AMDIS, have been published to be used free of charge, but their source code is not disposable. Thus, it is not possible to evaluate the algorithms implemented in the software. Especially in the case of scientific research, it is not possible to figure them out and to extend them. Even if algorithms are described in published papers [2,4,9,12,24], it is often impossible to validate them manually due to the complexity of chromatographic data. Other applications like ChemStation, Xcalibur, ACD/MS Manager are proprietary and closed source. They are only commercially available. There is no means of revealing the correctness of their utilized algorithms. Efforts have been made to solve the problems of missing interoperability and restricted access to source codes and algorithms [25]. Bioclipse is a sophisticated project that is open source and is focused with its algorithms on metabolism analysis and gene sequencing. Its techniques are state-of-the-art. Some other projects are mMass [26], COMSPARI [27] and fityk [28], but they do have some restrictions regarding their interoperability and extensibility. BioSunMS [29] is a tool to read TOF (Time of Flight) mass spectral data files, but it is not able to read instrument vendors' native data files. The Chemistry Development Kit (CDK) [30] implements convenient features to edit chemical data and structures, but it has no appropriate user interface. The open source tool OpenMS [31] aims to edit mass spectrometric data, but it is not completely platform independent, as it is written in C++ programming language. Projects like Bioclipse, Sashimi [32] or TPP (Trans-Proteomic Pipeline) [33] are focused on the evaluation of metabolism products and gene sequencing and make extensive use of accurate mass resolution techniques. But there is still a lack of software systems that are capable to enhance nominal mass spectral data files, that are flexible, extensible and that offer an easy to use graphical user interface. According to the authors' knowledge, no application offers functions to import vendor systems chromatographic data files and has the ability to edit and analyze chromatograms in the way ChemStation and AMDIS do. No application combines the flexibility in analyses, is easily extensible, open source, platform independent and has a configurable graphical user interface. OpenChrom is an open source software that aims to solve the aforementioned constraints getting rid of several restrictions. It is based on the Eclipse Rich Client Platform (RCP) [34], which is an OSGi (Open Service Gateway Initiative) based application environment that allows to build modular and flexible software systems. With the OSGi platform it is possible to extend the functionality of an application by dividing its components into different bundles. It is written in Java which is an interpreted language that depends on the Java Virtual Machine (JVM) and allows the execution of the software on several operating systems (Microsoft Windows, Mac OSX, Unix, Linux) and processor platforms (x86, PPC, AMD64, IA64, SPARC). It utilizes SWT (Standard Widget Toolkit) to render its graphical user interface by using the native resources of the underlying operating system. The Rich Client Platform is state-of-the-art in today's software development. The platform is open to be extended afterwards due to the chosen concepts. It means that the platform doesn't need to be full-fledged at the beginning. Further methods and implementations can be developed separately. Nonetheless, still some effort is necessary to design a platform that covers all needs of a software application to edit, evaluate and modify chromatographic data. In contrast to Bioclipse, Sashimi or TPP, OpenChrom has a slightly different scope, as it is focused primarily on nominal mass resolution data. Mass spectrometers for nominal mass resolution are inexpensive, as for example quadrupole or ion trap instruments. But the data acquisition limits the range of possible applications. Software has the potential to enhance the quality of the recorded data, in contrary to the given limitations. Hence, the Rich Client Platform and the Java programming language were chosen, as they offer an excellent support for a highly extensible and abstract base framework. The OSGi based Rich Client Platform Equinox supports the definition of extension points. The use of different class paths makes it possible to execute code from separated bundles (Figure 1). New functionality, e.g. to export a given chromatogram to a PDF file, can be implemented in a separate bundle making use of the extension point mechanism to import and export chromatographic data. RCP/OSGi and OpenChrom architecture. The RCP/OSGi and OpenChrom architecture shows the supported processor platforms and operating systems.Tools in different areas have been implemented based on the Rich Client Platform, such as the Eclipse IDE (Integrated Development Environment), Lotus Notes, Bioclipse, BioSunMS, XMind, Apache Directory Studio and several more. It is part of the OpenChrom architecture to define useful extension points and to build a suitable object model. OpenChrom provides a designed object model to define chromatograms, scans, mass spectra, peaks and baselines. It is important to abstract the base model, as it reduces dependencies in code and allows for the implemention of further extensions. Therefore, the decision was to support an enhanced chromatogram, mass spectrum and peak model, written in Java. There is no preliminary compilation necessary on different operating systems. Further on, it is possible to cover special needs regarding the import of instrument vendors' binary chromatographic files. An excerpt of the OpenChrom object model is shown in a simplified UML (Unified Modeling Language) diagram (Figure 2). Java, as an object orientated language, supports the use of the four base strategies in object orientation: abstraction, encapsulation, polymorphic behavior and inheritance [35]. OpenChrom makes extensive use of the object orientated concept. The interface "IChromatogram" and the abstract class "AbstractChromatogram" define and implement methods, which are common for all types of chromatograms, independent of the instrument vendors' data format. Therefore, it is not necessary to implement them iterative in each vendor specific chromatogram class. The base framework and extension points, like peak detectors and integrators, are working still with instances of the type "IChromatogram", instead of taking for example the differences of an Agilent and a NetCDF chromatogram into account. The object model for mass spectra and mass fragments, peaks and baselines is implemented in a similar way. OpenChrom chromatogram object model. The OpenChrom chromatogram object model shows a simplified UML diagram of the chromatographic model OpenChrom uses. The OpenChrom framework offers several bundles (Table 1). The most important one defines methods to implement specialized bundles that handle the import of chromatographic mass spectral data. It is possible to supply a bundle that is able to read binary chromatogram files, given by a specific instrument vendor. The bundle takes care of how to read a given file or directory. Furthermore, the framework offers extension points to detect and integrate peaks. The peak detection and integration have been separated, to make it possible to detect peaks with several peak detector methods and to integrate them with a specified integrator. This results in a more complex but also more flexible system. There is another extension point that allows to define bundles that are capable of detecting a baseline in the chromatogram model. Another flexible extension point was introduced, called filters. Bundles can extend the filter extension point to achieve a quality enhancement of the chromatographic data. They work comparable to filters in image processing software. One filter extension can for instance offer a set of methods to eliminate background signals from the chromatogram. Another filter can implement a routine to mean normalize the chromatogram. The filters offer editing steps, which are especially useful before peak detection and integration routines. Some selected bundles of the OpenChrom software. Bundle Description baseline.detector Detect baselines comparison Compare chromatograms and mass spectra converter Converter to read binary/textual data files converter.supplier.agilent Read Agilent data files converter.supplier.cdf Read and write NetCDF data files ... ... filter Modify chromatographic data identifier Identify chromatograms, mass spectra and peaks integrator Integrate peaks model Models (chromatogram, mass spectrum, peak,...) peak.detector Detect peaks logging Logging facility rcp Base application thirdpartylibraries.* Third party libraries (SWTChart, log4j,...) ... ... The Rich Client Platform offers a wide support to present an appropriate graphical user interface. Concepts detailing this include editors, views, perspectives, wizards, menus, cheat sheets, settings and help pages. OpenChrom makes extensive use of the available concepts. The editor shows the graphical representation of a chromatogram and several options, as for example a page to select or exclude distinct mass fragments. It also supports functions to save, edit and analyze chromatograms. The views are used to show different aspects of the chromatographic model. It is possible to show peaks in different kind of views. One view could show a peak including the background of the chromatogram. Another could show the peak with its increasing and decreasing tangents and its width at 50% height. A flexible mechanism was introduced to inform all views if the chromatogram selection has been changed. The update functionality is also realized by an extension point. Views and editors are composed in a task specific way using perspectives. The OpenChrom software offers several options to edit and evaluate chromatographic data. It currently implements native converters to import mass spectrometric chromatograms from Agilent Technologies and to import and export NetCDF and mzXML files as well as a custom XML format to store the chromatographic data and additional information. The chromatogram file explorer (Figure 3) shows a representation of the local file system and marks those files and directories that contain importable chromatographic data files or directories. The chromatogram can either be stored in a file, a directory or a set of files, as the converter extension point and the import and export converters take care of it. The chromatogram will be opened by a double click on the file. Additionally, a preview of the selected chromatogram file is shown in a specialized view in the user interface. The chromatogram itself is shown in a multi-page editor that is divided into a chromatogram as well as an options page. It is possible to save the chromatogram in several file formats. The NetCDF, mzXML and the customized OpenChrom XML format are actually supported. Nonetheless, the time to import and to save a chromatogram depends on its format and size. It takes more time to process XML based formats like mzXML than binary formats like NetCDF or Agilents data format. The graphical elements are drawn using SWTChart and SWT. Chromatogram selections can be chosen by applying a "zoom in" or "zoom out" action in the chromatogram editor. All views will be updated after a zoom action. OpenChrom software showing editors, views, menus and menu entries. The OpenChrom software is using editors, views, menus and menu entries showed in the figure.The menu "Chromatogram Edit" allows to access functions that modify or evaluate the chromatographic data. For example, all registered bundles that support filters will be listed in the sub menu "Filter". It is possible to add a filter that implements a Savitzky-Golay [36] smoothing operation or to add filters that remove the background of the chromatogram. Each action will be performed on the active chromatogram selection. Actions are commonly very fast, due to the fact that the chromatogram is kept in the random access memory (RAM), depending on the implemented algorithms. Furthermore, the filter actions are reversible. This editing support is well known from modern IDEs and office suites. But the support for do/undo and redo operations does cost processing time. If the reversibility is not needed, it can be deactivated in the applications preference dialog. Another extension point is responsible to register baseline detectors. Different baseline detectors can be implemented in separated bundles and will be offered in the "Baseline Detectors" sub menu. Peak detection and integration are done commonly in one run. One improvement achieved through OpenChrom is a division of the detection and the integration of peaks into two separated actions. The peak detectors can be applied by calling an appropriate detector in the sub menu "Peak Detectors" and the peak integration can be performed by using an listed integrator from the sub menu "Integrators". The separation of detector and integrator methods makes it possible to detect peaks in a chromatogram using several algorithms and methods. The chosen peak detectors could be of different types, as for example detectors using deconvolution techniques like AMDIS or CODA. All detected peaks can afterwards be integrated by a unique integrator, which leads to comparable results. This feature offers a high flexibility in using different kinds of detectors and integrators. The view mechanism of the Eclipse Rich Client Platform makes it possible to show chromatographic data in different kind of views. A peak can be displayed in multiple ways, for example by its area (Figure 4), its increasing and decreasing tangents and its width at 50% of peak height. Thus, the system provides additional graphical information, especially useful for educational purposes. Each view can be shown in a small (Figure 3) and extended format (Figure 4 and 5), which allows an appropriate user interaction even on small displays. Peak with increasing and decreasing tangents and its width at 50% height in extended format. The view shows a maximized version of a selected peak. Graphical representation of a mass spectrum in extended format. The view shows a maximized version of a selected mass spectrum.Further on, property views show miscellaneous values of the selected chromatogram. Due to the chromatogram object model, different values will be shown if different chromatogram files have been loaded. Chromatograms from Agilent Technologies and NetCDF differ in their information content. Hence, the properties view helps to inspect the files. There are additional extension points implemented that enable adding bundles to compare mass spectra using different methods [24,37-40] or to identify peaks or chromatograms. A method similar to the one implemented in the software F-Search [41] from Frontier Laboratories Ltd. could be used to identify chromatograms, for example. Moreover, the OpenChrom platform supports bundles with a system built-in logging mechanism that extends the Apache project log4j. Each module can use the logging mechanism which makes it easier to detect problems and failures. Bundles are further separated into fragments, which allows the separation of concerns. Each OpenChrom bundle supports an internationalization (i18n) and JUnit test fragment. At the moment, approximately 3000 unit tests are written and can be executed to ensure the quality of the software. If necessary, the extension point mechanism gives the flexibility to add functions needed by users at any time. Thus, OpenChrom can be connected to other systems, as for example to LIMS (Laboratory Information Management System), databases, existing software tools or workflow systems. The object model of OpenChrom offers a convenient access to values and results from the edited chromatograms. Specialized modules take care of how to handle specific concerns, for example how to store results in an information management system. Further on, it is possible to implement bundles for specific analyses or for an automated experimentation. OpenChrom enables several ways to edit and analyze chromatographic data. The advantage of the flexibility and the abstract architecture makes it partly difficult to get started with the platform, even if the functionality is provided by different bundles to decrease its complexity and to focus on special tasks. The intention to publish the software under an open source license is to support code contributions and to open the project for individual solutions. Moreover, the separation into bundles makes it easier for others to contribute new functionality. Further improvements will be done to optimize the current algorithms and to develop new and better filters, peak detectors and integrators. OpenChrom has been designed to become an extensible cross-platform open source software for the mass spectrometric analysis of chromatographic data. It provides extension points to enable built-in import capabilities for binary or textual instrument vendors' data formats. In addition to its custom XML format it supports the Agilent Technologies, mzXML and NetCDF mass spectrometric data format. Further development is planned to support more data formats. The open source concept has been chosen to initiate the contributions of third parties, as it depends on the ideas and needs of the community to extend the capabilities of the presented concept. OpenChrom offers extension points that enable the implementation of different baseline detectors as well as peak detectors and integrators. Furthermore, there is an option to implement filters, used to increase the chromatographic quality. The framework offers a full support of do/undo and redo operations. The examples Bioclipse and BioSunMS show how to use the Eclipse Rich Client Platform in a specific way, but no software has been published until now that is capable to import binary chromatographic files natively, offers support to edit and analyze chromatograms and makes it possible to implement new algorithms and methods. As it is open source, everybody has the possibility to inspect the implemented algorithms and methods, especially for verification. OpenChrom is a software with a special focus on the editing and evaluation of mass spectrometric chromatographic data. OpenChrom will be hopefully extended by contributing developers, scientists and companies in the future. Project name: OpenChrom Project homepage: Operating systems: Platform independent Programming language: Java Java Runtime Environment: Sun/Oracle JVM 1.6.0, OpenJDK Minimum RAM: 500 MB Minimum Processor: 1 GHz Commercial restrictions: none OpenChrom is available for download free of charge from the project home page. The Agilent data file input converter must be installed separately using the OpenChrom update mechanism. The instructions how to install the converter can be found at the following website: . OpenChrom is licensed under the Eclipse Public License 1.0 (EPL). The EPL is an OSI approved open source license that ensures, that the source code will remain open source. OpenChrom uses some third party libraries that are partly published under different open source licenses. All third party libraries are available in separated bundles, to ensure that no license conflicts occur. The third party library bundles are published under the Apache, LGPL, AGPL and EPL license, depending on the bundle. The GPL licenses are viral, it means that derivative works must be published under the GPL license too. The EPL and Eclipse Rich Client Platform enable a different licensing for the bundles, as a bundle using methods of another bundle can not be seen as a derivative work, though it only uses its interfaces. PW designed and implemented the core API (Application Programming Interface), the software and its extension points. PW drafted most of the manuscript. JO gave feedback and corrected the manuscript. All authors performed extensive testing of the software and approved the final manuscript. Philip Wenig, Email: philip.wenig@gmx.net. Juergen Odermatt, Email: j.odermatt@holz.uni-hamburg.de. The authors thank all participants at the Department of Wood Science (University of Hamburg, Germany) for their support and their helpful suggestions. Biller JE, Herlihy WC, Biemann K. Identification Of Components Of Complex-Mixtures By Gcms. Abstracts Of Papers Of The American Chemical Society. 1977;173(MAR20):23–23. [Google Scholar] Biller JE, Biemann K. Reconstructed Mass-Spectra - Novel Approach For Utilization Of Gas Chromatograph - Mass-Spectrometer Data. Analytical Letters. 1974;7(7):515–528. [Google Scholar] Dromey RG, Stefik MJ, Rindfleisch TC, Duffield AM. Extraction Of Mass-Spectra Free Of Background And Neighboring Component Contributions From Gas Chromatography Mass Spectrometry Data. Analytical Chemistry. 1976;48(9):1368–1375. doi: 10.1021/ac50003a027. [DOI] [Google Scholar] Colby BN. Spectral Deconvolution For Overlapping Gc Ms Components. Journal of the American Society for Mass Spectrometry. 1992;3(5):558–562. doi: 10.1016/1044-0305(92)85033-G. [DOI] [PubMed] [Google Scholar] Hindmarch P, Demir C, Brereton RG. Deconvolution and spectral clean-up of two-component mixtures by factor analysis of gas chromatographic mass spectrometric data. The Analyst. 1996;121(8):993–1001. doi: 10.1039/an9962100993. [DOI] [Google Scholar] Halket JM, Przyborowska A, Stein SE, Mallard WG, Down S, Chalmers RA. Deconvolution gas chromatography mass spectrometry of urinary organic acids - Potential for pattern recognition and automated identification of metabolic disorders. Rapid Communications In Mass Spectrometry. 1999;13(4):279–284. doi: 10.1002/(SICI)1097-0231(19990228)13:4<279::AID-RCM478>3.0.CO;2-I. [DOI] [PubMed] [Google Scholar] Kong HW, Ye F, Lu X, Guo L, Tian J, Xu GW. Deconvolution of overlapped peaks based on the exponentially modified Gaussian model in comprehensive two-dimensional gas chromatography. Journal Of Chromatography A. 2005;1086(1-2):160–164. doi: 10.1016/j.chroma.2005.05.103. [DOI] [PubMed] [Google Scholar] Taylor J, Goodacre R, Wade WG, Rowland JJ, Kell DB. The deconvolution of pyrolysis mass spectra using genetic programming: application to the identification of some Eubacterium species. FEMS Microbiology Letters. 1998;160(2):237–246. doi: 10.1111/j.1574-6968.1998.tb12917.x. [DOI] [PubMed] [Google Scholar] Pool WG, deLeeuw JW, vandeGraaf B. Backfolding applied to differential gas chromatography mass spectrometry as a mathematical enhancement of chromatographic resolution. Journal Of Mass Spectrometry. 1996;31(5):509–516. doi: 10.1002/(SICI)1096-9888(199605)31:5<509::AID-JMS323>3.0.CO;2-B. [DOI] [Google Scholar] Pool WG, deLeeuw JW, vandeGraaf B. Automated extraction of pure mass spectra from gas chromatographic mass spectrometric data. Journal Of Mass Spectrometry. 1997;32(4):438–443. doi: 10.1002/(SICI)1096-9888(199704)32:4<438::AID-JMS499>3.0.CO;2-N. [DOI] [Google Scholar] Davies A. The new Automated Mass Spectrometry Deconvolution and Identification System (AMDIS) spectrosceur. 1998;10(3):22–26. [Google Scholar] Stein SE. An integrated method for spectrum extraction and compound identification from gas chromatography/mass spectrometry data. Journal of the American Society for Mass Spectrometry. 1999;10(8):770–781. doi: 10.1016/S1044-0305(99)00047-1. [DOI] [Google Scholar] AMDIS. Windig W, Smith WF. Chemometric analysis of complex hyphenated data - Improvements of the component detection algorithm. Journal Of Chromatography A. 2007;1158(1-2):251–257. doi: 10.1016/j.chroma.2007.03.081. [DOI] [PubMed] [Google Scholar] Windig W, Phalp JM, Payne AW. A noise and background reduction method for component detection in liquid chromatography mass spectrometry. Analytical Chemistry. 1996;68(20):3602–3606. doi: 10.1021/ac960435y. [DOI] [Google Scholar] ACD Labs. Palisade Corporation. NetCDF. Pedrioli PGA, Eng JK, Hubley R, Vogelzang M, Deutsch EW, Raught B, Pratt B, Nilsson E, Angeletti RH, Apweiler R, Cheung K, Costello CE, Hermjakob H, Huang S, Julian RK, Kapp E, McComb ME, Oliver SG, Omenn G, Paton NW, Simpson R, Smith R, Taylor CF, Zhu WM, Aebersold R. A common open representation of mass spectrometry data and its application to proteomics research. Nature Biotechnology. 2004;22(11):1459–1466. doi: 10.1038/nbt1031. [DOI] [PubMed] [Google Scholar] Falkner JA, Falkner JW, Andrews PC. ProteomeCommons.org IO Framework: reading and writing multiple proteomics data formats. Bioinformatics. 2007;23(2):262–263. doi: 10.1093/bioinformatics/btl573. [DOI] [PubMed] [Google Scholar] Scientific Linux. Wienux. LiMuX. Alfassi ZB. On the normalization of a mass spectrum for comparison of two spectra. Journal of the American Society for Mass Spectrometry. 2004;15(3):385–387. doi: 10.1016/j.jasms.2003.11.008. [DOI] [PubMed] [Google Scholar] Spjuth O, Helmus T, Willighagen EL, Kuhn S, Eklund M, Wagener J, Murray-Rust P, Steinbeck C, Wikberg JE. Bioclipse: an open source workbench for chemo- and bioinformatics. BMC Bioinformatics. 2007;8:59–68. doi: 10.1186/1471-2105-8-59. [DOI] [PMC free article] [PubMed] [Google Scholar] mMass. Compari. Fityk. Cao Y, Wang N, Ying XM, Li AL, Wang HS, Zhang XM, Li WJ. BioSunMS: a plug-in-based software for the management of patients information and the analysis of peptide profiles from mass spectrometry. BMC Medical Informatics and Decision Making. 2009;9:1–9. doi: 10.1186/1472-6947-9-13. [DOI] [PMC free article] [PubMed] [Google Scholar] Steinbeck C, Han YQ, Kuhn S, Horlacher O, Luttmann E, Willighagen E. The Chemistry Development Kit (CDK): An open-source Java library for chemo- and bioinformatics. Journal of Chemical Information and Computer Sciences. 2003;43(2):493–500. doi: 10.1021/ci025584y. [DOI] [PMC free article] [PubMed] [Google Scholar] Sturm M, Bertsch A, Gropl C, Hildebrandt A, Hussong R, Lange E, Pfeifer N, Schulz-Trieglaff O, Zerck A, Reinert K, Kohlbacher O. OpenMS-An open-source software framework for mass spectrometry. BMC Bioinformatics. 2008;9:163–173. doi: 10.1186/1471-2105-9-163. [DOI] [PMC free article] [PubMed] [Google Scholar] Sashimi. TPP (Trans-Proteomic Pipeline) Eclipse Rich Client Platform. Horstmann CGCS. Core Java 2: Fundamentals. Upper Saddle River, NJ, Prentice Hall; 2002. [Google Scholar] Savitzky A, Golay MJE. Smoothing + Differentiation Of Data By Simplified Least Squares Procedures. Analytical Chemistry. 1964;36(8):1627–1639. doi: 10.1021/ac60214a047. [DOI] [Google Scholar] McLafferty FW, Zhang MY, Stauffer DB, Loh SY. Comparison of algorithms and databases for matching unknown mass spectra. Journal of the American Society for Mass Spectrometry. 1998;9:92–95. doi: 10.1016/S1044-0305(97)00235-3. [DOI] [PubMed] [Google Scholar] Loh SY, McLafferty FW. Exact-mass probability based matching of high-resolution unknown mass-spectra. Analytical Chemistry. 1991;63(6):546–550. doi: 10.1021/ac00006a002. [DOI] [Google Scholar] Damen H, Henneberg D, Weimann B. Siscom - a new library search system for mass spectra. Analytica Chimica Acta. 1978;103(4):289–302. doi: 10.1016/S0003-2670(01)83095-6. [DOI] [Google Scholar] Alfassi ZB. Vector analysis of multi-measurements identification. Journal Of Radioanalytical And Nuclear Chemistry. 2005;266(2):245–250. doi: 10.1007/s10967-005-0899-y. [DOI] [Google Scholar] Frontier Labs. OpenChrom is an open source software for chromatography and mass spectrometry based on the Eclipse Rich Client Platform (RCP). Its focus is to handle mass spectrometry systems (e.g. GC/MS, LC/MS, Py-GC/MS, HPLC-MS) data files natively. OpenChrom is able to import binary and textual chromatographic data files, such as *.D chromatograms from Agilent Technologies or NetCDF. Moreover, it offers a nice graphical user interface and is available for various operating systems, e.g. Windows, Linux, Solaris and Mac OS X. A basis set of methods to detect baselines, peaks and to integrate peaks in a chromatogram are implemented. Preprocessing steps, e.g. to remove certain mass fragments (m/z) such as nitrogen (28) or water (18), are supported by applying filter on the chromatogram or mass spectrum. Extensions are welcome, as OpenChrom is open source and uses a flexible approach, which allows others to implement their own methods, algorithms, filters, detectors or integrators. Therefore, OpenChrom should be an efficiently system to process chromatographic and mass spectrometric data using an extensible and flexible plugin approach. OpenChrom is distributed under the Eclipse Public License 1.0 (EPL). Third-party libraries are separated into single bundles and are released under various OSI compatible licenses. Find OpenChrom at: Rating: 4.5/5. From 2 votes. Please wait... This entry was posted in Analytical. Bookmark the permalink. Computer software installed on multiple computing platforms "Cross-platform" redirects here. For the railway station interchange, see cross-platform interchange. For the game term, see cross-platform play. "Multi-platform" redirects here. For the mode of storytelling in television, see multi-platform television. Within computing, cross-platform software (also called multi-platform software, platform-agnostic software, or platform-independent software) is computer software that is designed to work in several computing platforms.[1] Some cross-platform software requires a separate build for each platform, but some can be directly run on any platform without special preparation, being written in an interpreted language or compiled to portable bytecode for which the interpreters or run-time packages are common or standard components of all supported platforms.[2] For example, a cross-platform application may run on Linux, macOS and Microsoft Windows. Cross-platform software may run on many platforms, or as few as two. Some frameworks for cross-platform development are Codename One, ArkUI-X, Kivy, Qt, GTK, Flutter, NativeScript, Xamarin, Apache Cordova, Ionic, and React Native.[3] Main article: Computing platform Platform can refer to the type of processor (CPU) or other hardware on which an operating system (OS) or application runs, the type of OS, or a combination of the two.[4] An example of a common platform is Android which runs on the ARM architecture family. Other well-known platforms are Linux/Unix, macOS and Windows, these are all cross-platform.[4] Applications can be written to depend on the features of a particular platform—either the hardware, OS, or virtual machine (VM) it runs on. For example, the Java platform is a common VM platform which runs on many OSs and hardware types. A hardware platform can refer to an instruction set architecture. For example: ARM or the x86 architecture. These machines can run different operating systems. Smartphones and tablets generally run ARM architecture, these often run Android or iOS and other mobile operating systems. A software platform can be either an operating system (OS) or programming environment, though more commonly it is a combination of both. An exception is Java, which uses an OS-independent virtual machine (VM) to execute Java bytecode. Some software platforms are: Android (ARM64) ChromeOS (ARM32, ARM64, IA-32, x86-64) Common Language Infrastructure (CLI) by Microsoft, implemented in: The legacy .NET Framework that works only on Microsoft Windows. The newer .NET framework (simply called ".NET") that works across Microsoft Windows, macOS, and Linux. Other implementations such as Mono (formerly by Novell and Xamarin[5]) HarmonyOS (ARM64, RISC-V, x86, x64, and LoongArch) iOS ((ARMv8-A)) iPadOS (ARMv8-A) Java Linux (Alpha, ARC, ARM, C-Sky, Hexagon, LoongArch, m68k, Microblaze, MIPS, Nios II, OpenRISC, PA-RISC, PowerPC, RISC-V, s390, SuperH, SPARC, x86, Xtensa) macOS x86, ARM (Apple silicon) Microsoft Windows (IA-32, x86-64, ARM, ARM64) PlayStation 4 (x86), PlayStation 3 (PowerPC) and PlayStation Vita (ARM) Solaris (SPARC, x86) SPARC Unix (many platforms since 1969) Web browsers – mostly compatible with each other, running JavaScript web-apps Xbox Minor, historical AmigaOS (m68k), AmigaOS 4 (PowerPC), AROS (x86, PowerPC, m68k), MorphOS (PowerPC) Atari TOS, MiNT BSD (many platforms; see NetBSDnet,[clarification needed] for example) DOS-type systems on the x86: MS-DOS, PC DOS, DR-DOS, FreeDOS OS/2, eComStation BeOS (PowerPC, x86) Main article: Java (software platform) The Java language is typically compiled to run on a VM that is part of the Java platform. The Java virtual machine (Java VM, JVM) is a CPU implemented in software, which runs all Java code. This enables the same code to run on all systems that implement a JVM. Java software can be executed by a hardware-based Java processor. This is used mostly in embedded systems. Java code running in the JVM has access to OS-related services, like disk input/output (I/O) and network access, if the appropriate privileges are granted. The JVM makes the system calls on behalf of the Java application. This lets users to decide the appropriate protection level, depending on an access-control list (ACL). For example, disk and network access is usually enabled for desktop applications, but not for browser-based applets. The Java Native Interface (JNI) can also be used to access OS-specific functions, with a loss of portability. Currently, Java Standard Edition software can run on Microsoft Windows, macOS, several Unix-like OSs, and several real-time operating systems for embedded devices. For mobile applications, browser plugins are used for Windows and Mac based devices, and Android has built-in support for Java. There are also subsets of Java, such as Java Card or Java Platform, Micro Edition, designed for resource-constrained devices. For software to be considered cross-platform, it must function on more than one computer architecture or OS. Developing such software can be a time-consuming task because different OSs have different application programming interfaces (API). Software written for one OS may not automatically work on all architectures that OS supports. Just because software is written in a popular programming language such as C or C++, it does not mean it will run on all OSs that support that language—or even on different versions of the same OS. Web applications are typically described as cross-platform because, ideally, they are accessible from any web browser: the browser is the platform. Web applications generally employ a client–server model, but vary widely in complexity and functionality. It can be hard to reconcile the desire for features with the need for compatibility. Basic web applications perform all or most processing from a stateless server, and pass the result to the client web browser. All user interaction with the application consists of simple exchanges of data requests and server responses. This type of application was the norm in the early phases of World Wide Web application development. Such applications follow a simple transaction model, identical to that of serving static web pages. Today, they are still relatively common, especially where cross-platform compatibility and simplicity are deemed more critical than advanced functionality. Prominent examples of advanced web applications include the Web interface to Gmail and Google Maps. Such applications routinely depend on additional features found only in the more recent versions of popular web browsers. These features include Ajax, JavaScript, Dynamic HTML, SVG, and other components of rich web applications. Because of the competing interests of compatibility and functionality, numerous design strategies have emerged. Many software systems use a layered architecture where platform-dependent code is restricted to the upper- and lowermost layers. Graceful degradation attempts to provide the same or similar functionality to all users and platforms, while diminishing that functionality to a least common denominator for more limited client browsers. For example, a user attempting to use a limited-feature browser to access Gmail may notice that Gmail switches to basic mode, with reduced functionality but still of use. Some software is maintained in distinct codebases for different (hardware and OS) platforms, with equivalent functionality. This requires more effort to maintain the code, but can be worthwhile where the amount of platform-specific code is high. This strategy relies on having one codebase that may be compiled to multiple platform-specific formats. One technique is conditional compilation. With this technique, code that is common to all platforms is not repeated. Blocks of code that are only relevant to certain platforms are made conditional, so that they are only interpreted or compiled when needed. Another technique is separation of functionality, which disables functionality not supported by browsers or OSs, while still delivering a complete application to the user. (See also: Separation of concerns.) This technique is used in web development where interpreted code (as in scripting languages) can query the platform it is running on to execute different blocks conditionally.[6] Third-party libraries attempt to simplify cross-platform capability by hiding the complexities of client differentiation behind a single, unified API, at the expense of vendor lock-in. Responsive web design (RWD) is a Web design approach aimed at crafting the visual layout of sites to provide an optimal viewing experience—easy reading and navigation with a minimum of resizing, panning, and scrolling—across a wide range of devices, from mobile phones to desktop computer monitors. Little or no platform-specific code is used with this technique. Cross-platform applications need much more integration testing. Some web browsers prohibit installation of different versions on the same machine. There are several approaches used to target multiple platforms, but all of them result in software that requires substantial manual effort for testing and maintenance.[7] Techniques such as full virtualization are sometimes used as a workaround for this problem. Tools such as the Page Object Model allow cross-platform tests to be scripted so that one test case covers multiple versions of an app. If different versions have similar user interfaces, all can be tested with one test case. Web applications are becoming increasingly popular but many computer users still use traditional application software which does not rely on a client/web-server architecture. The distinction between traditional and web applications is not always clear. Features, installation methods and architectures for web and traditional applications overlap and blur the distinction. Nevertheless, this simplifying distinction is a common and useful generalization. Traditional application software has been distributed as binary files, especially executable files. Executables only support the platform they were built for—which means that a single cross-platform executable could be very bloated with code that never executes on a particular platform. Instead, generally there is a selection of executables, each built for one platform. For software that is distributed as a binary executable, such as that written in C or C++, there must be a software build for each platform, using a toolset that translates—transcompiles—a single codebase into multiple binary executables. For example, Firefox, an open-source web browser, is available on Windows, macOS (both PowerPC and x86 through what Apple Inc. calls a Universal binary), Linux, and BSD on multiple computer architectures. The four platforms (in this case, Windows, macOS, Linux, and BSD) are separate executable distributions, although they come largely from the same source code. In rare cases, executable code built for several platforms is combined into a single executable file called a fat binary. The use of different toolsets may not be enough to build a working executables for different platforms. In this case, programmers must port the source code to the new platform. For example, an application such as Firefox, which already runs on Windows on the x86 family, can be modified and re-built to run on Linux on the x86 (and potentially other architectures) as well. The multiple versions of the code may be stored as separate codebases, or merged into one codebase. An alternative to porting is cross-platform virtualization, where applications compiled for one platform can run on another without modification of the source code or binaries. As an example, Apple's Rosetta, which is built into Intel-based Macintosh computers, runs applications compiled for the previous generation of Macs that used PowerPC CPUs. Another example is IBM PowerVM Lx86, which allows Linux/x86 applications to run unmodified on the Linux/Power OS. Example of cross-platform binary software: The LibreOffice office suite is built for Microsoft Windows, macOS, Linux, FreeBSD, NetBSD, OpenBSD, Android, iOS, iPadOS, ChromeOS, web-based Collabora Online and many others.[8][9] Many of these are supported on several hardware platforms with processor architectures including IA-32, x86-64, ARM (ARMel, ARMhf, ARM64), MIPS, MIPSel, PowerPC, ppc64le, and S390x[9][10] A script can be considered to be cross-platform if its interpreter is available on multiple platforms and the script only uses the facilities built into the language. For example, a script written in Python for a Unix-like system will likely run with little or no modification on Windows, because Python also runs on Windows; indeed there are many implementations (e.g. IronPython for .NET Framework). The same goes for many of the open-source scripting languages. Unlike binary executable files, the same script can be used on all computers that have software to interpret the script. This is because the script is generally stored in plain text in a text file. There may be some trivial issues, such as the representation of a new line character. Some popular cross-platform scripting languages are: bash – A Unix shell commonly run on Linux and other modern Unix-like systems, as well as on Windows via the Cygwin POSIX compatibility layer, Git for Windows, or the Windows Subsystem for Linux. Perl – First released in 1987. Used for CGI programming, small system administration tasks, and more. PHP – Mostly used for web applications. Python – A language which focuses on rapid application development and ease of writing, instead of run-time efficiency. Ruby – An object-oriented language which aims to be easy to read. Can also be used on the web through Ruby on Rails. Tcl – A dynamic programming language, suitable for a wide range of uses, including web and desktop applications, networking, administration, testing and many more. Cross-platform or multi-platform is a term that can also apply to video games released on a range of video game consoles. Examples of cross-platform games include: Miner 2049er, Tomb Raider: Legend, FIFA series, NHL series and Minecraft. Each has been released across a variety of gaming platforms, such as the Wii, PlayStation 3, Xbox 360, personal computers, and mobile devices. Some platforms are harder to write for than others, requiring more time to develop the video game to the same standard. To offset this, a video game may be released on a few platforms first, then later on others. Typically, this happens when a new gaming system is released, because video game developers need to acquaint themselves with its hardware and software. Some games may not be cross-platform because of licensing agreements between developers and video game console manufacturers that limit development to one particular console. As an example, Disney could create a game with the intention of release on the latest Nintendo and Sony game consoles. Should Disney license the game with Sony first, it may be required to release the game solely on Sony's console for a short time or indefinitely. Main articles: Cross-platform play and List of video games that support cross-platform play Several developers have implemented ways to play games online while using different platforms. Psyonix, Epic Games, Microsoft, and Valve all possess technology that allows Xbox 360 and PlayStation 3 gamers to play with PC gamers, leaving the decision of which platform to use to consumers. The first game to allow this level of interactivity between PC and console games (Dreamcast with specially produced keyboard and mouse) was Quake 3.[11][12] Games that feature cross-platform online play include Rocket League, Final Fantasy XIV, Street Fighter V, Killer Instinct, Paragon and Fable Fortune, and Minecraft with its Better Together update on Windows 10, VR editions, Pocket Edition and Xbox One. Cross-platform programming is the practice of deliberately writing software to work on more than one platform. There are different ways to write a cross-platform application. One approach is to create multiple versions of the same software in different source trees—in other words, the Microsoft Windows version of an application might have one set of source code files and the Macintosh version another, while a FOSS *nix system might have a third. While this is straightforward, compared to developing for only one platform it can cost much more to pay a larger team or release products more slowly. It can also result in more bugs to be tracked and fixed. Another approach is to use software that hides the differences between the platforms. This abstraction layer insulates the application from the platform. Such applications are platform agnostic. Applications that run on the JVM are built this way. Some applications mix various methods of cross-platform programming to create the final application. An example is the Firefox web browser, which uses abstraction to build some of the lower-level components, with separate source subtrees for implementing platform-specific features (like the GUI), and the implementation of more than one scripting language to ease software portability. Firefox implements XUL, CSS and JavaScript for extending the browser, in addition to classic Netscape-style browser plugins. Much of the browser itself is written in XUL, CSS, and JavaScript. There are many tools[13][14] available to help the process of cross-platform programming: 8th: a development language which utilizes Juce as its GUI layer. It currently supports Android, iOS, Windows, macOS, Linux and Raspberry Pi. Anant Computing: A mobile application platform that works in all Indian languages, including their keyboards, and also supports AppWallet and native performance in all OSs. AppearIQ: a framework that supports the workflow of app development and deployment in an enterprise environment. Natively developed containers present hardware features of the mobile devices or tablets through an API to HTML5 code thus facilitating the development of mobile apps that run on different platforms. Boden: a UI framework written in C++. Cairo: a free software library used to provide a vector graphics-based, device-independent API. It is designed to provide primitives for 2-dimensional drawing across a number of different backends. Cairo is written in C and has bindings for many programming languages. Cocos2d: an open-source toolkit and game engine for developing 2D and simple 3D cross-platform games and applications. Codename One: an open-source Write Once Run Anywhere (WORA) framework for Java and Kotlin developers. Delphi: an IDE which uses a Pascal-based language for development. It supports Android, iOS, Windows, macOS, Linux. Ecere SDK: a GUI and 2D/3D graphics toolkit and IDE, written in eC and with support for additional languages such as C and Python. It supports Linux, FreeBSD, Windows, Android, macOS and the Web through Emscripten or Binaryen [Wikidata] (WebAssembly). Eclipse: an open-source development environment. Implemented in Java with a configurable architecture which supports many tools for software development. Add-ons are available for several languages, including Java and C++. FLTK: an open-source toolkit, but more lightweight because it restricts itself to the GUI. Flutter: A cross-platform UI framework for IOS, Android, Mac, Windows and developed by Google. fpGUI: An open-source widget toolkit that is completely implemented in Object Pascal. It currently supports Linux, Windows and a bit of Windows CE. GeneXus: A Windows rapid software development solution for cross-platform application creation and deployment based on knowledge representation and supporting C#, COBOL, Java including Android and BlackBerry smart devices, Objective-C for Apple mobile devices, RPG, Ruby, Visual Basic, and Visual FoxPro. GLBasic: A BASIC dialect and compiler that generates C++ code. It includes cross compilers for many platforms and supports numerous platform (Windows, Mac, Linux, Android, iOS and some exotic handhelds). Godot: an SDK which uses Godot Engine. GTK+: An open-source widget toolkit for Unix-like systems with X11 and Microsoft Windows. Haxe: An open-source language. Juce: An application framework written in C++, used to write native software on numerous systems (Microsoft Windows, POSIX, macOS), with no change to the code. Kivy: an open-source cross-platform UI framework written in Python. It supports Android, iOS, Linux, OS X, Windows and Raspberry Pi. LEADTOOLS: Cross-platform SDK libraries to integrate recognition, document, medical, imaging, and multimedia technologies into Windows, iOS, macOS, Android, Linux and web applications.[15] LiveCode: a commercial cross-platform rapid application development language inspired by HyperTalk. Lazarus: A programming environment for the FreePascal Compiler. It supports the creation of self-standing graphical and console applications and runs on Linux, MacOSX, iOS, Android, WinCE, Windows and WEB. Max/MSP: A visual programming language that encapsulates platform-independent code with a platform-specific runtime environment into applications for macOS and Windows A cross-platform Android runtime. It allows unmodified Android apps to run natively on iOS and macOS Mendix: a cloud-based low-code application development platform. MonoCross: an open-source model–view–controller design pattern where the model and controller are cross-platform but the view is platform-specific.[16] Mono: An open-source cross-platform version of Microsoft .NET (a framework for applications and programming languages) MoSync: an open-source SDK for mobile platform app development in the C++ family. Mozilla application framework: an open-source platform for building macOS, Windows and Linux applications. OpenGL: a 3D graphics library. Pixel Game Maker MV: A proprietary 2D game development software for Windows for developing Windows and Nintendo Switch games. PureBasic: a proprietary language and IDE for building macOS, Windows and Linux applications. ReNative: The universal development SDK to build multi-platform projects with React Native. Includes latest iOS, tvOS, Android, Android TV, Web, Tizen TV, Tizen Watch, LG webOS, macOS/OSX, Windows, KaiOS, Firefox OS and Firefox TV platforms. Qt: an application framework and widget toolkit for Unix-like systems with X11, Microsoft Windows, macOS, and

other systems—available under both proprietary and open-source licenses. Simple and Fast Multimedia Library: A multimedia C++ API that provides low and high level access to graphics, input, audio, etc. Simple DirectMedia Layer: an open-source multimedia library written in C that creates an abstraction over various platforms' graphics, sound, and input APIs. It runs on OSs including Linux, Windows and macOS and is aimed at games and multimedia applications. Smartface: a native app development tool to create mobile applications for Android and iOS, using WYSIWYG design editor with JavaScript code editor. Tcl/Tk Titanium Mobile: open source cross-platform framework for Android and iOS development. U++: a C++ GUI framework for performance. It includes a set of libraries (GUI, SQL, etc..), and IDE. It supports Windows, macOS and Linux. Unity: Another cross-platform SDK which uses Unity Engine. Uno Platform: Windows, macOS, iOS, Android, WebAssembly and Linux using C#. Unreal: A cross-platform SDK which uses Unreal Engine. V-Play Engine: V-Play is a cross-platform development SDK based on the popular Qt framework. V-Play apps and games are created within Qt Creator. WaveMaker: A low-code development tool to create responsive web and hybrid mobile (Android & iOS) applications. WinDev: an Integrated Development Environment for Windows, Linux, .Net and Java, and web browers. Optimized for business and industrial applications. wxWidgets: an open-source widget toolkit that is also an application framework.[17] It runs on Unix-like systems with X11, Microsoft Windows and macOS. Xojo: a RAD IDE that uses an object-oriented programming language to compile desktop, web and iOS apps. Xojo supports natively compiling to Windows, macOS, iOS and Linux, and can also create compiled web apps that are able to be run as standalone servers or through CGI. This section possibly contains original research. Please improve it by verifying the claims made and adding inline citations. Statements consisting only of original research should be removed. (March 2025) (Learn how and when to remove this message) There are many challenges when developing cross-platform software: Testing cross-platform applications may be considerably more complicated, since different platforms can exhibit slightly different behaviors or subtle bugs. This problem has led some developers to deride cross-platform development as "write once, debug everywhere", a take on Sun Microsystems' "write once, run anywhere" marketing slogan. Developers are often restricted to using the lowest common denominator subset of features which are available on all platforms. This may hinder the application's performance or prohibit developers from using the most advanced features of each platform. Different platforms often have different user interface conventions, which cross-platform applications do not always accommodate. For example, applications developed for macOS and GNOME are supposed to place the most important button on the right-hand side of a window or dialog, whereas Microsoft Windows and KDE have the opposite convention. Though many of these differences are subtle, a cross-platform application which does not conform to these conventions may feel clunky or alien to the user. When working quickly, such opposing conventions may even result in data loss, such as in a dialog box confirming whether to save or discard changes. Scripting languages and VM bytecode must be translated into native executable code each time they are used, imposing a performance penalty. This penalty can be alleviated using techniques like just-in-time compilation; but some computational overhead may be unavoidable. Different platforms require the use of native package formats such as RPM and MSI. Multi-platform installers such as InstallAnywhere address this need. Cross-platform execution environments may suffer cross-platform security flaws, creating a fertile environment for cross-platform malware.[18] Operating context List of widget toolkits Hardware virtualization Language binding Source-to-source compiler Binary-code compatibility Comparison of user features of messaging platforms ^ "Design Guidelines: Glossary". java.sun.com. Archived from the original on 2012-02-13. Retrieved 2011-10-19. ^ "SDD Technology blog: Definition of cross platform". SDD Technology. Retrieved 2020-10-18. ^ Lee P Richardson (2016-02-16). "Xamarin vs Ionic: A Mobile, Cross Platform, Shootout". ^ a b "Platform Definition". The Linux Information Project. Retrieved 2014-03-27. ^ "About Mono". mono-project.com. Retrieved 2015-12-17. ^ Corti, Sascha P. (October 2011). "Browser and Feature Detection". MSDN Magazine. Retrieved 28 January 2014. ^ Choudhary, S.R. (2014). "Cross-platform testing and maintenance of web and mobile applications". Companion Proceedings of the 36th International Conference on Software Engineering. pp. 642–645. doi:10.1145/2591062.2591097. hdl:1853/53588. ISBN 9781450327688. S2CID 1903037. ^ Mehrotra, Pranob (2020-12-01). "Collabora Office suite gets a new layout for Android tablets and Chromebooks". XDA-Developers. Retrieved 2021-01-15. Collabora Office is a popular open-source alternative to the Microsoft Office suite. It's based on LibreOffice, and it's available on a variety of platforms, including Windows, Linux, iOS, and Android. This year in July, a major update for the office suite brought support for Chrome OS devices. ^ "Collabora Office on iOS and Android Just got Better!". Adfinis. 2020-12-15. Retrieved 2021-01-15. ...touch optimized interfaces: one for tablets and one for phone screens. ...(iOS, iPadOS, Chromebooks, Android). ^ "Nextcloud Ubuntu Appliance adds Collabora Online to Raspberry Pi image". MuyLinux. 2021-03-26. Retrieved 2021-03-30. the first viable self-hosted web office solution for the popular Raspberry Pi 4 platform ^ Cribba. Quake III Arena, Giant Bombcast, February 15, 2013. ^ A Closer Look At The Dreamcast Internet Starter Kit ^ The GUI Toolkit, Framework Page ^ "Platform Independent FAQ". Archived from the original on 2008-08-16. Retrieved 2009-04-25. ^ "Cross-Platform SDK Libraries for Recognition, Document, Medical, Imaging, and Multimedia". www.leadtools.com. Retrieved 2021-03-03. ^ "12 benefits of Xamarin Cross-platform app development". HeadWorks. 15 Mar 2019. ^ WxWidgets Description ^ Warren, Tom (2020-01-14). "Microsoft bids farewell to Windows 7 and the millions of PCs that still run it". The Verge. Retrieved 2020-02-06. Retrieved from " You can’t perform that action at this time.