Dear Readers,

We present to you the first issue of MDT Magazine. Our aim is to share information about MDT Magik Development Tools™ (www.mdt.net) - powerful and ergonomic environment created by ASTEC for developing Smallworld™ solutions.

The Magazine is addressed to anyone who is interested in geospatial systems built with use of Smallworld™ technology and Magik programming language. We would like to publish some technical details, useful hints for developers, results of tests and many other interesting information. We welcome all readers to send us remarks about published content and the wishes concerning topics worth to write about.

In the first issue of MDT Magazine we would like to introduce ourselves and MDT. In the News section you can read about the acknowledgement of our project by GE Energy. The Technology section contains an interesting comparison of MDT and Emacs efficiency and some information about the new debugger for Magik. In the last Interview section Mr. Oliver Löken shares with us his impressions about implementing MDT in GIS Consult company.

ASTEC, MDT Team

Contents

Let us introduce ourselves and our product - MDT 1
GE Energy acknowledges MDT as reliable IDE 1
Benchmark: MDT versus Emacs 2
MDT Magik Debugger 3
Interview with Mr. Oliver Löken - Implementation of MDT in GIS Consult company 4

Let us introduce ourselves and our product - MDT

ASTEC (www.astec.net) established in 1993 in Zielona Góra, Poland, is a software company which implements geospatial solutions addressed to the companies from utility as well as for telecommunication and transportation sectors. Since 2003 company has been cooperating with GE Energy and became a certified GE Solution Provider. ASTEC delivers software development services to GE Energy and their partners.

The history of creating MDT starts in 2006. During developing complex Smallworld™ systems the need of improving programming efficiency appeared quite quickly. The available environments were not satisfied for ASTEC developers. Strong knowledge about different tools used in other projects additionally stimulated us to create an integrated environment that can help with efficient creating, controlling and debugging of Smallworld™ applications.

For next years MDT has been constantly carried out. During the work we have become sure that MDT will be a perfect answer to the needs of decreasing cost and increasing efficiency of the programming process.

And what is MDT now? It is a modern Integrated Development Environment (IDE) for Smallworld™ software developers following industrial standard based on Eclipse. MDT Professional Edition enhances development capabilities and optimizes the software creation process. To learn more about MDT we invite you to visit www.mdt.net.

GE Energy acknowledges MDT as reliable IDE

Magik Development Tools became a significant part of GE Energy’s strategy related to geospatial software applications development.

GE is pleased to see third party applications such as MDT that enable customers to customize applications on Smallworld™. Confirmed Martin Ansell, General Manager Smallworld™ of GE Energy. The architecture of our Smallworld software products and the use of open standards has encouraged a fast growing development community. We will continue to closely support companies like ASTEC in support of our position as a leading provider of geospatial solutions.

Thanks to GE Energy’s support ASTEC plans to publish a new MDT release compatible with the incoming Smallworld™ CST version in 2010. Strong cooperation with GE Energy is essential for providing Smallworld™ community with the new powerful development environment.
Benchmark: MDT versus Emacs

MDT is getting more and more popular among the Smallworld™ community while the editor of choice for most Smallworld™ Magik programmers is still Emacs. MDT is a graphical integrated environment based on Eclipse. On the other hand, Emacs is a feature-rich text editor with lots of commands and macros. But which one is better, more productive and more comfortable? To answer this question we performed specially prepared test.

The comparison was performed with using a MDT Professional Edition 1.0.0 and Emacs provided with Smallworld™ CST 4.1.1. Test cases were carried out by four Magik developers with different skill levels, trained in both Emacs and MDT.

### Environment Tests

<table>
<thead>
<tr>
<th></th>
<th>Emacs</th>
<th>MDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>First launch</td>
<td>15 sec</td>
<td>20 sec</td>
</tr>
<tr>
<td>Next launch</td>
<td>10 sec</td>
<td>12 sec</td>
</tr>
<tr>
<td>Allocated RAM</td>
<td>14 MB</td>
<td>56 MB</td>
</tr>
</tbody>
</table>

### Running Cambridge session

<table>
<thead>
<tr>
<th></th>
<th>Emacs</th>
<th>MDT</th>
</tr>
</thead>
<tbody>
<tr>
<td>First launch</td>
<td>16 sec</td>
<td>25 sec</td>
</tr>
<tr>
<td>Next launch</td>
<td>10 sec</td>
<td>12 sec</td>
</tr>
<tr>
<td>Allocated RAM</td>
<td>14 MB</td>
<td>64 MB</td>
</tr>
</tbody>
</table>

### Development Benchmarks

The test consists of ten cases which reflect daily routines of developing Magik based software.

#### Test 1: File Reviewing
Discovering given artefacts in a large Magik file: variables, definitions, methods declarations, etc.

#### Test 2: File Comparison
Comparing and finding differences in two slightly different Magik files.

#### Test 3: Navigating among Files
Opening and navigating among many files edited in the development environment.

#### Test 4: Traversing Source Code
Jumping to method definitions of given methods contained in a Magik file.

#### Test 5: Code Writing
Creating a new file and typing in a Magik class source.

#### Test 6: Misspelling Correction
Finding, correcting syntax errors contained in a Magik file and formatting the source code.

#### Test 7: Building Products and Modules
Creating folder structure and template files.

#### Test 8: Searching in Files
Searching a group of files for a given text.

#### Test 9: Loading Products and Modules
Loading sources of products and modules to a session.

#### Test 10: Creating Load List
Creating load lists containing given Magic source files.

### Summary
The test results show that, in context of productivity and work ergonomics, work with MDT is more efficient than with Emacs. In comparison with Emacs, using MDT reduced development time by 38%. Also the work ergonomics has been improved over 30% thanks to reducing of mouse clicks and keystrokes. The results confirm that MDT significantly accelerates the software development process.
How important for programmers a good debugger is – everyone knows. Debuggers offer the ability to stop a program at a chosen place due to an explicit request. They also provide the current state of involved objects together with a stack trace. Good debuggers should also offer more sophisticated functions such as running a program step by step or stepping into method invocations. They should give access to all properties and methods of the object and permit modifying values and passing parameters to object methods. A well-planned and implemented debugger is one of the most crucial feature in IDE for every software developer. Debugging is also something that a Smallworld™ software developer cannot avoid.

History of debugging tools for Smallworld™ began with Magik Studio, which included an integrated debugger. This tool was well assessed by users but its further development has been stopped. Subsequently a functionality allowing the debugging was added to the Smallworld Core. This mechanism let us build a new Magik debugger in order to make developers’ life easier. The goal was to deliver a professional, efficient and user-friendly debugger integrated with the MDT environment.

MDT is built on Eclipse that provides a framework for building and integrating debuggers known as the debug platform. Large amount of modern programming environments are based on this framework (e.g. debugger in JDT). The debug platform defines a group of interfaces that are modeling a set of artefacts and actions common to many debuggers, collectively known as the debug model. The platform provides a basic debugger user interface (the debug perspective). However, it does not provide an implementation of a debugger. To enable debugging of Smallworld™ applications the platform was enhanced with features specific for Magik language. The MDT Magik Debugger is a new specialized debugger built in this way.

MDT Magik Debugger architecture is divided into two layers (Picture 1). Magik Debug Helper module is a part of Smallworld™ Layer. This module is responsible for managing the debugging process, controlling the application state and collecting data necessary for user interface. Smallworld™ Layer contains advanced mechanism which evaluates Magik expressions. This mechanism is used by Magik Debugger which is the main element of the MDT Layer. Magik Debugger implements the debug model described earlier. This element requests data from the Smallworld™ Layer and presents processed information into specific IDE views (e.g. Debug view). Communication between the layers is realized through TCP/IP connections.

Main advantage of the new Magik Debugger is complete integration with the MDT environment. For debugging purposes MDT provides a special Debug Perspective designed for debugging Magik programs (Picture 2). It includes a Magik editor and a set of views: Debug (Picture 3), Variables, Breakpoints and other. This perspective presents the current state of the debugged program.

MDT Magik Debugger is currently the best tool designed to debug Smallworld™ 4.1.x applications. The new debugger aroused great interest in the Smallworld™ community. The MDT Magik Debugger, together with numerous tools and wizards integrated in MDT, significantly accelerate the development of industrial-strength Smallworld™ solutions.

**MDT Magik Debugger features:**
- Running GIS session in debug mode
- Managing breakpoints
- Managing application state
- Performing step by step program execution
- Continuing program execution
- Inspecting variables and values
- Tracking and changing variables value
- Managing watch expressions
Interview with Mr. Oliver Löken
- Implementation of MDT in GIS Consult company

Mr. Löken, you are the Project Manager in GIS-Consult Company and you decided to introduce MDT as an IDE for your Smallworld™ developers. Please let our readers know, what was the reason to make this decision?

I have been observing MDT development since 2006. I downloaded a version of this environment quite early and I immediately tried to apply it to our everyday work. Emacs was the development environment of choice for many years, so I was very pleased because someone raised Smallworld™ development to a new level and finally created a new, integrated environment. Usage of Eclipse framework was a very good decision. Having carried out some Java projects we already gained experience in the use of this IDE. The first steps with MDT were so promising that I took a decision to observe the further development of it. Together with ASTEC co-workers I made intensive and constructive talks on this subject on every occasion during various conferences. A changeover to MDT Professional Version 1.0.0 was then for us only a consequent conclusion of a long-time evaluation process. I think it was a good decision.

What benefits do you expect after MDT implementation?

Through the implementation of MDT I promise myself a simplified, accelerated and bug-free software environment. Of course, at the beginning there appeared an increased need of adaptation among developers. Someone who has been making only Emacs development for many years, has to first get used to working with IDE. However, experience shows that they can succeed within a short time and from then on the advantages of MDT and Eclipse start to have an effect.

What are the strongest points of MDT in comparison with previously used environment?

Let us be honest, previously there was no real development environment for Smallworld™ GIS. Certainly we all got used to Emacs, but thinking about a tool for modern software development I imagined something else. Some Smallworld™ developers can possibly still recall Magik Studio. It was already a step in the right direction, which we welcomed very enthusiastically. All the greater was the disappointment as this tool disappeared from the scene after a short time. However, with MDT appeared a reliable successor which, in the meantime, outruns Magik Studio.

One of the most important functions of MDT is a debugger. By means of this tool, an error search can be immensely facilitated and accelerated. Unfortunately, this functionality is available only under 4.1.x and our projects run largely under GIS 4.0. Further very helpful functions are Code Assist, Magik Hyperlinks, Product Explorer and Outline View.

What kind of projects are you currently carrying out using MDT?

The use of MDT in our company takes place during implementation of customer projects, as well as during development of our own projects and support. A big advantage for us is the possibility to import and export sessions and runtimes. Thereby it is very easy for us to pass on the projects to other developers.

How many developers currently work with MDT?

About 25 Smallworld developers work for GIS Consult GmbH. In the first round we purchased MDT Professional licenses for 11 developers. These were the employees who earlier already used the free version of MDT. This facilitated the switch to MDT as well as familiarisation.

Do developers consider MDT a good set of tools?

Before purchasing the licenses we conducted an intern survey among our developers and asked them about advantages and disadvantages of MDT. The positive feedback from the developers was crucial for our switch to MDT.

Does the training help developers to improve their productivity?

11 developers took part in 4-days ASTEC training. Even though on the first day there were many familiar issues discussed, we could learn a lot of new functions during the following training days, especially a possibility to set diverse preferences was not previously familiar to us. Therefore I can recommend the training to everyone.

Would you recommend MDT to other companies? Why?

The answer is obvious: YES. The approval of MDT among our developers set here a clear example. When I walk through our offices today, I see MDT more often than Emacs. In my opinion, an everyday work in the field of software development will be easier and better owing to MDT.

Thank you for the interview!

Dr. Oliver Löken, Project Manager in GIS Consult GmbH
Email: loeken@gis-consult.de