Maturity and stable development with MDT PE 3.2-3.3 release
by Błażej Sytar, Sales & Support Leader, ASTEC

Version 3.2 and 3.3 were focused greatly on refactoring and optimization efforts. Among others we have introduced many significant improvements and new functionalities in the Debugger and greatly refurbished our Magik Parser for even better performance. MDT MStyle plug-in, released with MDT 3.2, was another big addition to the MDT Enterprise package of dedicated add-ons.

New Features

• **Upgraded Debugger** – many significant enhancements and performance optimizations were introduced to the MDT Debugger. Now you are able to run multiple threads hitting the same breakpoint. Furthermore the support for `print`, `debug_print` commands and support for database objects has been added. You are now also able to track changing values for all kinds of variables.

• **Categorize your Sessions** – you can create and manage categories for grouping elements available in Runtimes and Sessions tabs. Categories were introduced for cleaner and more organized workbench.

• **Session-oriented focus** – Magik programming is often more focused on working with sessions rather than projects, so we have introduced a new way of organizing your Product Explorer by drawing out the Session to the top-level and placing all dependent elements underneath.

• **Support for slotted exemplar slots** – slots are now supported by Magik Browsing and Magik Search functionalities. Magik Editor also supports various actions for slots edition and validates them accordingly.

New Extensions

• **MStyle Plug-in** – it can check many different aspects of the source code, like metric rules, some invalid/deprecated source code elements. You run it on any resource container (project, folder or file) to check its code using defined rules. It contains over 30 most commonly known predefined and customizable rules. This plug-in allows to introduce modern team standards and automatized Quality Assurance to your Magik projects.

Dear Readers,

Another year has passed. In many ways it was a very challenging year. We are very proud that thanks to the friendly cooperation with Realworld Systems company from Netherlands we have been able to create and publish our Quality Assurance tool - the MStyle plug-in. It now serves many users out there to keep their code at the highest standard.

We are happy to place in your hands the 2013 Issue of MDT Magazine. In it you can find what is new in the recent releases of MDT 3.2 and 3.3. We have done a lot of refactoring and optimizations work so the tool is now even faster and more reliable.

Furthermore you can read about the first impressions and learn a handful of hints about recently published evolution of Magik SDK – SWIFT.

Next there is a short introduction of our latest product – Gislet. It’s a state-of-the-art mobile client for Smallworld running on Android. It is one of the first attempts to bring access to your GIS data natively from the mobile device also in offline mode.

In the middle you can find a full list of MDT’s references from around the world. In our customary interview section one of our recent users, Peter Reale from Central Hudson company, will share with us his impressions after switching to MDT Professional.

Last but not least we again encourage you to benefit from our training capabilities to even better maximize your productivity with Magik Development Tools.

Best wishes,

MDT Team

Are You new to MDT?

Visit MDT online if you would like to know

» why MDT is the most intuitive IDE for Magik development,
» how it will increase your programming efficiency and
» why it is being used by companies all over the world.

Try out the 2-month trial, watch our video tutorials and subscribe to the newsletter.

Visit us today at [www.mdt.net](http://www.mdt.net)

Inside...

Maturity and stable development with MDT PE 3.2-3.3 release ............1
SWIFT - revolution or evolution? ............2
Gislet - GIS mobile client for Smallworld™ .................3
MDT Users Worldwide .....................4
From Standard to Professional – MDT in Central Hudson ............6
Get Yourself Trained! .....................7
SWIFT - revolution or evolution?

by Paweł Jeremicz, Project Manager, ASTEC

I would bet that every Magik developer, who at least once designed and implemented complex SmallWorld™ user interface, will admit that it is not an easy task. Repeatedly nested rowcol elements, numerous tree_item parameters required for configuration or limited customizability of image_choice_item elements are not the only obstacles along the way. The use of SWAF (SmallWorld Application Framework) architecture components such as classes inheriting from gui_framework and defining the graphical interface in XML files with the use of sw_action also has its limitations. Finally it is impossible to use SWAF-based solutions for special case scenarios, such as displaying interactive and multilayered data structures. ThIng well known due to the limited list of available controls (sw_action, gui_control_constructor) and containers (<toolbar>, <menu>, <statusbar>, <paned_window>, <tab_box>) . Despite the fact that combining the manual creation of GUI in the code with SWAF architecture based on XML gives a wide spectrum of possibilities, it is not a complete remedy. What about operating systems other than Windows? UNIX users are completely deprived of the possibility to run any application with graphical environment. What about adjusting the look & feel of specific controls to customer requirements? The style of used system widgets is also far away from the design of newest systems from Microsoft Windows family. And here SWIFT comes into play (SmallWorld user InterFace Toolkit). Its official premiere was launched together with the new SmallWorld Core Spatial Technology 4.3, but already in previous version 4.2 we could observe first experiments with the use of this technology (check „Upgrade GUI module”). Now, I would like to share with you my impressions from the upcoming version updates. I noticed the benefits of using SWIFT right away. To me personally, the biggest advantage is the ability to create hierarchically organized components with the use of basic SWIFT elements and the freedom to shape the control directly from the code level. We can have an entirely new GUI element for certain requirements. For example thanks to combining sw_text_item and sw_button_item into one sw_border_container you receive a completely new widget ready to reuse. Imagine how easily we can now create even the most complex lists based on sw_table. Each cell of such table is nothing else but a container, which content is put together with the available or newly created SWIFT controls.

For demonstration purposes. We are supposed to use an orderly knowledge base. Personally, I class Browser or Datamodel Tools and Backup Manager. Since SW 4.3 doesn’t yet give the possibility to create an entire application using only SWIFT, I decided to prepare an independent module as a testing ground. To be able to check the declared full compatibility with SWAF I used the gui_framework to write my dialog. You have to remember to set the value of gui_type on swifT - only this way we can force to translate the entries in gui.xml file to SWIFT components. Building GUI in XML with the use of new components is possible and delivers a few additional options, such as new container classes like sw_tab_container or sw_card_container. You cannot avoid certain limitations typical for every software’s early versions. The lack of statusbar implementation, unsupported layouts or GUI parameter hidden are the most perceptible of them. Yet, the documentation clearly states that this version delivers only a handful of ready-to-use widgets and a whole bunch of unsupported component prototypes for demonstration purposes. We are supposed to be getting new widgets together with the forthcoming version updates.

What is SWIFT? It is a library of GUI components rendered completely in the Magik code, intended to be fully compatible with SWAF architecture. Example of its usage in SW 4.3 is, among others, the Quality Manager or Datamodel Tools and Backup Manager dialogs.

A first positive surprise was the Swift Development Guide— a documentation in form of a HTML, located in magik_gui_components module – which brings to mind the best standards of Java-doc or MSDN Library. Of course there is nothing that can substitute the Class Browser, but it is heartwarming to be able to use an orderly knowledge base. Personally, I only missed the possibility to search through the documentation, which is in fact quite big. Since SW 4.3 doesn’t yet give the possibility to create an entire application using only SWIFT, I decided to prepare an independent module optimized for best performance. To definitively answer the title question: I think that SWIFT is a long awaited upgrade that brings the front-end of the application closer to modern standards. Thanks to maintained compatibility with SWAF it is not a costly revolution, but a natural evolution of Magik SDK. I will be impatiently awaiting the new releases of libraries and announced improvements.
Gislet - GIS mobile client for Smallworld™

by Błażej Sytar, Sales & Support Leader, ASTEC

In Spring 2011 we started working on a concept of a mobile solution supporting Smallworld GIS. The plan was to create a true native client – so it would be fast and responsive; a simple one – so it would be easy for the end-user; a flexible one – so it was highly customizable and could integrate with other systems on different levels; and a solution that would have a genuine offline mode – so it can work anywhere and anytime. This is how Gislet came to life…

What is Gislet?

It’s a solution dedicated to the field workers to allow access to your company’s GIS data in the field. It gives you the ability to perform most crucial actions during routine inspections or maintenance tasks, such as viewing, searching and navigating over the assets. With the use of a wide range of available redlines you can easily add your comment, highlight important objects, mark faults or leaks directly on the map. Gislet was created as an Android native application to deliver the best responsiveness, stability and seamless communication with various built-in devices.

On top of that, the solution provides you with the means for central management and deployment of mobile GIS data, maps and tasks. It helps to improve your field operations and make timely and informed decisions. The big advantage is that you can work either in Online or in Offline mode.

Offline

Offline mode provides constant access to spatial data despite your location or network coverage. All required data has to be previously prepared in form of an extract and uploaded to the mobile device. A part of Gislet solution is a set of tools that allow you to easily prepare and manage any set of spatial data in form of extracts. Incremental extracts are also supported to minimize the update efforts in case of frequent changes in the database.

Both modes, Online and Offline, have the identical set of functionalities such as: searching, viewing, inspecting, making notes, navigating, measurement actions, etc. The only difference is the source of spatial data.

Online

Online mode provides a real-time access to the up-to-date asset data taken directly from your company’s GIS system. We are also able to immediately send some information to the GIS database in form of notes with redlines. The only downside of this approach is that we have to stay connected to the Internet via GSM or WiFi, which is not always possible in field environment.

Integration & Customization

Gislet was designed to be a flexible and customizable platform with extension points for easy integration with various GIS systems, as well as Task or WorkFlow Management systems. The default configuration allows for displaying data taken from several map sources: external and customer’s internal. The integration can be done on either front- or back-ends for which purpose all necessary APIs are delivered.

We are also able to immediately send some information to the GIS database in form of notes with redlines. The only downside of this approach is that we have to stay connected to the Internet via GSM or WiFi, which is not always possible in field environment.

Integration & Customization

Gislet was designed to be a flexible and customizable platform with extension points for easy integration with various GIS systems, as well as Task or WorkFlow Management systems. The default configuration allows for displaying data taken from several map sources: external and customer’s internal. The integration can be done on either front- or back-ends for which purpose all necessary APIs are delivered.

FUNCTIONALITIES

- Access the company geospatial data wherever you need it
- Online access to your GIS system
- Offline mode for extra reliability
- Integration with Task & WorkFlow Systems
- GPS for navigation, footprints and positioning
- Notes and marking objects on the map
- Searching for objects and viewing their attributes
- Displaying external maps (served as WMTS, WMS)
- Utilizing administration modules

Learn more at www.gislet.net
MDT USERS WORLDWIDE

America
- Central Hudson
- City of Bloomington
- FirstEnergy
- Integrys
- Manitoba Hydro
- Videotron

Europe
- GE
- ASTEC
- ENERGIE GRAZ
- friendlyGIS
- GEOMAGIC
- INTARI
- MAANMITT AUSLAITOS
- regioDATA
- RWTH Aachen University

Asia
- BOSCH
- ewZ
- GIS Consult
- LOGIBALL
- NSW Transport RailCorp
- Smallcases GmbH

Australia
- NSW Transport RailCorp
- NIS
- realworld systems

www.mdt.net
From Standard to Professional – MDT in Central Hudson

One of the first releases of MDT was a Standard version – a simple Eclipse-based editor with a handful of functionalities. Still, it gained a good deal of interest and appreciation from Smallworld users across the globe thanks to its modern looks and work methodology. It was also a good introduction for the fully-featured Professional, which came out soon after. Today we will share a short interview with Peter Reale, a former MDT Standard user from Central Hudson – a Gas & Electric Corporation from the state of New York. Peter has just recently upgraded to the Professional Edition. Let’s hear what he has to say…

[Blazej Sytar] Good day Peter. Thank you for agreeing to answer some questions. For starters I would like to ask you to tell us a few words about yourself and your experience with Smallworld.

[Peter Reale] I am a software developer with experience in a wide range of programming languages on various platforms, most of which has been with IBM Global Services creating enterprise-wide, Lotus Notes applications and Domino web sites. I have struggled to learn Smallworld/Magik on my own, and have been disappointed at how little can be found on the internet, as far as knowledge sharing and tutorials. I have been supporting our company’s GE PowerOn system for 3 years, with no previous experience in Magik or Emacs.

[BS] What is your current job description at Central Hudson? What kind of work do you do there mostly?

[PR] I am a Systems Analyst, supporting our Outage Management System (PowerOn), portions of our company’s internet and intranet web sites and our time sheet system. I also developed and maintain our mobile app for iPhone, Android and BlackBerry, which allows customers to pay their bill, report an electrical outage and get status of their outage.

[BS] You weren’t completely new to MDT when you have decided to get the commercial licenses. In fact you had been using the free Standard version for quite some time. What is your impression about it after working with it for the last few years?

[PR] Since I have had previous experience with the Eclipse platform, I found the IDE familiar. Creating projects in the Product Explorer allows me to quickly access code I’ve written for either our Test or Production environment.

[BS] How does the current Professional version compares to the former Standard?

[PR] I use Project Dependencies to access classes, instead of Linking various resources. This is convenient because once I add the folder for our customizations, I can quickly get to all our modules. I also find the Hierarchy view helpful to see the complete set of methods defined in a class, even if they are located in various Magik files.

[BS] And how does Emacs compare to them both?

[PR] I am not a fan of Emacs. My only experience with it was when I started learning Magik Language a few years ago, and I only used it until I found MDT. Having to remember key combinations, and figure out how to switch between buffers was very frustrating. With MDT, I’m not slowed down by the user interface; it’s intuitively obvious.

[BS] What kind of project are you currently carrying out using MDT?

[PR] I use MDT to provide production support of PowerOn, and create any minor customizations.

[BS] Does MDT correspond with your needs concerning teamwork and/or version control?

[PR] Yes, MDT/Eclipse provides tools to easily share and compare different versions of our code.

[BS] What MDT functionalities do you value the most in your work?

[PR] The MDT functions that I use the most are: Linked Files, Code Assist, Class Browser and Magik File Search.

[BS] How can the Eclipse multi-platform ability facilitate the everyday work with Magik via MDT?

[PR] Having a single IDE for working with multiple languages (Java and Magik) makes it simpler. It gives a consistent framework so you don’t waste time trying to remember separate development processes.

[BS] If there is one thing you could change, what would it be?

[PR] If there is one thing I could change, it would be the ability to associate my projects in the Product Explorer with more than one session. That way I can compile/transmit my Magik code into different environments (Dev/Test/Prod), without having to duplicate code or re-associate my project with a different Session.

[BS] Would you recommend MDT to other Magik developers?

[PR] Yes, definitely; it will improve your productivity!
MANY SATISFIED CUSTOMERS FROM DIFFERENT CONTINENTS ALL CLAIMING THAT THANKS TO THE TRAINING THEY ACQUIRED A COMPREHENSIVE KNOWLEDGE OF THE TOOL AND COULD UTILIZE THE MOST ADVANCED DEVELOPMENT TECHNIQUES WITHIN FEW WEEKS FROM THE INTRODUCTION!

Trainings can be held in our dedicated training room in ASTEC HQ or at your site!

2-day Standard Course

<table>
<thead>
<tr>
<th>Days</th>
<th>Topics</th>
<th>Exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1. Introduction to MDT</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2. Installation and configuration</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>3. Managing Routines and Sessions</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>4. Working with Magik projects</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>5. Magik code development techniques</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>6. Using additional tools supporting</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>7. development</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>8. Interacting with Sessions</td>
<td>✓</td>
</tr>
<tr>
<td>Day 2</td>
<td>9. Working with Class Browser</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>10. Using Magik Search tools</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>11. Performing text searches</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>12. Browsing Type Hierarchy</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>13. Code navigation with Magik Hyperlinks</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>14. Using Magik Debugger</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>15. Tips &amp; Tricks</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>16. Overall discussion</td>
<td>-</td>
</tr>
</tbody>
</table>

1-day Extension Course

<table>
<thead>
<tr>
<th>Days</th>
<th>Topics</th>
<th>Exercises</th>
</tr>
</thead>
<tbody>
<tr>
<td>Day 1</td>
<td>1. Extending MDT and Eclipse</td>
<td>-</td>
</tr>
<tr>
<td></td>
<td>2. Installing additional plug-ins</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>3. Team work in MDT</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>4. Integration with Issue Tracking System</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>5. Task-oriented development</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>6. Composing multiplatform environment</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>7. Using additional useful plug-ins</td>
<td>✓</td>
</tr>
<tr>
<td></td>
<td>8. Overall discussion</td>
<td>-</td>
</tr>
</tbody>
</table>

“Even though on the first day these were many familiar issues discussed, we could learn a lot of new functions during the following training days. (…). Therefore I can recommend the training to everyone.”

Oliver Läken, GIS Consult

WRITE TO US AT mdt@astec.net
OR DIAL +48 68 422 68 12

Need help with environment set-up?

Except for dedicated trainings we also offer our help with more complex implementations, including adaptation and configuration of the entire developement environment to best fit your systems working with Smallworld™ GIS.
Powerful solutions through advanced technologies.

As a strong GE Energy authorized Solutions Provider we offer software development services for Smallworld™ users from utilities and telecommunication industry sectors.