software AG

NaturalONE

Introduction

Version 8.3.7

March 2016

ADABAS & NATURAL

This document applies to NaturalONE Version 8.3.7.

Specifications contained herein are subject to change and these changes will be reported in subsequent release notes or new editions.

Document ID: ONE-INTRODUCTION-837-20160330

Table of Contents

Preface

This introduction is organized under the following headings:

What is NaturalONE?	A brief overview of the different components of NaturalONE.
Different Modes for Developing Natural Applications	Information on the different modes for developing Natural applications using NaturalONE (local mode and Natural server mode).
Using a Version Control System	Information on the supported version control systems.

1 What is NaturalONE?

General Information

NaturalONE is part of the Software AG Designer. It is an Eclipse-based development environment for developing and maintaining Natural applications with web-based user interfaces and Natural services. NaturalONE combines the functionality of several tools into a single development framework, and covers functionality across the entire product development lifecycle, including application development, testing, automated documentation of the sources in Predict, versioning, and deployment of the application into the production environment.

NaturalONE uses the standard Eclipse functionality and adds its own perspective, views, editors and commands to the Eclipse workbench. NaturalONE is integrated seamlessly into the Eclipse workbench. This means that you do not see where the standard Eclipse workbench ends and where the NaturalONE-specific user interface starts.

NaturalONE addresses developers who are used to work natively on mainframe, UNIX, Linux, OpenVMS or Windows platforms. Owing to the graphical user interface and many features for fast Natural source development (such as code assist), developers gain productivity using this environment. Developers who are already familiar with Eclipse can comfortably edit Natural sources in an environment to which they are already used; the environment always behaves in the same way, no matter whether Natural or Java applications are developed.

If you are not yet familiar with Eclipse, see the Eclipse online help at ***http://www.eclipse.org/documentation/***, or start the Software AG Designer and then choose **Help > Help Contents**. General information on Eclipse can then be found under *Workbench User Guide*. When working with the Software AG Designer, the online help also provides help for the currently installed Software AG products; this can be found under *Software AG Designer Guides*.

Basic Functionality for Natural Application Development

The basic functionality of NaturalONE is described in *Using NaturalONE*.

In the Eclipse workspace, all files are organized in projects. A direct connection to a Natural server is not required for editing the Natural sources. However, for executing and debugging a Natural application, NaturalONE establishes a connection to the appropriate Natural runtime.

The following topics are covered below:

- Powerful Tool Set
- Powerful Natural Environment

- Direct Development on a Natural Server

Powerful Tool Set

Applications can be developed or maintained in the Eclipse environment.

Editors

To support developers and gain productivity during the development life-cycle, specific Natural editors are available:

- **Source Editor**
  The source editor uses a real incremental Natural parser. It supports the developer with syntax coloring of the Natural source and content assist for fast code writing. Both Natural programming modes (reporting mode and structured mode) are supported. For the development of international applications, bidirectional language support is enabled.

  NaturalONE makes use of the Natural source editor for data area editing. Data areas are defined with the `DEFINE DATA` statement. With the assistance of the Natural parser, syntactically correct data areas can be developed quickly and easily.
- **Map Editor**
  With the map editor, Natural maps can be defined graphically. This also includes bidirectional language support. It is possible to navigate through the parts of a map on a graphical basis using the **Outline** view. Inline rules or Predict rules can also be edited.
- **DDM Editor**
  DDMs which have been generated by Predict can be adopted for your application. It is also possible to create DDMs from scratch. This can be done for all types supported by Natural: Adabas, SQL, Tamino, VSAM and more. In the **Outline** view, the DDM structure is visualized in a hierarchical manner.

Debugging and Execution

Debugging and execution of applications is possible. Watchpoints, breakpoints, etc. make bug fixing less complex.

XML

With the XML toolkit, it is possible to generate functionality for the processing of XML documents. DTDs or schemas can be used for generating Natural data areas, parser implementations and serializers for XML documents, and vice versa.

Database Retrieval

The data browser provides fast access to Adabas or SQL databases. With just a few mouse clicks, it is possible to write a database retrieval and to display the retrieved data in the **Report Data** view. It is easy to check the content of the database and to test whether the application works correctly.

Powerful Natural Environment

Offloading the application into the Eclipse workspace gives you several advantages. The Natural builder keeps the dependencies of the application. This is visualized in the **Dependencies** view. The builder keeps track of the source changes. Owing to the label decorations, you can easily see which Natural objects have not yet been saved locally, not yet been compiled on the Natural server or not yet been versioned in your version control system. When the preferences are set properly, the builder recatalogs the appropriate objects on the Natural server using the Natural parameters that are defined in the Eclipse environment.

Basically, the builder supports two different workspace structures: one which is very Natural-related (that is, the application is based on the conventional Natural library structure), and another which allows you to define folders which follow the naming conventions of the underlying file system (Windows or Linux). With the latter workspace structure, the folders are mapped to "real" Natural libraries. Thus, you can structure your Natural applications in the workspace in a more logical way than with the conventional library structure. Folders can be nested.

Besides using folders in the workspace, it is also possible to use alternative (long) file names for the objects. These file names also have to follow the rules of the underlying file system. There is a mapping between the alternative file name and the Natural object name, where the Natural object name always follows the Natural naming conventions.

Where useful, wizards (for example, for creating new projects or new objects) are used.

Error messages are edited with the error message editor. They are also stored in the Eclipse workspace.

To gain productivity, features which support often repeated actions are available. For example, you can externalize code fragments into separate Natural objects. Or you can rename Natural objects using the refactoring feature. When you create a Natural source from scratch, a skeleton according to the Natural object type is automatically generated.

You can customize the NaturalONE environment to your specific needs. In the Natural preferences, you can change the behavior of the Natural builder, you can define the platform for which the Natural parser is to check the Natural syntax, or you can create your own code templates.

You can version your applications in a version control system. A Natural application consists of the Natural objects, configuration parameters, error messages, etc. A version control system enables you to merge the application pieces from several developers. After merging the sources in the repository of the version control system, the application is ready for deployment onto the designated platform. The deployment wizard ensures that the appropriate parts of the application are deployed onto the Natural server. See also ***Using a Version Control System***.

Out of the box, it is possible, for example, to execute, debug or test Natural applications or parts of them. The local Natural runtime is already configured for usage with NaturalONE. By default, no additional configuration is required. With the local Natural runtime, an EntireX broker environment is available and an RPC server is started. This allows you to develop RPC-based applications or to make use of web services.

Direct Development on a Natural Server

If you are used to work with Natural Studio and Natural's Single Point of Development (SPoD) concept, you will find comparable functionality in NaturalONE. The **Natural Server** view makes visible the information from the Natural servers which may be located on different platforms (on a mainframe, UNIX, Linux, OpenVMS or Windows platform). Using this view, you can edit your Natural sources directly on a Natural server, and catalog or even execute them directly on the server. Information about the server configuration is available; this is similar to the output of the Natural system commands `SYSPROD`, `SYSPROF`, `SYSFILE` and `UNLOCK`.

Natural for Ajax / Ajax Developer

Besides the basic functionality for Natural application development, NaturalONE enables you to create rich internet applications which use the Ajax (Asynchronous JavaScript and XML) technology. This feature, the Ajax Developer, is always installed together with the basic functionality, thus providing an integrated development and runtime environment for Natural for Ajax applications.

Ajax Developer includes a number of tools for creating and maintaining complex graphical user interfaces. Its central tool is the Layout Painter which is used to define layouts for HTML pages.

For detailed information, see *Natural for Ajax* and *Ajax Developer*.

Optional Components

In addition to the above-mentioned application development functionality which is always installed with NaturalONE, you can also install optional components for NaturalONE. Some of these optional components require that additional software is installed on a server.

The headings below correspond to the names that are used in the product selection tree of the Software AG Installer. See also *Installing NaturalONE*.

- Application Testing
- Mainframe Tools
- Natural Construct
- Natural Engineer
- Predict

- Service Development

Application Testing

The Application Testing component is used to directly test or create unit tests for various Natural objects, including subprograms, maps, subroutines and business services. It requires that EntireX is installed.

When you select **Application Testing** in the installer, the **Testing** context menu will be available in the **Navigator** view. For detailed information, see the *Application Testing* documentation.

Mainframe Tools

The Mainframe Tools component includes Mainframe Navigation which allows you to access and manipulate objects stored on a mainframe from Eclipse. These objects include datasets and members, as well as system objects such as active jobs or the console under the z/OS operating system. With Mainframe Navigation, the objects are displayed in a tree structure and can be browsed and edited in Eclipse. On the mainframe server, Mainframe Navigation is supported by Natural ISPF.

Mainframe Navigation requires that additional software is installed on a server. See *Installation and Configuration* in the *Mainframe Navigation* documentation.

When you select **Mainframe Tools** in the installer, the **Mainframe Navigation** view will be available. For detailed information, see the *Mainframe Navigation* documentation.

Natural Construct

The Natural Construct component allows you to use your existing Natural Construct models in NaturalONE, and to create new Natural Construct models or code frames. It requires that additional software is installed on a server. See *Requirements* in the *Using Natural Construct* section of the *Code Generation* documentation.

When you select **Natural Construct** in the installer, the **Code Generation > New Using Construct Model** context menu will be available in the **Navigator** view. For detailed information, see *Using Natural Construct* in the *Code Generation* documentation.

Natural Engineer

The Natural Engineer component is used to convert Natural objects containing `INPUT` statements for use with Natural for Ajax pages. It requires that additional software is installed on a server. See *Setting Up a Natural Engineer Environment* in the *Natural Engineer* documentation.

When you select **Natural Engineer** in the installer, the **Natural Engineer** context menu will be available in the **Natural Server** view. For detailed information, see the *Natural Engineer* documentation.

Predict

The Predict component is used to document Natural sources in Predict via NaturalONE, manipulate and retrieve data stored by Predict on the server, and manage external objects (generation and administration). It requires that additional software is installed on a server. See *Setting Up a Predict Environment* in the *Predict Description and Generation* documentation.

When you select **Predict** in the installer, the **Predict Description and Generation** context menu will be available in the **Natural Server** view. For detailed information, see the *Predict Description and Generation* documentation.

Service Development

When you select **Service Development** in the installer, the following context menus will be available in the **Navigator** view:

- **Business Services**
  Used to create and maintain business services. For detailed information, see the *Business Services* documentation.

  This optional component requires that EntireX is installed. It also requires that additional software is installed on a server. See *Prerequisites* in the *Business Services* documentation.
- **Code Generation**
  Used to generate Natural subprograms and data areas. For detailed information, see the *Code Generation* documentation.

2 Different Modes for Developing Natural Applications

General Information

NaturalONE has two modes for developing Natural applications. One, the Natural server mode, is similar to Natural Studio when used as a development client in a SPoD environment. The other, the so-called local mode, is the preferred development mode in NaturalONE; this mode allows you to work in a way that an Eclipse user expects.

When you are used to work natively on a Natural server or with Natural Studio in a SPoD environment, you face a “paradigm shift” when switching to local mode. In local mode, the sources are no longer stored or modified directly on the Natural server. The central place for keeping the sources is now the Eclipse workspace which is connected to a version control system.

More information on these development modes is provided in the topics below.

Local Mode

This is the preferred way when working with Eclipse.

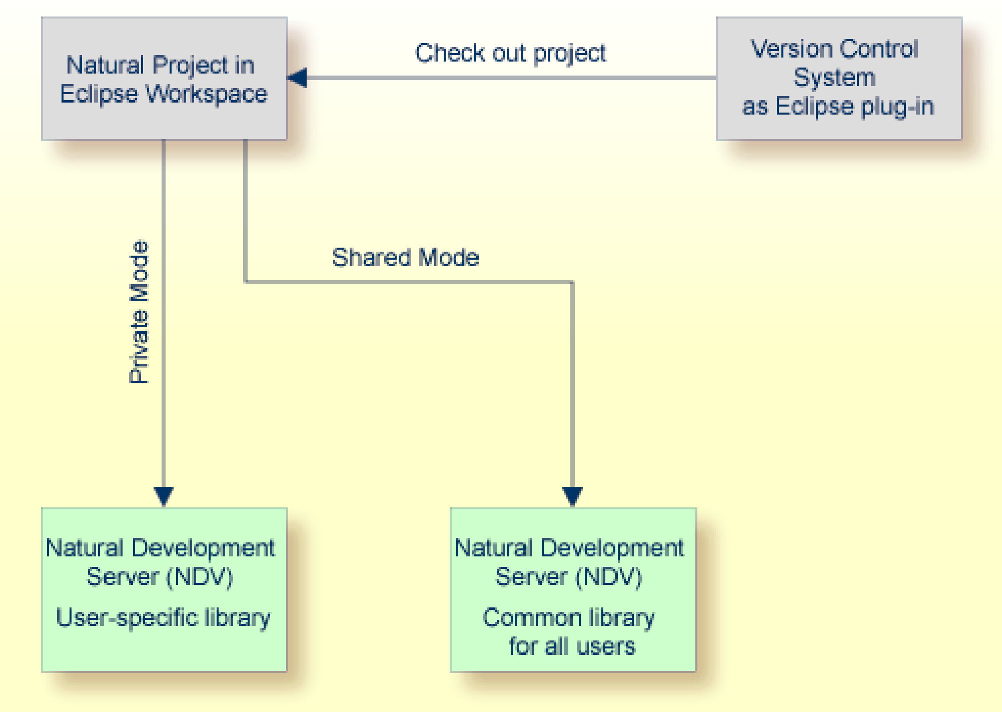
As a rule, you download a library from a Natural server into a Natural project in the local Eclipse workspace and then make your changes to the source code. In order to build the project, you have to update the appropriate Natural server, that is, you upload the changes to the server and catalog them there.

When several developers are working on the same Natural application, the Natural application is now spread over several Eclipse workspaces on different PCs. Two different modes in which the build is to be performed can be defined in the project properties:

- **Shared Mode**
  The build is performed in a common library on the Natural server that is shared by all users. This is the default mode.
- **Private Mode**
  The build is performed in a user-specific private library on the Natural server. This avoids a situation where one developer overwrites the changes of another developer.

For detailed information on these modes, see *Steplibs* in *Changing the Project Properties*.

Since the Eclipse features or plug-ins (for example, a plug-in for a version control system) expect the sources to be stored in the Eclipse workspace, you gain additional productivity in local mode because you can make use of third-party tools from other vendors.


Natural Project in Eclipse Workspace
Check out project
Version Control System as Eclipse plug-in
Private Mode
Shared Mode
Natural Development Server (NDV)
User-specific library
Natural Development Server (NDV)
Common library for all users

NaturalONE provides team support so that several developers can work in parallel with one application at the same time. Therefore, the configuration files are kept with the project and can thus be versioned in the version control system. Compiler options are always stored with a project.

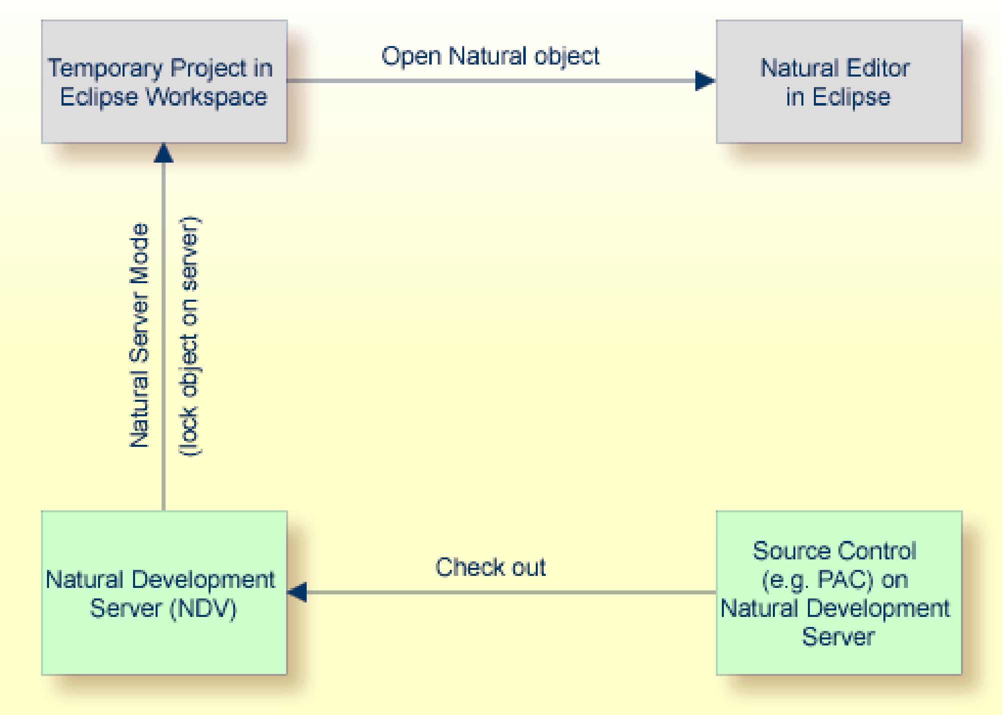
With NaturalONE, the versioned source is always considered to be the original source. You have to make sure that you do not copy a source to the Natural server and then manipulate the source on the server. If this happens, team support will fail. See also ***Using a Version Control System***.

Natural Server Mode

This is the classical way when working with Natural Studio. This mode mimics the SPoD behavior known from Natural Studio.

In this case, you do not have to download libraries to the Eclipse workspace. Instead, you work virtually directly with the objects on a Natural server. “Virtually”, because the objects are temporarily downloaded to the Eclipse workspace for editing purposes. As long as you edit a source, it is locked on the Natural server in order to prevent concurrent modifications by different users.

Since the Natural sources remain on the Natural server, versioning is also done on the Natural server.


Temporary Project in Eclipse Workspace
Open Natural object
Natural Editor in Eclipse
Natural Server Mode
(lock object on server)
Natural Development Server (NDV)
Check out
Source Control (e.g. PAC) on Natural Development Server

Note: In Natural server mode, there is no distinction between private mode and shared mode.

3 Using a Version Control System

With NaturalONE, you have the possibility to version Natural applications in the repository of a version control system. For this purpose, the following tools are supported:

- **CVS**
  CVS is shipped with Eclipse. The *Workbench User Guide* in the Eclipse help (***http://www.eclipse.org/documentation/***) provides detailed information on how to set up a CVS repository and on team programming with CVS.
- **Subversion (SVN)**
  If you want to use Subversion, you have to install an Eclipse plug-in such as Subclipse. Subclipse can be downloaded from ***http://subclipse.tigris.org/***. Make sure to download a plug-in version which supports your current Eclipse version. Subversion is considered to be the successor of CVS.
- **GIT**
  GIT is a distributed version control system that is described in detail on *https://git-scm.com/*. An Eclipse plug-in version of GIT called egit usually is already included in the standard Eclipse distributions.

CVS and SVN are client/server applications. Both require an Eclipse plug-in as the user interface and a server part (the versioning repository) which keeps the application.

GIT is a distributed (or decentralized) versioning tool and as such any GIT working directory is a fully-fledged repository. Instead of a single central server repository (as with CVS or SVN) there are many distributed repositories on different machines.

All plug-ins have their own perspectives. Each of these perspectives has to be enabled using the **Open Perspective** command.

For more information, see the documentation of your preferred version control system.