

Programming Languages and Data Access Methods							July 2024						
Manufacturer	Programming Language Name	Version	Current Classification	General Availability	Projected	Approved	Divest: Plan	Divest: Execution	Prohibited	Vendor End of Support	References	Notes	
General programming language recommendation	NA	NA	NA	NA	NA	NA	NA	NA	NA	NA	<a href="https://media.defense.gov/2022/Nov/10/2003112742/-1/-1/0/CSL_SOFTWARE_MEMORY_SAFETY.PDF">https://media.defense.gov/2022/Nov/10/2003112742/-1/-1/0/CSL_SOFTWARE_MEMORY_SAFETY.PDF</a> <a href="https://insights.sei.cmu.edu/blog/rust-software-security-a-current-state-assessment/">https://insights.sei.cmu.edu/blog/rust-software-security-a-current-state-assessment/</a> <a href="https://survey.stackoverflow.co/2022/">https://survey.stackoverflow.co/2022/</a>	<p><b>EA Direction:</b> Use of a "memory safe" language is recommended and preferred.</p> <p><b>Examples</b> of memory safe languages include: C#, Go, Java®, Ruby™, Rust®, Python, and Swift®.</p> <p>Use memory safe languages! There are lots of great ones to choose from. Writing an operating system kernel or web browser? Consider Rust! Building for iOS and macOS? Swift's got you covered. Network server? Go's a fine choice. <a href="https://www.memorysafety.org/docs/memory-safety/">https://www.memorysafety.org/docs/memory-safety/</a></p> <p>The White House is urging programmers to move away from older programming languages like C and C++ in favor of "memory-safe" languages like Rust. There's evidence that building software with memory-safe languages can significantly reduce vulnerabilities and prevent cyberattacks, according to a report issued Monday.</p> <p><a href="https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf">https://www.whitehouse.gov/wp-content/uploads/2024/02/Final-ONCD-Technical-Report.pdf</a>  <a href="https://www.codecademy.com/resources/blog/memory-safe-programming-languages/">https://www.codecademy.com/resources/blog/memory-safe-programming-languages/</a></p> <p>In no particular order, the NSA suggests these memory-safe programming languages: Go; Rust; C#; Swift; Java; Ruby; Python; Delphi/Object Pascal; and Ada. <a href="https://readwrite.com/the-nsa-list-of-memory-safe-programming-languages-has-been-updated/">https://readwrite.com/the-nsa-list-of-memory-safe-programming-languages-has-been-updated/</a></p> <p>The data bears out, over and over again, that when projects use unsafe languages like C and C++ they are burdened by an avalanche of security vulnerabilities. No matter how talented the engineers, how great the investment in privilege reduction and exploit mitigations, using a language that is not memory safe simply results in too many bugs. These bugs greatly reduce security, as well as stability and productivity.</p> <p><a href="https://www.memorysafety.org/docs/memory-safety/">https://www.memorysafety.org/docs/memory-safety/</a></p> <p>Commonly used languages, such as C and C++, provide a lot of freedom and flexibility in memory management while relying heavily on the programmer to perform the needed checks on memory references. Simple mistakes can lead to exploitable memory-based vulnerabilities. Software analysis tools can detect many instances of memory management issues and operating environment options can also provide some protection, but inherent protections offered by memory safe software languages can prevent or mitigate most memory management issues. NSA recommends using a memory safe language when possible. While the use of added protections to non-memory safe languages and the use of memory safe languages do not provide absolute protection against exploitable memory issues, they do provide considerable protection. Therefore, the overarching software community across the private sector, academia, and the U.S. Government have begun initiatives to drive the culture of software development towards utilizing memory safe languages.</p> <p><a href="https://endoflife.date/">https://endoflife.date/</a></p>	
Oracle	Java	25 (LTS)	Projected		2026						End of Life Info: <a href="https://endoflife.date/java">https://endoflife.date/java</a>	<p><b>EA Direction:</b> Continue updating to the latest LTS major release.</p> <p><a href="https://en.wikipedia.org/wiki/Java_(programming_language)">https://en.wikipedia.org/wiki/Java_(programming_language)</a></p>	
		21 (LTS)	Approved	9/19/2023	N	12/31/2023	9/30/2026	9/30/2027	9/30/2028	9/30/2029	End of Life Info: <a href="https://endoflife.date/java">https://endoflife.date/java</a>	<p><a href="https://en.wikipedia.org/wiki/Java_(programming_language)">https://en.wikipedia.org/wiki/Java_(programming_language)</a></p> <p>Note: Tracking N and N-1 to LTS versions only while skipping all other version between LTS versions.</p>	
		20	Skipped										
		19	Skipped										
		18	Skipped										
		17 (LTS)	Approved	9/14/2021	N-1	12/31/2021	9/30/2025	9/30/2026	9/30/2027	9/30/2029	End of Life Info: <a href="https://endoflife.date/java">https://endoflife.date/java</a>	<p>Java, as developed by the OpenJDK Project, which is owned and primarily employed by Oracle, has been on a 6-month rapid-release cycle since the release of Java 10. Starting with Java 11, new LTS releases occur every six releases, or three years. Java 8 is the last release on the old cycle methodology still in active support. Non-LTS releases are supported for 6 months. The latest supported release in each release cycle can be found at <a href="https://www.oracle.com/java/technologies/java-se-glance.html">https://www.oracle.com/java/technologies/java-se-glance.html</a>.</p>	
		16	Skipped	3/16/2021					Prohibited	9/30/2021			
		15	Skipped	9/16/2020					Prohibited	3/31/2021			
		14	Skipped	3/17/2020					Prohibited	9/30/2020			
		13	Skipped	9/17/2019					Prohibited	3/31/2021			
		12	Skipped	3/19/2019					Prohibited	9/30/2019			
		11 (LTS)	Divest Plan	9/25/2018			9/30/2023	9/30/2024	9/30/2025	9/30/2026	End of Life Info: <a href="https://endoflife.date/java">https://endoflife.date/java</a>		
		10	Skipped	3/20/2018					Prohibited	9/25/2018			
		9	Skipped	9/21/2017					Prohibited	3/20/2018			
		8 (LTS)	Prohibited	3/18/2014					Prohibited	3/31/2025			
		7 and earlier	Prohibited	7/7/2011					Prohibited	7/31/2019		<p>First version GA May 1995 from Sun Microsystems.</p> <p><a href="https://en.wikipedia.org/wiki/Java_(programming_language)">https://en.wikipedia.org/wiki/Java_(programming_language)</a></p>	
Oracle	PL/SQL	All versions	Approved	1995							<a href="https://en.wikipedia.org/wiki/PL/SQL">https://en.wikipedia.org/wiki/PL/SQL</a>	<p>Oracle: <a href="https://www.oracletutorial.com/plsql-tutorial/what-is-plsql/">https://www.oracletutorial.com/plsql-tutorial/what-is-plsql/</a></p> <p>Tech on the Net: <a href="https://www.techonthenet.com/oracle/index.php">https://www.techonthenet.com/oracle/index.php</a></p> <p><a href="https://endoflife.date/">https://endoflife.date/</a></p>	

Programming Languages and Data Access Methods							July 2024						
Manufacturer	Programming Language Name	Version	Current Classification	General Availability	Projected	Approved	Divest: Plan	Divest: Execution	Prohibited	Vendor End of Support	References	Notes	
Adobe	ColdFusion	2023	Divest Plan	5/31/2023			1/1/2024	6/1/2027	1/1/2028	5/16/2028	Support Date Info: <a href="https://helpx.adobe.com/support/programs/eol-matrix.html">https://helpx.adobe.com/support/programs/eol-matrix.html</a>	<p><b>EA Direction:</b> As applicable update to ColdFusion 2023. Long term - proceed to divest from the ColdFusion programming language entirely prior to it being prohibited in the COV enterprise in 2029.</p> <p><b>EA Rationale:</b> Supporting too many programming languages can create long-term maintenance issues and skill shortage issues within the COV IT enterprise. Licensing, packaging, frameworks, extended ecosystems and support are key considerations in selecting any programming language.</p> <p><b>EA goal:</b> Reduce the quantity of programming languages we officially support. Per Gartner (Dec-2021) and industry programming language trends as evidenced by job postings, the top languages were: Python; Java; JavaScript; C/C++; C#; and PHP. While ColdFusion's CMFL compares to PHP's scripting components, it's tag syntax resembles HTML and it's script syntax resembles JavaScript, EA's position is to divest from this programming language and to not continue investing in it.</p>	
		2021	Divest Plan	11/11/2020			1/1/2024	1/1/2026	6/1/2026	11/10/2026	Support Date Info: <a href="https://helpx.adobe.com/support/programs/eol-matrix.html">https://helpx.adobe.com/support/programs/eol-matrix.html</a>	<p>Release graphic: <a href="https://endoflife.date/coldfusion">https://endoflife.date/coldfusion</a></p> <p>Release announcement: <a href="https://community.adobe.com/t5/coldfusion-discussions/live-introducing-the-2023-release-of-adobe-coldfusion/td-p/13797706">https://community.adobe.com/t5/coldfusion-discussions/live-introducing-the-2023-release-of-adobe-coldfusion/td-p/13797706</a></p> <p>Downloads: <a href="https://helpx.adobe.com/coldfusion/kb/coldfusion-downloads.html#download0">https://helpx.adobe.com/coldfusion/kb/coldfusion-downloads.html#download0</a></p> <p>ColdFusion's lifecycle is typically 5 years after release, with new releases usually about every two years. Adobe never announces the release date in advance. <a href="https://community.adobe.com/t5/coldfusion-discussions/coldfusion-2023-release-date/td-p/13785784">https://community.adobe.com/t5/coldfusion-discussions/coldfusion-2023-release-date/td-p/13785784</a>.  Adobe release info: <a href="https://helpx.adobe.com/coldfusion/kb/coldfusion-downloads.html#downloads1">https://helpx.adobe.com/coldfusion/kb/coldfusion-downloads.html#downloads1</a></p>	
		2018 and earlier versions	Prohibited	7/12/2018						7/13/2024	Support Date Info: <a href="https://helpx.adobe.com/support/programs/eol-matrix.html">https://helpx.adobe.com/support/programs/eol-matrix.html</a>	<p>One of the distinguishing features of ColdFusion is its associated scripting language, ColdFusion Markup Language (CFML). CFML compares to the scripting components of ASP, JSP, and PHP in purpose and features, but its tag syntax more closely resembles HTML, while its script syntax resembles JavaScript. ColdFusion is often used synonymously with CFML, but there are additional CFML application servers besides ColdFusion, and ColdFusion supports programming languages other than CFML, such as server-side Actionscript and embedded scripts that can be written in a JavaScript-like language known as CFScript.</p> <p>Originally a product of Allaire and released on July 2, 1995, ColdFusion was developed by brothers Joseph J. Allaire and Jeremy Allaire. In 2001 Allaire was acquired by Macromedia, which in turn was acquired by Adobe Systems Inc in 2005. ColdFusion is most often used for data-driven websites or intranets, but can also be used to generate remote services such as REST services, WebSockets, SOAP web services or Flash remoting. It is especially well-suited as the server-side technology to the client-side ajax. ColdFusion can also handle asynchronous events such as SMS and instant messaging via its gateway interface, available in ColdFusion MX 7 Enterprise Edition. <a href="https://en.wikipedia.org/wiki/Adobe_ColdFusion">https://en.wikipedia.org/wiki/Adobe_ColdFusion</a></p>	
CII Honewell Bull	Ada	2012	Prohibited	2012							<a href="https://en.wikipedia.org/wiki/Ada_(programming_language)">https://en.wikipedia.org/wiki/Ada_(programming_language)</a>	<a href="https://en.wikipedia.org/wiki/Groupe_Bull">https://en.wikipedia.org/wiki/Groupe_Bull</a> <a href="https://curie.org/Computers/Programming/Languages/Ada">https://curie.org/Computers/Programming/Languages/Ada</a>	
Association for Computing Machinery (ACM)	ALGOL	DG/L	Prohibited	1972							<a href="https://en.wikipedia.org/wiki/ALGOL">https://en.wikipedia.org/wiki/ALGOL</a>		
IBM	APL	APL2	Prohibited	1984							<a href="https://en.wikipedia.org/wiki/APL_(programming_language)">https://en.wikipedia.org/wiki/APL_(programming_language)</a>		
IBM	Assembler	BAL	Prohibited	1964							<a href="https://en.wikipedia.org/wiki/IBM_Basic_Assembly_Language_and_successors">https://en.wikipedia.org/wiki/IBM_Basic_Assembly_Language_and_successors</a>		
Dartmouth	BASIC	All versions	Prohibited	1964							<a href="https://en.wikipedia.org/wiki/BASIC">https://en.wikipedia.org/wiki/BASIC</a>		
Bell Labs	"C"	All versions	Prohibited	1973							<a href="https://en.wikipedia.org/wiki/The_C_Programming_Language">https://en.wikipedia.org/wiki/The_C_Programming_Language</a>	<a href="https://www.section.io/engineering-education/history-of-c-programming-language/">https://www.section.io/engineering-education/history-of-c-programming-language/</a>	
Nantucket Corporation	Clipper	xBase	Prohibited	1997							<a href="https://en.wikipedia.org/wiki/Clipper_(programming_language)">https://en.wikipedia.org/wiki/Clipper_(programming_language)</a>	aka CA-Clipper	

Programming Languages and Data Access Methods							July 2024						
Manufacturer	Programming Language Name	Version	Current Classification	General Availability	Projected	Approved	Divest: Plan	Divest: Execution	Prohibited	Vendor End of Support	References	Notes	
CODASYL	COBOL	All versions	Prohibited	1959							<a href="https://en.wikipedia.org/wiki/COBOL">https://en.wikipedia.org/wiki/COBOL</a>	<a href="https://en.wikipedia.org/wiki/CODASYL">https://en.wikipedia.org/wiki/CODASYL</a>	
Embarcadero Technologies	Delphi	All versions	Prohibited	1995							<a href="https://en.wikipedia.org/wiki/Delphi_(software)">https://en.wikipedia.org/wiki/Delphi_(software)</a>	<a href="https://en.wikipedia.org/wiki/History_of_Delphi_(software)">https://en.wikipedia.org/wiki/History_of_Delphi_(software)</a>	
IBM	Fortran	All versions	Prohibited	1957							<a href="https://en.wikipedia.org/wiki/Fortran">https://en.wikipedia.org/wiki/Fortran</a>		
Software AG	Natural		Prohibited	1979							<a href="https://en.wikipedia.org/wiki/ADABAS#Natural_(4GL)">https://en.wikipedia.org/wiki/ADABAS#Natural_(4GL)</a>	<a href="https://en.wikipedia.org/wiki/Natural_language">https://en.wikipedia.org/wiki/Natural_language</a> <a href="https://www.xenonstack.com/blog/evolution-of-nlp/">https://www.xenonstack.com/blog/evolution-of-nlp/</a>	
MIT	Lisp	All LISP-based languages	Prohibited	1958							<a href="https://en.wikipedia.org/wiki/Lisp_(programming_language)">https://en.wikipedia.org/wiki/Lisp_(programming_language)</a>	<a href="https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#List-based_languages_%E2%80%93_LISPs">https://en.wikipedia.org/wiki/List_of_programming_languages_by_type#List-based_languages_%E2%80%93_LISPs</a>	
Unisys / Sperry	Mapper	All versions	Prohibited	1975							<a href="https://en.wikipedia.org/wiki/MAPPER">https://en.wikipedia.org/wiki/MAPPER</a>		
Corel	Paradox	All versions	Prohibited	1985							<a href="https://en.wikipedia.org/wiki/Paradox_(database)">https://en.wikipedia.org/wiki/Paradox_(database)</a>		
Berkeley	Pascal	All versions	Prohibited	1970							<a href="https://en.wikipedia.org/wiki/Pascal_(programming_language)">https://en.wikipedia.org/wiki/Pascal_(programming_language)</a>	<a href="https://www.britannica.com/technology/Pascal-computer-language">https://www.britannica.com/technology/Pascal-computer-language</a>	
IBM	PL/I	All versions	Prohibited	1964							<a href="https://en.wikipedia.org/wiki/PL/I">https://en.wikipedia.org/wiki/PL/I</a>	aka PL/1	
IBM	Rexx	All versions	Prohibited	1979							<a href="https://en.wikipedia.org/wiki/Rexx">https://en.wikipedia.org/wiki/Rexx</a>		
IBM	RPG X	All versions	Prohibited	1959							<a href="https://en.wikipedia.org/wiki/IBM_RPG">https://en.wikipedia.org/wiki/IBM_RPG</a>		

**Programming Languages and Data Access Methods** **July 2024**

Manufacturer	Programming Language Name	Version	Current Classification	General Availability	Projected	Approved	Divest: Plan	Divest: Execution	Prohibited	Vendor End of Support	References	Notes
--------------	---------------------------	---------	------------------------	----------------------	-----------	----------	--------------	-------------------	------------	-----------------------	------------	-------

The following **data access methods** are considered *Prohibited* technologies per Enterprise Technical Architecture (ETA) direction:

<https://www.vita.virginia.gov/media/vitavirginiagov/it-governance/ea/pdf/Legacy-IT-Solutions-Topic-Report.pdf>

Vendor	Data Access Name	Version	Current Classification	General Availability	Divest: Plan	Divest: Execution	Divest: Plan	Divest: Execution	Prohibited	Vendor End of Support	References	Notes
	Adabas		Prohibited								<a href="https://documentation.softwareag.com/natural/nat6313win/pg/pg_dbms_ada.htm">https://documentation.softwareag.com/natural/nat6313win/pg/pg_dbms_ada.htm</a>	<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>
	IMS		Prohibited								<a href="https://flylib.com/books/en/2.869.1.57/1/">https://flylib.com/books/en/2.869.1.57/1/</a>	<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a> <a href="https://flylib.com/books/en/2.869.1.58/1/">https://flylib.com/books/en/2.869.1.58/1/</a> <a href="https://en.wikipedia.org/wiki/IBM_Information_Management_System">https://en.wikipedia.org/wiki/IBM_Information_Management_System</a>
	VSAM		Prohibited								<a href="https://en.wikipedia.org/wiki/Virtual_Storage_Access_Method">https://en.wikipedia.org/wiki/Virtual_Storage_Access_Method</a>	<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>
	ISAM		Prohibited								<a href="https://en.wikipedia.org/wiki/ISAM">https://en.wikipedia.org/wiki/ISAM</a>	<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>
	xBase		Prohibited								<a href="https://en.wikipedia.org/wiki/Microsoft_Data_Access_Components">https://en.wikipedia.org/wiki/Microsoft_Data_Access_Components</a>	<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>
	Paradox		Prohibited								<a href="https://en.wikipedia.org/wiki/Microsoft_Data_Access_Components">https://en.wikipedia.org/wiki/Microsoft_Data_Access_Components</a>	<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>
	Hierarchical Database Access	All	Prohibited								<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>	e.g. HSAM, HISAM, HIDAM, HDAM, PHDAM, SHSAM, SHISAM, etc.
	Network Database Access	All	Prohibited								<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>	e.g. BTAM, QTAM, TCAM, VTAM, etc.
	Non-security patch updated data access methods	All	Prohibited								<a href="https://en.wikipedia.org/wiki/Access_method">https://en.wikipedia.org/wiki/Access_method</a>	All data access methods and versions without security patching

The following represents COV **programming language classifications** per Enterprise Technical Architecture (ETA):

<https://www.vita.virginia.gov/media/vitavirginiagov/it-governance/ea/pdf/Legacy-IT-Solutions-Topic-Report.pdf>

Key	Description
	Emerging
	Projected
	Approved
	Divest: Plan
	Divest: Execute
	Prohibited
	Skipped