Get ready to meet Magik on Java™
with the release of MDT PE 3.6

With the release of MDT 3.6 and MDT 3.7 release we are focusing on the development of Magik on Java™ plug-in to meet the challenges of upcoming Magik on Java™, which is part of Smallworld™ 5 technology. Nevertheless we have added some new features and improved some of the existing ones.

New Features

- **Magik on Java™ plug-in** – as the Magik on Java™, the leading edge of Smallworld™ platform is approaching, the plug-in has been equipped with a new Magik debugger. The debugger comes with features known from the Smallworld™ 4.x debugger. Moreover, it has some new ones, thanks to Java™ technology. In future releases we will focus on bringing more functionalities to Magik on Java™ plug-in and support to any new Smallworld™ 5 features made available by GE.

- **Edit Actions** – as developers often need to copy, past or simply delete code we have decided to add support to these actions in Product Explorer and Outline context menus for Magik file elements. The behaviour of these actions differ based on the element the user is dealing with. These actions are dedicated to Magik source files, message files and load lists. This greatly improves developer performance especially while refactoring existing code.

- **Support for #DEBUG statement** – as we know many a developers use code instructions preceded by #DEBUG statement, which are meant to be run only when debugging. In the version MDT PE 3.6 on, such a solution is supported when running a session in a debug mode.

- **Support for load lists** – to ease load list editing we have added a load list structure, visible in Product Explorer and Outline view. By dragging load list elements, one can change the order of loading files or move them between load lists.

- **Bug Report action** – this feature allows you to quickly send information about problems you are experiencing while using MDT – if any. You can also use it to request a new feature or simply send us your feedback about MDT.

- **New icons** – there are also looking GIS session and runtime icons as well as updated wizards, because the visual effect also matters to us.

Dear Readers,

We are very happy to announce the seventh issue of our MDT Magazine. For the past six years MDT Magazine has now been published and yet there is no end in sight.

If we look back to how it all started with the very first version of MDT, the changes, the development and the improvements have been impressive. Only last year did we celebrate our fifth anniversary and we are able to present to you Magik on Java™ for the very first time.

With all the changes in the Smallworld™ environment and the ongoing rework and improvement of MDT, our dear colleague Błażej Sytar has already revealed the fact that in his opinion MDT is the only modern IDE for Magik development, as it also includes Java development techniques. In retrospect, when he made this statement, it could have seemed somehow bold to say so, nevertheless we could not agree more with this at the present time.

Of course is Magik on Java™ still one of the most highlighted and important topics, only time will tell. We are also proud to publish more interesting topics in this issue, such as the debugger or the Network Viewer™.

Our partnership with GE as a Solutions Provider has been very fruitful over the last year and we are pleased to show our expertise in EO Web in this issue of MDT Magazine.

Please make your own impression by reading the magazine and feel free to ask any questions!

We would be happy if you would share your thoughts on the magazine with us, feedback is always helpful (support@mdt.net).

With best wishes and regards.

MDT Team
Further support for Magik on Java™ provided by MDT: debugger

Introduction

In one of the recently issued MDT Magazines we described the benefits and possibilities of the brand new Magik on Java™. As a main gain of the new Smallworld™ platform the significantly higher performance of the running application was mentioned.

However, in our opinion the greatest change will be the new feature that allows the user to combine the programming languages Magik and Java with other tools to develop GIS applications. The Java environment, as a very up-to-date technology, provides an extremely wide and “ready to use” set of tools, libraries and solutions, in many different fields. Those tools, libraries and solutions are, in my opinion, the strongest benefit which is provided by the entire Java “world”.

However, to ensure that developers stay productive in this combined environment, the used IDE (Integrated Development Environment) should provide as much support as possible. Here MDT comes into play as this modern IDE will follow up the changes made to the entire Smallworld™ platform and provide the best support for its new runtime and available features.

The recently released version of MDT 3.4.1, together with the first version of the Magik on Java plug-in, has provided basic support for Magik on Java runtimes and sessions. New versions of the MDT Product and the plug-in have provided further Magik on Java™ related features including the debugger.

Debugger

We might meet smaller or larger bugs almost every day working as a developer, especially when the given complexity of the application increases. In such cases help, provided by a debugger, might be invaluable. It is really difficult to imagine of finding bugs, in the implementation and analysing related code execution flows without a debugger.

Therefore the integrated MDT Magik debugger was always considered to be a very
important part of the entire MDT product. Keeping that in mind, the MDT Team has taken on the challenge to create an MDT Magik on Java debugger for the incoming Smallworld™ 5 platform.

Features and possibilities

The entire Magik on Java™ platform is based on the standard Java Runtime Environment at the bottom of the technology stack. Therefore the MDT Magik on Java debugger is based on the commonly used Java debugger provided by the Eclipse Community in Java Development Tools (https://eclipse.org/jdt/). The main challenge to accomplish was to revert the job done by Magik on Java runtime. To do this we had to translate back the Java debug information to a format visible from the Magik language perspective. This goal has been successfully accomplished and Magik on Java Smallworld™ developers are now able to set breakpoints in a Magik source file, inspect Magik variables and navigate on the entire execution stack, which are expressed as Magik methods or procedures.

Additionally all the other standard execution flow control commands (Step Over, Step Into or Step Return) are fully supported and interpreted, from a Magik perspective, by default.

Another great feature is the possibility to debug Magik and Java code at the same time. The new platform supports the execution of Java code from within the Magik code. Debugging such a combined development is also fully supported and the transition of the execution flow from Magik to Java can also be performed smoothly. There is only the need that corresponding Java sources can be accessed by the debugger. This can be ensured for example if the mentioned source files are available in any Java project which has already been opened in MDT.

For more advanced developers

However more advanced SW GIS developers, who have more detailed knowledge about Java and the entire Magik on Java runtime, can utilise more of the low level view, where the exact output of the Java runtime is shown. For example, the debugger options can be modified in Window > Preferences > Magik > Debug > Magik on Java as is shown in Fig 2. The following things can be customised depending on the current developer needs:

- Stack frames names translation to its representation in Magik might be bypassed.
- Variable-type-names translation to its Magik form might be bypassed.
- Feature to hide Java stack frames between “Magik ones” might be bypassed.

Comparison with the MDT Magik debugger for Smallworld™ 4.x

The fundamental concept of the MDT Magik debugger for Magik on Java™ was to keep it as similar as possible to the debugger for Smallworld™ 4.x. However, Java and Magik languages and their runtimes are very different in many aspects which has resulted in difficulties by making the new debugger identical to the previous one.

Eventually, the new tool consists of features which could not be possibly achieved using the previous platform but also some of the features are not available yet or work in a little bit different way. One of the limitations which have to be mentioned is that it is currently not possible to inspect values of _local or _global variables. Also evaluation of custom expressions in the Expressions view is not yet supported. However on the other hand the new platform provides full information about the entire execution stack. Therefore all the stack frames can be shown in the exact lines, in which they are currently executed. Unlike in the former version the entire stack can be inspected in the debugger to resolve the full context in which the potential bug appears. Previously it was only possible to resolve that, if the given method was executed from another method, but the exact line of code could not be shown.
We have expertise in Electric Office Web™!

In any electric utility, there will be many (perhaps 5 to 10 times) more users who need read-only access to EO data, as compared to those who actively update the network via the main EO client. The large majority of these users will use software daily, often for long periods of time, performing various query, mark-up and navigation functions to complete often short-cycle tasks that feed into the wider network of planning, maintenance and service operation functions.

In the long run, after the initial release, a smaller number of engineers/clerks will need to make small changes to their designs or apply updates to the system from time to time. These changes can include adding new services or the placement of simple objects, like poles and meters, to the network.

To fulfill the user-needs the applications must meet some crucial and specific criteria like:

- Simple and intuitive to use.
- Workflow driven to support specific user expectations.
- Simple to deploy allowing a quick installation and an upgrade model.
- Scalable to meet large user base (~2,500 users).
- Limited impact on the existing IT infrastructure.

The solution will be a standalone web application covering all the key workflows outlined within this document.

The EO Web product consists of extension modules to the Smallworld™ Geospatial Server and client applications that are deployed, using a light-weight application server. It includes a number of third-party open-source components, used for map rendering and databases, for storing temporary information from the Smallworld™ VMDS, to providing the best performance of map rendering of the network data and asset details.

GE Smallworld™ product suite:

- SW Core Product™ with the latest TSB available for the EO Web release.
- Task Management™
- Electric Office™
- GSS™
- GSS Extension Modules
- NV base components

Application Server:

- JBOSS Application Server

Client Web Server:

- Node.js®

Web Browser Support. The following web browsers will be supported for this release:

- Internet Explorer®
- Firefox®
- Chrome®
- Safari®
Network Viewer™ is an application used to view Smallworld™ Geospatial Information in a web-browser window.

The user can choose from Google™, Bing™ or OSM maps as the base-layer and change the look to their preference. Choosing the Google™ base-layer enables advanced features like Google™ location search or Google Street View™.

Network Viewer™ can be run on all modern web browsers like Chrome™, Firefox™, Internet Explorer™ and Safari™, and is installed on Windows™ and Linux™.

The users can customize functionalities and even develop new modules themselves, which is due to the modular structure of the application and the public API.

Besides the basic functions like showing maps, both cached and dynamic, Network Viewer Plus™ has a variety of additional advanced features like:

- **Trace** – perform, an upstream trace to find the feeder or downstream trace to find customers.
- **Find** – searches through the database, using pre-canned queries.
- **Sketch** – draws a sketch on the map that can be sent and stored in the database to rework it.
- **Object Viewer** – different properties and relations of objects can be displayed.
- **Plot** – creates an excerpt in PDF format, also works for internal maps of structures.
- **Asset search** – speed optimized quick search of assets, using prepopulated SQLite database.
- **Address search** – finding the nearest address, using the Google™ engine for the best accuracy.

When the user launches the application he is presented with the map view and a tool panel, where all additional features can be found.

Only the best, most modern and verified frameworks. Among all used technologies we want to enlighten some of them:

- **Node.JS™** – it is a fairly new open-source server-side technology that is a great alternative for Apache server. The main advantage is that you only need to know JavaScript™ and no other language. You can develop only one node application and run it on all web browsers. The node is asynchronous by default and works just on one thread. It makes it as simple as possible. When you first install the node you are provided with almost no functionality. Node.JS™ is equipped with Node Package Manager (npm) that allows you to download all modules needed to build your application.
- **Openlayers** – is a JavaScript™ framework used for displaying maps in web browse. It has a variety of features and a very big API, allowing to build rich WebGIS applications that are very similar to google maps.
- **Bootstrap™** – is a free html, css and javascript framework that allows rich client applications to be built.

These solutions have made the application very stable and robust. It has also improved the whole development process. Node Package Manager provides lots of additional tools that were used during the development process. One of the most important was the Node task runner called grunt.

Example tasks that were used are: jsbeautifier, cucumberjs (starts BDD test), karma (starts unit tests), jsdoc (build js API doc) and watch (listen to changes in code and runs all the other tasks).

Summing up, Network Viewer™ is a very modern WebGIS platform which can be scaled and adjusted by the user to his personal requirements. It enriches the usability of Smallworld™ greatly.

Pawel Pawlak, Senior Developer, ASTEC
Types of licenses available for MDT Professional Edition

Our flexible licensing system allows you to select the best solution from a range of various possibilities. Depending on your needs and requirements you can choose an option which will perfectly match with your way of working.

Find out more about licensing types in the following section!

**Single User License**

**Single User License** – is dedicated to use the MDT Professional Edition on a single computer. Once activated Single User License becomes bound to a specific computer and its hardware configuration. To install MDT on a different computer, you have to purchase an additional license. You are required to buy Annual Maintenance Services for the first year.

*Example:* Two people in your team need to use MDT. Each person has his own computer. Two Single User licenses are required.

**Customers:**

- BOSCH
- Atos
- Worldgrid
- vodafone
- NETIA
- friendlyGIS®
- regioDATA
- Central Hudson
- Gas & Electric Corporation
- Hertener Stadtwerke
- Leistung voller Leben
- mettenmeier.
- indra
- Ubisense
- GEO MAGIC
- CrowTen

**Portable License**

**Portable License** – is sold in form of a Hardware Key. It allows to install and use the MDT Professional Edition on any computer with Hardware Key plugged in. The Hardware Key is an USB flash drive with preinstalled MDT. You are required to buy Annual Maintenance Services for the first year.

*Example:* A user has both a laptop and a desktop computer. He needs to run MDT on both, but never at the same time. A Portable license will enable MDT to be installed on both computers, but only run on the one with the Hardware Key plugged in.

**Customers:**

**Floating License**

**Floating License** – enables a fixed number of users to simultaneously use MDT. Floating License Server software is installed on a server machine that acts as a license server by issuing a license to any client computer that requests one. If the number of user requests exceeds the number of floating licenses purchased, a user will need to wait until one of the active MDT users finishes working with MDT. You are required to buy Annual Maintenance Services for the first year.

For floating licensing you need two components: Floating License Server and a desired number of Floating Licenses.

*Example:* Ten people need to use MDT but only five people will likely need to be using the software at the same time. You purchase Floating License Server and five Floating Licenses. MDT is installed on each user’s computer. Any workstation can run MDT, but only a maximum of five workstations can access MDT at the same time. If a sixth person tries to use MDT, he will have to wait until one of the first five users has finished using the application.

**Customers:**

**Annual Subscription**

**Annual Subscription** – licenses offered in Annual Subscription model are valid for 1 year. After this time the license expires. You can easily extend it for another year. The Annual Subscription model is available for Single User and Floating licenses (Floating licenses require Floating Server). Annual Maintenance Services charge is included.

**Trial License**

**Trial License** – gives the possibility to use MDT Professional Edition for the period of 60 days free of charge. To continue using it later you need to buy Single User License or Portable License. The Trial has all the functionalities of the Professional Edition.

**MDT Team**
HUMOR

3500 B.C.

A WHEEL? WHO NEEDS THAT?

A.D. 1938

A COMPUTER? WHO NEEDS THAT?

A.D. 1954

ASSEMBLER? WHO NEEDS THAT?

A.D. 2009

MDT? WHO NEEDS THAT?

MDT Magazine
Publisher ASTEC Sp. z o.o.
© 2015 ASTEC Sp. z o.o.
All rights reserved.

MDT is a product of
ASTEC
ul. Wyspianskiego 11
PL 65-036 Zielona Góra
Phone: +48 68 422 68 00
Fax: +48 68 422 68 97
info@astec.net www.astec.net

Visit our website:
www.mdt.net
www.gislet.net
www.rublon.com

Trademarks and registered trademarks. MDT Magik Development Tools is a registered trademark of ASTEC Sp. z o.o. Smallworld is a registered trademark of General Electric Company. Eclipse is a registered trademark of Eclipse Foundation, Inc. Other company or product names mentioned in this publication may be trademarks or registered trademarks of their respective companies.
Your good choice

Advanced Software Technologies Experience Cooperation

www.astec.net