

AI Low Code Platform

A12 Release 2026.06
July 2026

Now
Open
Source

openCode 

GitHub 

Basic information about A12
<https://www.a12.ai>

What is A12?

A12 is an open source platform for developing enterprise applications in complex IT landscapes. It relies on model-driven software engineering (MDSE) and brings the low-code principle to the world of enterprise software. As an open platform, A12 simplifies the integration of best-of-breed solutions and the use of AI at all levels. A12's Modeling Environment provides tools to

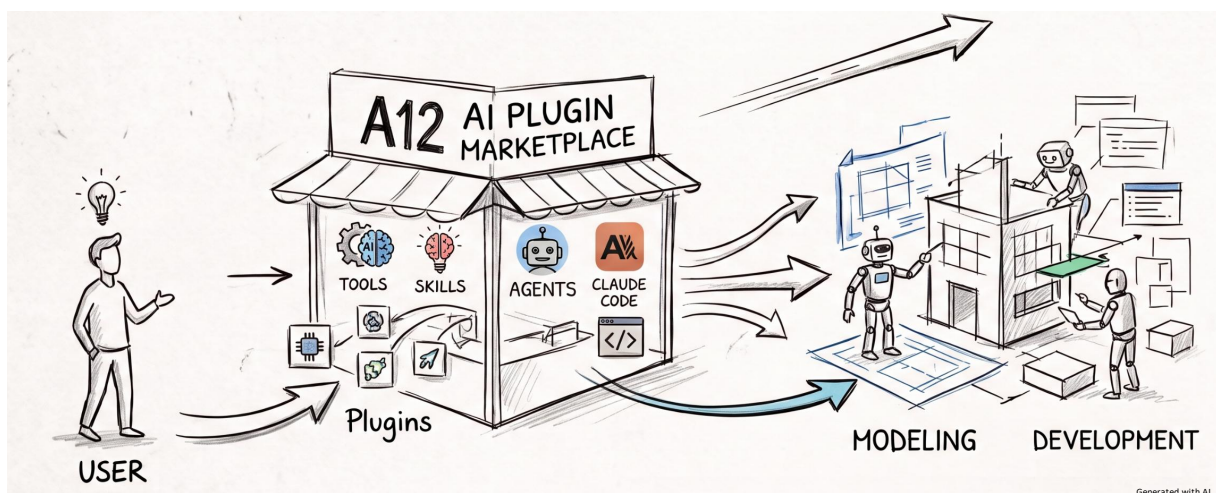
create and maintain parts of an application over the long term as independent business logic modules without programming experience. A12's Runtime Platform provides the flexibility needed to evolve business-critical applications with professional individual software development, AI support and system integration into fully integrated enterprise applications.

AI MARKETPLACE CLAUDE CODE

A12 AI Plugin Marketplace Unlocks A12's AI Synergies

With the A12 AI Plugin Marketplace, mgm provides all users with an Enterprise license access to a growing repertoire of tools, skills, and agents. These are designed to accelerate AI-assisted modeling and development with A12 and to make knowledge about A12 accessible to AI models in a structured way. Currently, in addition to an MCP server for A12 documentation, the offering includes plugins for Claude Code. These include, for

example, skills for initial project setup, migration planning, and end-to-end modeling support. Fundamentally, however, the A12 AI Plugin Marketplace is designed to be model-agnostic. The offering is continuously being developed and is intended to support sovereign alternatives in the future. mgm is collaborating here with the Fujitsu subsidiary Fsas Technologies on the European Sovereign Agentic Coding Appliance (ESACA) project.



Business Scripting Enables Non-Developers to Define Business Logic

Until now, the creation of an A12 application typically followed two clearly distinct threads: modeling and development. Business analysts modeled data models, workflows, and parts of the user interface. The development team was responsible for tasks such as technical integrations and the implementation of more complex business processes. With Business Scripting there is now a new middle ground: business analysts are empowered to define even more complex business processes on their own - using an intuitive scripting workflow in TypeScript. The experimental feature will initially be made available internally at mgm with the A12 release 2026.06. It is scheduled to be released as an experimental Enterprise component in October this year. The new *Business Scripting Platform* thus unlocks A12's feature set in a new way. It was designed from the ground

up for flexible extensions and AI support. The *Business Scripting API* offers simplified access to core platform functions and can be extended on a per-project basis via Custom API Extensions. The focus of the initial experimental support for Business Scripting is on a consistently convenient user experience - including a test framework for scripts, hot reload, error diagnosis, and debugging. AI agents can also use Business Scripting directly. In addition to an end-to-end workflow via the Claude plugin, an API summary in Markdown and descriptive error logs are available for the context. The scripts can already be integrated into workflows as executable tasks. In future releases, Business Scripting will also be supported in the frontend and expanded in the backend.

```

1  import { A12, DocumentReference, } from "@com.mgmt.a12.scripting/business-scripting
2
3  interface Inputs {
4      categoryId: string;
5      contractDocRef: DocumentReference;
6  }
7
8  interface Outputs {
9      categoryDocRef: DocumentReference;
10 }
11
12 async function execute(inputs: Inputs): Promise<Outputs> {
13
14     const templateDocRef = await A12.Documents.findSingle("CategoryTemplate_DM",
15         A12.Documents.Filter.field("/root/categoryId").equals(inputs.categoryId)
16     )
17
18     const copyDocRef = await A12.Documents.copy(templateDocRef)
19
20     await A12.Links.create({
21         relationshipModelName: "ContractCoverageCategory_RM",
22         sourceRole: "Contract",
23         sourceDocRef: inputs.contractDocRef,
24         targetRole: "CoverageCategory",
25         targetDocRef: copyDocRef,
26     })
27
28     A12.Log.info(`Successfully linked ${inputs.contractDocRef} to ${copyDocRef}`)
29
30     return { categoryDocRef: copyDocRef }
31 }

```

Consistent structure

Restricted TypeScript.
Define **Inputs**.
Define **Outputs**.
Define business logic in **execute**.

Consistent & convenient API

```

A12.Documents.create
A12.Documents.find
A12.Documents.writeFields
A12.Links.create
A12.Links.find
...

```

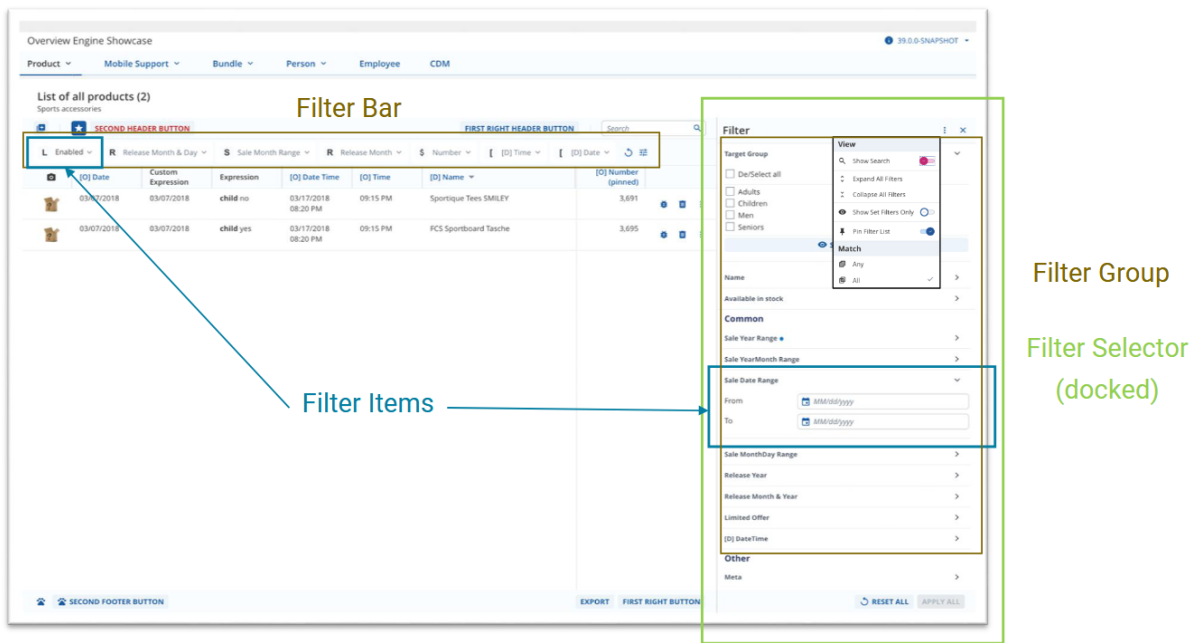
More News in a Nutshell

- Since May 27, 2026, the source code for A12 has been available on [openCode](#) and [GitHub](#).
- **Combination Modeling**, which was previously available as an experimental feature, is **now productive**. It offers three mechanisms for **cleanly encapsulated data modeling**: **addition**, **selection**, and **decoration**.
- The **architecture** of the **Relationship Engine** has been **revised** — including new modeling options. The original **"binding" approach** is still available, but **will be removed** from the platform in 2027.
- **Performance boost** for the **Form Model Editor** in SME: **Saving large models** is **more than ten times faster** than before. Loading times have also been drastically reduced.

Overview Engine Introduces New Filtering System

In tabular overviews, end users often work with large datasets and perform actions to narrow the view down to the entries they need. The more precisely users can filter the entries in an overview, the faster they can access the right data. The Overview Engine now supports this process even better with a new, flexibly configurable filter system. Modelers can select it via the “Custom Filter” mode in the Overview Model and then de-

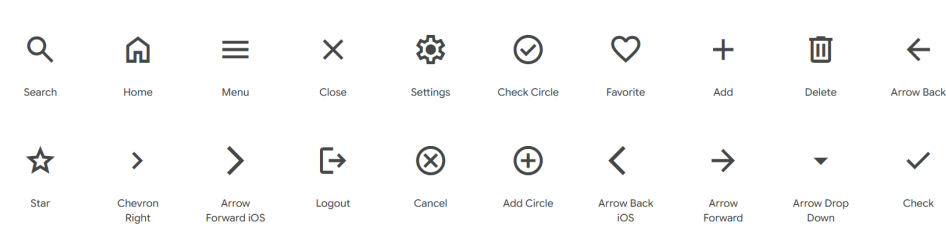
fine highly precise filter groups and items, which are displayed to users either in a filter bar for quick access and via a filter selector for additional filter options. Filter items can also be easily defined using a filter definition in our Query Language. This gives business analysts full control over filter functions without the need for custom development.



Icon Widget Supports Material Symbols

You don't have to reinvent the wheel every time. That's why, when it comes to the user interface, A12 has long relied on the Google standard Material Icons in the Icon Widget. With this new release, the latest version of the library—Material Sym-

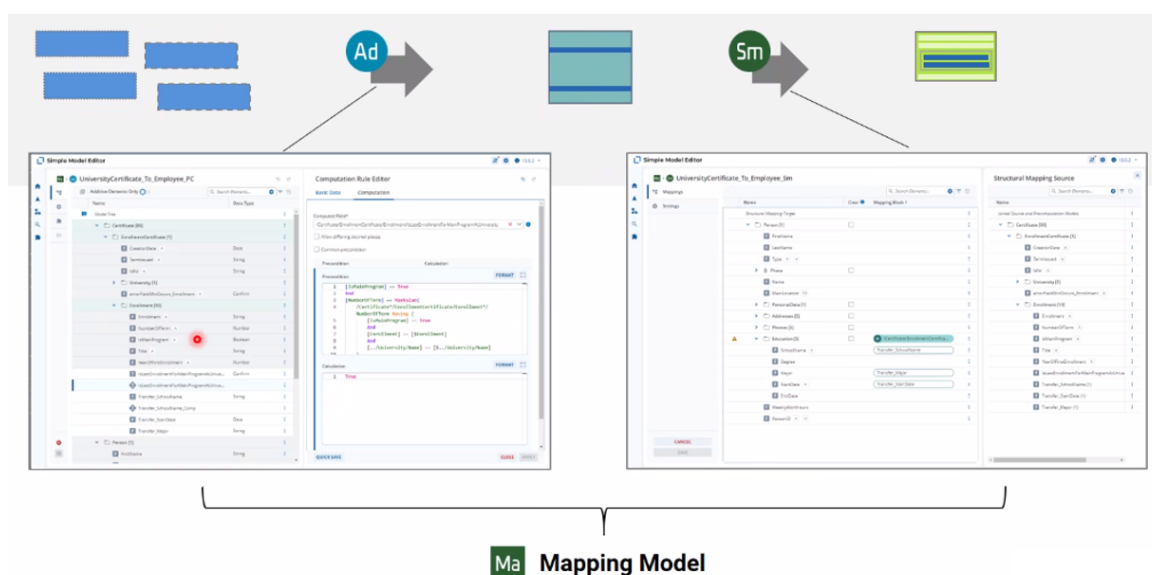
bols—is now supported. It not only offers more icons but also reduces loading times. The IconTheme type remains unchanged, and the Icon component's API is fully backward-compatible.



Compiling Data From Different Sources - Mapping Model Is Productive

Data mapping is traditionally a technical task. However, in many cases, it requires a deep understanding of the business domain. In A12, therefore, business experts and analysts can define—without programming—how data from various sources should be incorporated into target structures. This is made possible by the Mapping Model and the associated tools in the Simple Model Editor. As of Release 2026.06, this advanced form of data modeling has matured to the point where it has transitioned from experimental to production status. Mapping takes

place in two steps: First, the Precomputation Model is used to transform the data into the correct form according to business requirements—for example, by calculating target values from multiple source fields using the Kernel language. Then, using the Structural Mapping Model, modelers define how the preprocessed data should be inserted into the target structure. Detailed documentation, including extensive examples and “Mapping Recipes” for typical scenarios, is available at [GetA12.com](https://geta12.com).



Good to Know: Experimental vs. Productive

In the A12 development cycle, **new features** typically go through **two phases**:

- Initially, they are **experimental**. This means they are already part of the A12 platform but are still subject to more frequent changes. Breaking changes may occur at any time on the model and API sides. The editor and documentation may still be under development. Training and support are handled on a case-by-case basis. A12 projects can already use the feature under these conditions and provide feedback for further development.
- Once the feature is more mature, it becomes **productive**. Models and APIs are subject to the platform's breaking change management. Feedback and requests for improvements are submitted via the standard A12 channels. Editors and documentation are available in accordance with A12 standards. Standardized training and tutorials are provided. All A12 projects can use the feature with confidence.

PRINT MODEL SME

Print Model Gets Leaner and AI-Ready

In many print templates, images such as logos are important design elements. In the previous versions of A12's Print Model, these images were embedded directly. Now they are managed externally. During automatic migration in the SME, images are extracted from the Print Models and placed in the "Resources" directory. Outside the SME, a migration tool is available to update the models. In addition, the "Print Setting Model" becomes

obsolete, as the fonts in use are now managed—analogueous to the images—in the SME's workspace resources. Modelers benefit directly from the leaner Print Model through unified mechanisms for handling resources and settings. At the same time, these changes make it easier for AI models to process the print models, thereby supporting more advanced agentic modeling.

PERFORMANCE HAZELCAST INFINISPAN

Infinispan Instead of Hazelcast: Data Services Replaces Caching Solution

When data is accessed frequently, it is stored in RAM cache to improve performance. Technically, this task is handled by a cache provider. Until now, the Data Services component of A12 has used Hazelcast. However, due to its hybrid licensing model, Hazelcast conflicts with A12's open-source standards – particularly when integrating with the Java framework Spring. Furthermore, Hazelcast's feature set far exceeds what is necessary, resulting in complex configurations. Data Services is therefore replacing Hazelcast with the leaner Infinispan solution, which is fully licensed under Apache 2.0 and now requires only a single configuration file. From a project perspective, it is possible to continue using Hazelcast with minimal effort. However, this is not recommended, as all tests are now geared toward the new default solution with Infinispan.



PLASMA FIGMA

A12 Plasma on Figma: Open-Source Design Kit for Enterprise Applications

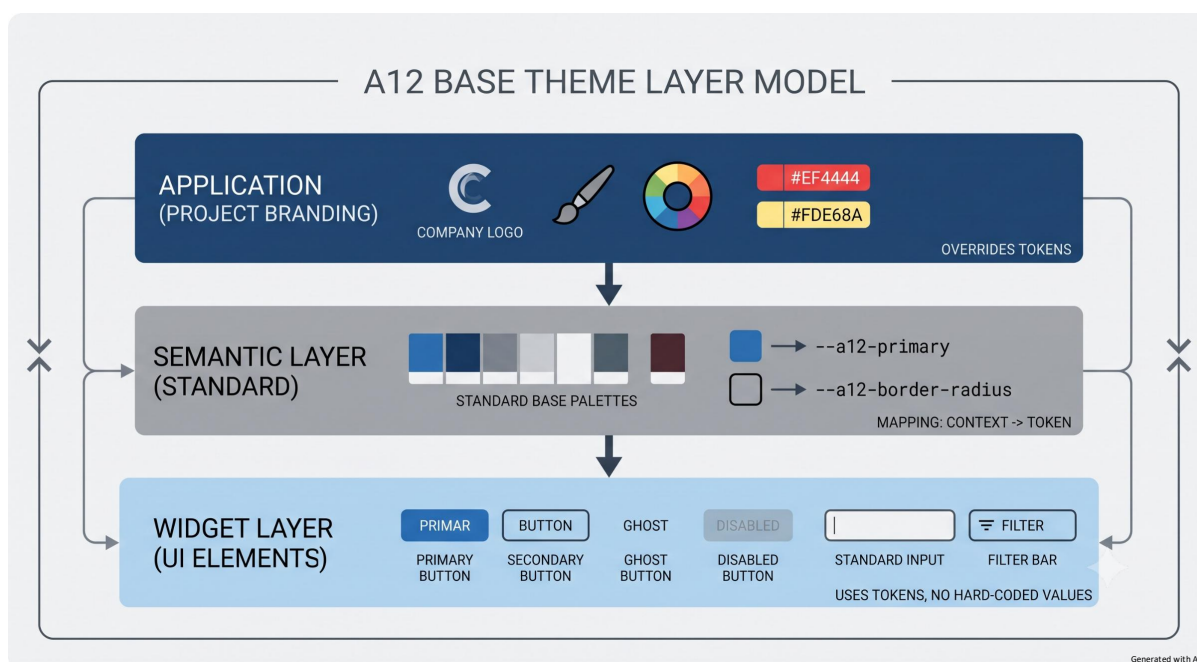
Since July 17, 2026, the [A12 Plasma Design Kit in the Figma Community](#) has been available - free of charge and without registration, licensed under CC BY 4.0. The Design Kit contains a curated selection of components for the majority of typical design tasks in enterprise applications. Initially, it includes a com-

ponent library for the most common UI building blocks in enterprise applications - such as form elements, data tables, navigation structures, and notification widgets - as well as a complete token system for consistent theming. The scope is set to be continuously expanded.

“One Theme to Rule Them All” – A12 Gets a New Base Theme

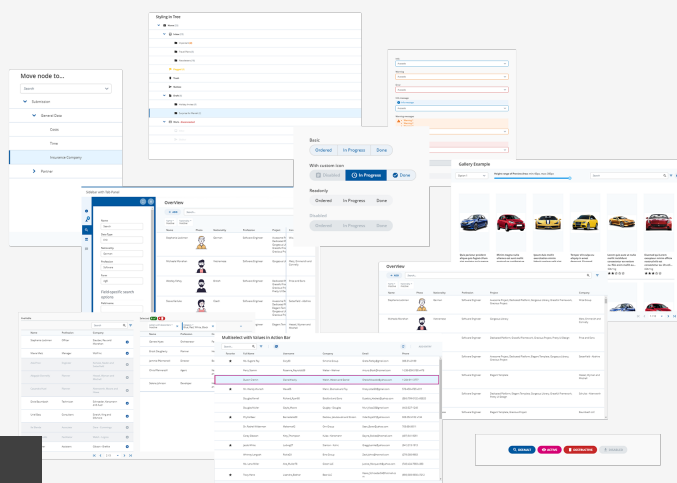
To make it easier for people and AI agents to create themes for A12 applications, the platform is introducing a new base theme. It is based on a three-tiered layer model, consisting—from top to bottom—of the Application Layer, Semantic Layer, and Widget Layer. Theming variables in lower layers inherit definitions from higher layers. For example, a color adjustment at the application level can trigger corresponding color adjustments in the applica-

tion's widgets. Custom color schemes can be implemented very easily using the new QuickThemePalette. The new base theme is visually modeled after the previous Flat theme and adopts the basic spacing definition of the Flat Compact theme (12px). All four previous standard themes (Default, Flat, Compact, Flat Compact) have been deprecated and will be removed in upcoming releases.



A12 Widgets

A12 Widgets is a comprehensive collection of prebuilt components designed to provide a pleasurable and accessible user experience.



A12-Widget-Showcase

<https://www.mgm-tp.com/a12.htmlshowcase/>