Fabasoft®

Software Product Information

Fabasoft app.ducx 2026 February Release

Valid from February 1, 2026

No rights to our software or our professional services, or results of our professional services, or other protected rights can be based on the handing over and presentation of these documents.

Contents

1 Introduction 3

2 Structure of Software Product Information 3

3 Fabasoft app.ducx 3

3.1 Benefits of Fabasoft app.ducx 3

3.1.1 Shorter Time to Value 3

3.1.2 Integrated in the Development Environment 3

3.1.3 Localization and Customization 3

3.1.4 Integrated Compiler 4

3.1.5 Loader 4

3.1.6 Automatic Address Management 4

3.1.7 “Convention Over Configuration” Principle 4

3.2 Domain-Specific Languages 4

3.2.1 DUCX Object Model Language 4

3.2.2 DUCX Resource Language 5

3.2.3 DUCX User Interface Language 5

3.2.4 DUCX Business Process Language 5

3.2.5 DUCX Use Case Language 5

3.2.6 DUCX Organizational Structure Language 5

3.2.7 DUCX Customization Language 5

3.2.8 DUCX Unit Test Language 6

3.3 General Use of Fabasoft app.ducx 6

3.3.1 Installation 6

3.3.2 Working With app.ducx Projects 6

3.3.3 Productivity Features 6

4 Conditions of Use 7

5 Supported Platforms and Minimum System Requirements 7

5.1 Microsoft Windows Environment 7

5.2 Additional Conditions 8

5.3 Supported Editions of Third-Party Products 8

6 Open Source Licenses 9

1 Introduction

Fabasoft app.ducx is a powerful use case-oriented application development system for developing composite content applications (CCAs). It is comprised of several declarative modeling languages that should be familiar to C++, C# and Java programmers. Fabasoft app.ducx combines the productivity gains of rapid application development languages and the power of Fabasoft Folio technology.

2 Structure of Software Product Information

The software product information for Fabasoft app.ducx defines the scope of services of the standard product Fabasoft app.ducx.

The chapter “Fabasoft app.ducx” describes the basic functionality of Fabasoft app.ducx.

The last chapters of the software product information describe conditions of use, supported platforms and minimum system requirements.

3 Fabasoft app.ducx

Fabasoft app.ducx provides powerful domain-specific languages supporting the rapid implementation of CCAs.

3.1 Benefits of Fabasoft app.ducx

Fabasoft app.ducx relies on cutting-edge technology to facilitate rapid application development. This chapter gives a concise overview of the main benefits of Fabasoft app.ducx.

3.1.1 Shorter Time to Value

The use case oriented approach of developing CCAs with Fabasoft app.ducx focuses on the perspective of the end user. Powerful domain-specific languages reduce the complexity of projects leading to less ambiguity and fewer errors between specification and implementation. As a result, maintenance and development costs can be reduced while at the same time increasing end user satisfaction.

3.1.2 Integrated in the Development Environment

In keeping with Fabasoft’s commitment to provide platform independence, Fabasoft app.ducx is a development platform for both the Microsoft Windows and Linux worlds and is therefore integrated into Eclipse. This means greater flexibility for your organization while developing solutions.

3.1.3 Localization and Customization

Fabasoft app.ducx allows for an easy and effortless localization and customization of your CCAs.

3.1.4 Integrated Compiler

Fabasoft app.ducx allows you to benefit from the rich metadata, compile-time syntax checking, static typing and auto-completion previously available only to imperative programming languages.

Using the compiler provided with Fabasoft app.ducx, a DUCX project can be compiled by simply triggering the build process of the respective development environment, which produces a Fabasoft Folio container file (with a .coo extension) and – if the project also contains use case implementations in Java – a Java archive (.jar) file. These files are then automatically packaged and loaded into the Fabasoft Folio domain.

3.1.5 Loader

The app.ducx compiler produces a Fabasoft Folio container file, which is uploaded to a web service connected to your Fabasoft DUCX domain.

3.1.6 Automatic Address Management

With Fabasoft app.ducx, you do not need to worry about address management anymore. Fabasoft app.ducx automatically assigns addresses to all component objects created and keeps track of address assignments.

3.1.7 "Convention Over Configuration" Principle

Fabasoft app.ducx uses a set of coding and naming conventions that when adhered to eliminate the need for configuring every single aspect of your software solution. As a result, you need to write less code which will make software development and maintenance much easier.

3.2 Domain-Specific Languages

Developing use case-oriented software solutions requires managing different aspects and elements such as data structures, user interface design, the implementation of methods and business rules.

In order to account for this concept in an optimal manner, Fabasoft app.ducx is comprised of several declarative modeling languages, each designed for covering a particular aspect of solution development. For example, Fabasoft app.ducx includes a modeling language that has been designed explicitly for the definition of an object model. In addition to this, Fabasoft app.ducx includes languages for defining resources, a user interface model, an implementation model, a process model, and an organizational structure model.

These modeling languages are referred to as domain-specific languages (DSLs), where each DSL was designed for addressing a certain aspect of use case-oriented software development.

3.2.1 DUCX Object Model Language

The purpose of the DUCX object model language is to define the persistent object model for a software component.

Using the DUCX object model language, you can easily define the basic elements that make up the object model:

- object classes
- properties and fields
- enumeration types
- compound types
- extensions of existing object classes and types

3.2.2 DUCX Resource Language

The DUCX resource language allows you to define resources such as string objects, error messages and icons. Using the DUCX resource language, you can create culture- and language-independent solutions as it allows you to avoid hard-coded, culture- and language-specific values in your solution.

3.2.3 DUCX User Interface Language

The DUCX user interface language allows you to define forms, form pages, menu items and other user interface elements for your object classes.

3.2.4 DUCX Business Process Language

The DUCX business process language allows you to define the process model for your software component. The process model is comprised of one or more template processes defining an ordered list of sequential or parallel activities and process control elements such as conditions or loops. Each activity definition consists of a list of steps that need to be carried out by a user responsible for completing a particular activity.

3.2.5 DUCX Use Case Language

The purpose of the DUCX use case language is to define and implement use cases. With the DUCX use case language, you can define new use cases, and provide method implementations for these use cases that are written in Java. Alternatively, you can also use DUCX expression language or virtual applications to implement use cases.

3.2.6 DUCX Organizational Structure Language

The abstract elements of the organizational structure required for your solution are defined using the DUCX organizational structure language.

3.2.7 DUCX Customization Language

The purpose of the DUCX customization language is to define, customize and tailor your software component to project- or solution-specific requirements.

3.2.8 DUCX Unit Test Language

Unit testing is an important part of the development process. By using the DUCX unit test language it is possible to create unit tests and bundle them within unit test groups. With scenarios you can establish a stable test environment.

3.3 General Use of Fabasoft app.ducx

Fabasoft app.ducx is seamlessly integrated in Eclipse, a popular and powerful development environment for Microsoft Windows and Linux.

The software development life cycle is controlled from within your development environment.

3.3.1 Installation

Fabasoft app.ducx must be installed in the Fabasoft Folio domain that is used for development.

The Fabasoft Folio domain can also be installed on a remote computer. Only the app.ducx Eclipse plug-in has to be installed on the developer’s computer. The app.ducx Eclipse plug-in supports basic authentication. All information required by Fabasoft app.ducx is retrieved through its service-oriented architecture. Thus, it is still possible to continue development if the connection to the development web server becomes unavailable.

3.3.2 Working With app.ducx Projects

The main use cases for working with app.ducx projects in Eclipse are:

- Create a new project.
- Compile and run projects. The project can be immediately tested in the automatically started Fabasoft Folio Web Client.
- Add software component references to be able to use short references.
- Add additional source files or resources like icons.
- Manage address ranges and work together using a third-party version control system.
- Export an app.ducx project and use it in a productive environment.

3.3.3 Productivity Features

Following features provide effective development with Fabasoft app.ducx:

- app.ducx Eclipse perspective and outline view
- Syntax highlighting
- Syntax check
- IntelliSense
- Quick fixes
- Folding of source code blocks
- Go to definition
- Code snippets

- Generated comment blocks e.g. for use cases
- Generated Java templates for use case implementations

4 Conditions of Use

- All statements contained in this software product information concerning the intended use and conditions of operation of the program are understood as information only on principle. To ensure and guarantee the intended use and conditions of operation in everyday use of the program, expert training by specially qualified staff is recommended in any case.
- Performance and reaction time of Fabasoft software products significantly depend on the used infrastructure. For layout, configuration and sizing of the infrastructure we refer to the information, recommendations and specifications of the particular manufacturers.
- Without a suitable infrastructure for online backups a consistent backup of Fabasoft Folio resources is only possible within dedicated backup time slots (planned downtime). In both cases functionality for backup, restore and disaster recovery has to be implemented according to a separately defined concept.
- This program is intended for customary commercial use. This does not include usage that demands special requirements (like controlling of vehicles, machines and facilities, real-time applications).

5 Supported Platforms and Minimum System Requirements

Backward compatibility: Fabasoft app.ducx 2026 can be used with limits with Fabasoft installations from version 2025.

5.1 Microsoft Windows Environment

Development Environment for Java
Microsoft Windows 11 Enterprise (x64)
Eclipse Temurin OpenJDK 21.0.9 (JDK, HotSpot) (x64)
Eclipse 2025-09 (x64) (incl. Eclipse Temurin OpenJDK 21.0.9)

BIRT Integration
Eclipse BIRT 4.21

Build with Apache Ant
Apache Ant 1.10.14

5.2 Additional Conditions

- The minimum hardware and software requirements for any future versions of this product may differ from the requirements of this version.
- A certain version of a third-party product integrated into any Fabasoft software product is supported by Fabasoft as long as the software product itself is supported by the third-party itself. For instance, a particular version of an integrated Microsoft software product is only supported by Fabasoft until the date on which the Microsoft “Extended Support Phase” for this product version ends.
- Third-party products that are required or supported by Fabasoft products, proper licensing and installation of them, necessary tests for product release as well as corresponding manufacturer support are not included in the scope of supply and services and therefore, they are not subject to warranty of Fabasoft regarding functionality, mode of operation or features. Fabasoft is not responsible for errors or malfunction, which are caused by third-party software products and/or products that are not included in the scope of supply and services.
- Fabasoft software products are only supported and tested on the mentioned reference system environments. In addition to the third-party product editions listed in the reference system environments the editions listed in chapter “Supported Editions of Third-Party Products” are supported but not tested.

5.3 Supported Editions of Third-Party Products

Following editions of third-party products are supported. Explicitly tested are only those editions that are listed in the reference system environments.

- The third-party product Microsoft Windows Server 2019 is supported in the following editions: Essentials and Standard. The Datacenter edition is only supported in projects (the following values for CPU, available RAM and free disk space do not apply for Datacenter Edition).
- The third-party product Microsoft Windows Server 2022 is supported in the following editions: Essentials and Standard. The Datacenter edition is only supported in projects (the following values for CPU, available RAM and free disk space do not apply for Datacenter Edition).
- The third-party product Microsoft Windows 10 is supported in the following editions: Pro and Enterprise.
- The third-party product Microsoft Windows 11 is supported in the following editions: Pro and Enterprise.

6 Open Source Licenses

The open source software contained in Fabasoft app.ducx is licensed under conditions that require to display the following notes.

- ANTLR (https://www.antlr2.org/)
- Apache Axiom (https://ws.apache.org/axiom/)
- Apache HttpClient (https://hc.apache.org/httpcomponents-client-ga/index.html)
- Apache XMLBeans (https://xmlbeans.apache.org/)
- Apache XmlSchema (http://ws.apache.org/xmlschema/)
- Eclipse (https://www.eclipse.org/)
- FasterXML jackson-core (https://github.com/FasterXML/jackson-core)
- Guice (https://github.com/google/guice)
- JUnit (https://junit.org/junit4/)
- mod4j (https://github.com/codehaus/mod4j)
- Xtext (https://www.eclipse.org/Xtext/)

The corresponding copyright annotations and terms of license can be found here: Open Source Licenses