



SPOTL IET

The Pull up to the Bumper Edition January 2008



Firstly, Happy New Year to you all from all of us. It's another year here at IET towers and now all the Christmas decorations have been taken down and the New Year celebrations are well and truly over, it's time to start thinking about the year ahead. This first SpotlIET of 2008 takes a look at our latest product offering, **VeriflEr**, as well as the usual random array of news, hints and tips from the IET information feed.



VeriflEr is a new quality assurance tool for checking your Gen models for compliancy with development and audit standards. It includes many pre-configured checks which scan and validate your Action Diagrams and other Gen objects.

STANDARDS

Most Gen sites have development standards covering aspects such as naming conventions, use of return codes, factors that affect performance, etc. Projects that out-source development activity will especially want to check the models for compliance to standards.

Checking that the model meets these standards can be a time consuming task, and this is usually allocated to an experienced Gen developer to perform manually via code inspection.

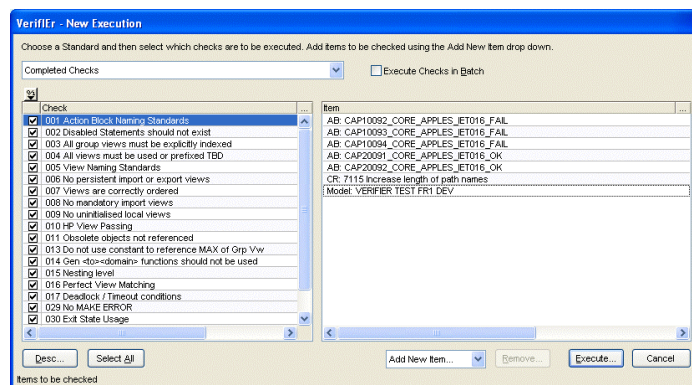
CHECKS

Using **VeriflEr**, verification checks can be performed on an entire model, selected objects or the contents of a GuardlEn Change Request, Release Pack or System Update. Automated checking means that a far greater range of checks can be performed compared with the effort involved in manual checking.

Checks can be invoked on demand, scheduled in batch, performed automatically on upload or as part of a GuardlEn System Update. The integration with GuardlEn ensures that checks are automatically performed as part of the normal development life-cycle, thus ensuring that the Gen models are compliant with standards.

Here are just some of the checks **VeriflEr** can perform...

- Naming Standards, e.g. action blocks, member names, trancodes, view names and other object names
- Coding Standards, e.g. use of exit states, checking return codes, perfect view matching, standard NOTES, avoiding use of certain functions, etc.
- CBD tiering, e.g. correct use of public operations, component consumption and data access layers
- SQL Efficiency, e.g. use of functions and choice of predicates in READ qualifiers
- Audit Checks, e.g. SOX compliance checks



The scope of the checks that **VeriflEr** can perform is almost limitless. It can verify objects meet naming standards, that code is structured in a certain way, that coding standards are adhered to, that READ statements are efficient, and much, much more.

VeriflEr contains a comprehensive library of checks that can be quickly adapted to meet a project's requirements, either via parameterisation or adapting the check logic.

VeriflEr has been designed with flexibility in mind to allow incorporation of site specific checks. These can either be delivered by IET as part of a tailored implementation, or developed as custom checks by the customer or third party developer.



Just a reminder that the IET contact telephone number has now changed to...

+44 1225 863060

...and should be used with immediate effect for contacting the office. Thank you.

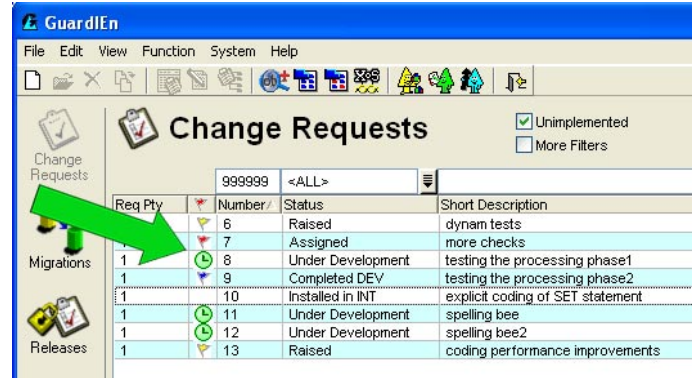


- ✓ Automated checking reduces the amount of time developers and QA staff spend on verification
- ✓ Checks can be performed without expert knowledge of site standards or the Gen meta-model
- ✓ Automatic verification on Gen upload provides immediate notification of errors to the developer, allowing them to be fixed at an early stage in the development life-cycle
- ✓ Checks can be defined as mandatory, thus ensuring that all Standards are adhered to
- ✓ Integration with GuardIEn - checks are performed as an integral part of the development life-cycle and can incorporate configuration management and audit requirements
- ✓ Allows the development of sophisticated and comprehensive checks that would otherwise be impractical to perform manually
- ✓ Gen Toolset plug-in allows developers to verify code before upload



From GuardIEn 7.5 onwards it is now possible to improve the readability of the CR list screen using a new Attention Flag. This is defined on the CR life-cycle state definition and indicates when and how to flag the CR for the user's attention.

- **Yellow** flag if the CR is in this state and the user is the CR owner
- **Red** flag if the CR is in this state and the user is the CR Assignee
- **Green** 'in progress' clock indicator if the CR is in this state and the user is the CR Assignee
- **Blue** flag if the CR is in this state and the user is a change administrator (C project access for the project)



This enhancement was enabled by the IETeGUI functionality introduced in GuardIEn 7.5 and now fully implemented in all GuardIEn list screens.

INTERESTED? THEN WHY NOT ATTEND A WEB DEMO

So are you interested? Why not see what VerifiEr has to offer from the comfort of your own desk? We have a number of IET product web demos, including VerifiEr, running in the coming weeks (all times GMT)...

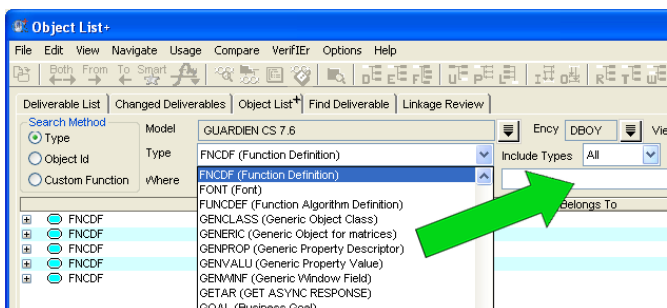
VerifiEr	30th January 10:00 & 18:00
GuardIEn	6th February 14:00
genIE	7th February 14:00
xTrace	13th February 14:00
IETeGUI	14th February 14:00
Object List+	27th February 14:00
VerifiEr	28th February 14:00

...where you can view a live product demonstration via your web browser. To sign up to these you can click [here](#) or visit the website at www.iet.co.uk/verifier for more information on VerifiEr.

Did you know that in Object List+ when searching by Type, you can choose which Object Types should be available in the Type dropdown by using the 'Include Types' dropdown. So you can display: 'All' Types valid for the selected model's



schema; 'Scoping' Types only; or 'Common' Types only. The default is 'Common', so that the list of available Types is small and includes the most commonly used Types that you are likely to be interested in. The "common" list is maintained via the Subset Object Types function available from the System Administration window.



POETRY CORNER

An irregular section bringing you illuminated prose on today's hot c.m. topics

VerifiEr Starter

*So welcome then to VerifiEr
It checks on things that could be dire*

*For ABs that were accidentally named
To subroutines of which you'd be ashamed
And commented code that shouldn't be there
Yes, it stops you pulling out your hair*

*And if your frustration is imperfect VIEWS
Or packaging that leaves you confused
A READ that reads but not quite right
This will help: it's pretty bright*

*To issue warnings when you perform DELETE
To make quite certain your code's complete
A USE that must return a code
All these and more you can now offload*

*To assess the level of your Nesting
To make quite sure you've done your testing
And, yes, if performance is your thing
Our checks will make your software sing*

*So sit back now and check out our rhyme
As with VerifiEr you'll have LOTS more time*

By EJ Bean (14%)



GuardIEn 7.7 IS COMING...

There's sadly no major CA Gen release this year but that hasn't stopped us pushing ahead with our latest GuardIEn release – to be launched in **March 2008**. The next SpotIET edition will focus on some of the main features of this exciting new release – including the launch of VerifiEr – so keep your eyes peeled!

Schuitema is a retail and wholesale company based in the Netherlands that owns supermarkets and also provides retail support to independent retailers and associated stores. Schuitema operates approximately 70 supermarkets and provides goods and services to nearly 400 independent associated food retailers, operating under the trade name C1000. Rob Jasper, a senior Gen consultant at Schuitema, tells us about their experiences with GuardIEn.

The Problem

Three years ago, the CA Gen support team at Schuitema was tasked with optimising the development process for CA Gen; specifically to raise the visibility of change management and, by extension, to reduce the time required to deploy code changes into production without any errors. There was a perception from all parts of the business that, in our complex development environment, we were being constrained by our ability to deliver changes on time, first time.

We quickly decided we needed to a) provide a change management process with clearly defined job responsibilities, workflows and actions; b) ensure decision making about testing and deployment was placed at the lowest possible organisational level and c) automate as much of the deployment of our changes as we could.

The Choice

We decided to implement IET's GuardIEn configuration management tool. It allowed us to quickly document our preferred way or working and clearly separate out roles and responsibilities: from the high level release management work of our Planners, the assignment of Change Requests by the Team Leaders, to the working of those individual changes by our Developers. Within a short space of time we had a clear and secure approach for documenting changes to the application and versioning our Gen source code as well as a fully automated mechanism for remotely deploying changes across our 3 testing platforms.

Our new approach quickly won approval from our development teams who had previously been sceptical. Suddenly they were progressing into our 3 test environments with no direct intervention in the deployment process. As an additional benefit it was soon apparent that implementation errors encountered during the build phase, previously managed with a lot of manual effort, had been reduced to negligible levels.

The Result

Using some pre-GuardIEn measurements on the activities of the 10 developers and support staff on the pilot application we made comparisons with the metrics after one year of GuardIEn use. We saved 3,400 man hours per annum over our previous approach – equivalent to nearly 2 members of our 10 person team. Schuitema was repaid its investment in GuardIEn within 18 months.

Since then Schuitema has embarked upon a major re-engineering of a core system with the assistance of CA Gen partner Jumar Solutions. GuardIEn was used throughout the project and proved essential – providing the mechanisms and controls to allow fault-free rapid deployment of our changes across all components during the re-engineering phases. We actually reduced our model management effort to a single individual as many of these tasks were completely automated by GuardIEn's robust migration facilities and huge code deployments were reduced from weeks of previous effort to less than a day. GuardIEn undoubtedly ensured the project was successfully completed within the timescales required.

In conclusion, **GuardIEn more than met our expectations** for tightly controlling the development phases of the original project in a clear and manageable way while fully automating the deployment process – winning approval from our Sarbanes-Oxley audit. Deployment was also improved with a significant reduction in build errors – thus freeing up a lot more time for developer testing and so improving software stability into the later stages of each launch.

The significant re-factoring exercise on our main application has since demonstrated such a project to be **near impossible without GuardIEn's control and deployment facilities** – the alternative is manual, error prone processes that extend project timescales.



A customer writes...

Dearest Mr Guru

Happy New Year! I was wondering if it was possible for my RI Trigger deliverable types to have a version lifecycle that is more closely tied to the Entity types lifecycle I already have in place. Somebody asked me this in 2nd life yesterday and I was intrigued to know.

Thank you for your kind attention

Clacton Hilton (avatar)

Guru responds...

Dear Simon

Yes I know it's you. A very interesting question nonetheless. We have, in GuardIEn 7.6 SP2, enhanced the RIT processing in GuardIEn to allow your RIT lifecycle to emulate a more 'typical' version lifecycle. To enable this, the RIEXSOP target property can now be updated to 'Version Status' causing the expansion to ignore an RIT in the CR content if the RIT status is at the completed state for the target environment or higher – exactly as 'normal' objects are processed.

In addition, if the version status is in a previous environment and there is an allowed state change from the current state to a state in the target environment, the RIT version status will be updated to the first allowed state in the target environment – even though it is not strictly speaking 'migrated'.

GuardIEn 7.7 will enhance this processing still further to allow RITs to be 'migrated' but that is for another time, place and world. For now be at peace with your present trigger support.

Guru

Guru thanks everyone for their questions, but reserves the right to trigger off a rush to lunch at any time.



VerifEr had left the Gen development team with extra time on their hands

Ah well. We've reached the end of our bumper edition of SpotIET – hope you enjoyed it and maybe found something useful. We'll be back very soon...



The IET Team

To ensure you don't miss future copies of SPOTLIET why not send us a blank email at spotliet-subscribe@iet.co.uk