.NET
e-zine issue 1

Automated Error Reporting for .NET and your bottom line

A dramatically different way to boost software quality

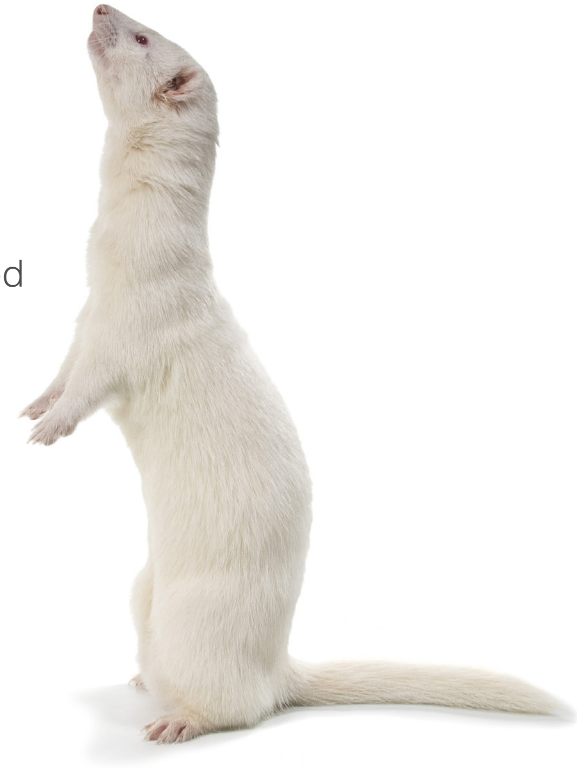
- The bottom-line rewards of finding and fixing bugs
- Your best testers still come up short
- Five reasons to use Automated Error Reporting
- How SmartAssembly works
- The power of two: Automated Error Reporting and obfuscation
- SmartAssembly and JIRA working together

redgate
ingeniously simple tools

The bottom-line rewards of finding and fixing bugs

Ignorance might be bliss, but knowledge goes straight to the bottom line when it comes to software bugs.

According to software quality experts, those who can ferret out bugs and improve the quality of their software will be rewarded with greater customer trust, higher renewal rates, lower development and maintenance costs, streamlined delivery schedules, and fewer opportunities for the competition to poach their customers.



Quality counts – big time

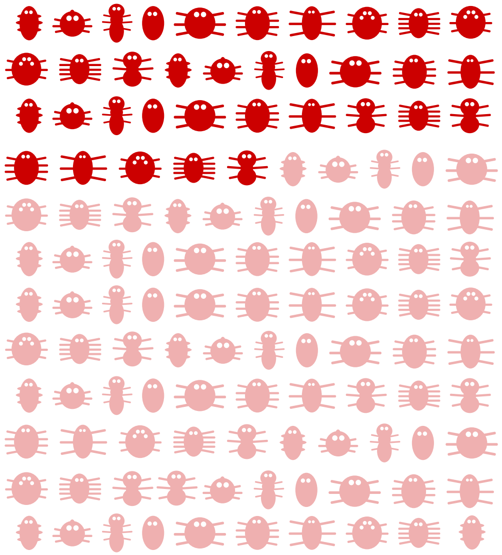
If you think quality doesn't count all that much, consider these points made by Capers Jones in an article published in the June 2011 issue of the American Society for Quality's *Software Quality Professional* journal:

- High-quality product schedules are 15% shorter than those for lower-quality products.
- Total cost of ownership for high-quality applications, from the first release through the next five years, will be about 30% lower than for identical projects with poor quality.
- Annual maintenance costs are lower by 40% for high-quality applications.
- The larger the application, the more valuable quality becomes.

It takes more than testing

Testing is only part of the answer, according to Jones, as it is less than 35% effective in finding bugs. So, what's a concerned developer to do?

"There's a testing system that will find more bugs than all the unit testing, integration testing, and QA you could possibly do," says Alex Davies, software developer for Red Gate. "It's called your users."



Effortless, but comprehensive

Tapping into the user base is the approach Red Gate took in implementing Automated Error Reporting in its SmartAssembly software. The trick – not an easy one – is to make the software simple, complete, and customizable, so the reporting process becomes almost effortless from the customer side, but comprehensive from the developer's side. Each report submitted includes the full stack trace and all the contextual information needed to find and fix the bug.

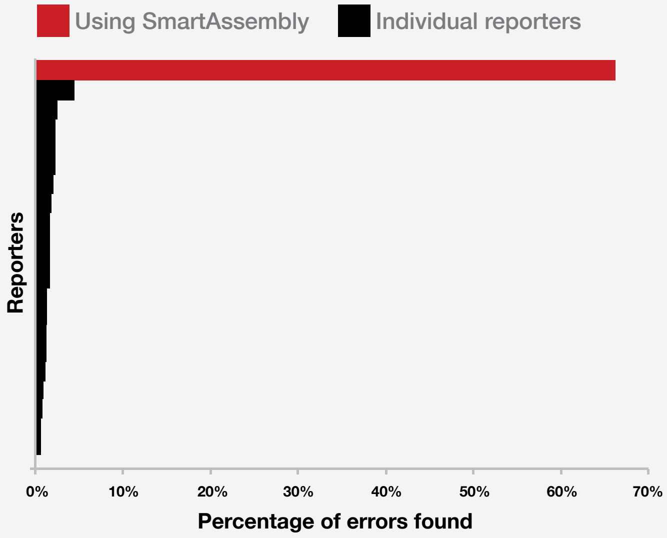
"At the end of the day we are making the reporting of issues incredibly simple for our clients," says Andrew Neville, senior software engineer at Neville & Rowe and a SmartAssembly user.

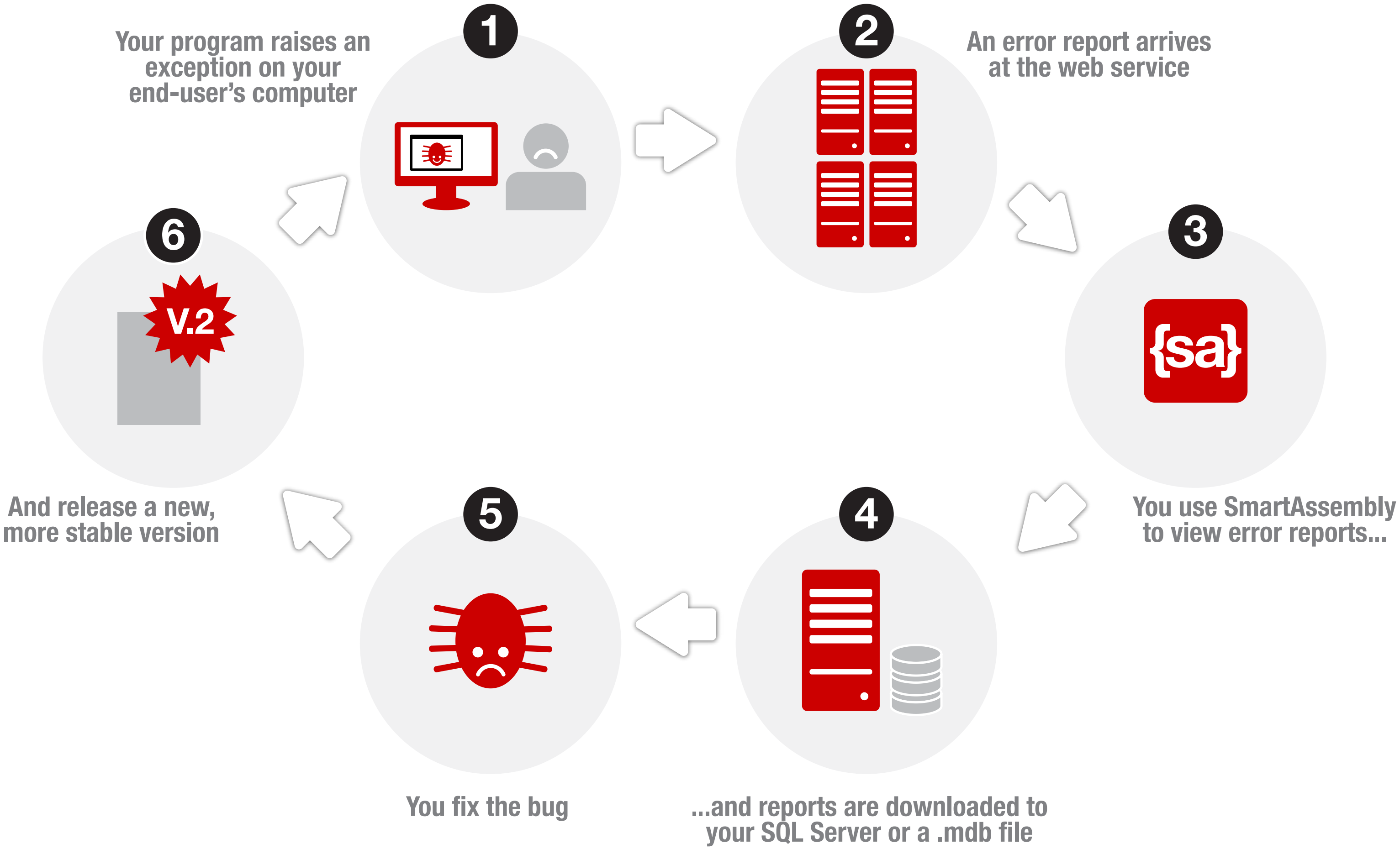
"We receive all the information we need in order to rectify the issue simply through the mechanism of the user pressing the send button. By having SmartAssembly built into our solutions we are offering our clients an extra level of service and a more direct line of communication and dialog with us."

Your best testers still come up short

In a typical quality assurance process, your development team relies on in-house testing on the one hand, and the willingness of customers to take the time and trouble to contact the product support team on the other.

The chart below, published by Red Gate, shows the percentage of errors captured by SmartAssembly compared to Red Gate's top 20 testers. The benefit of Automated Error Reporting is clear: SmartAssembly captured twice as many errors as the combined effort of the top 20 testers, or 15 times as many errors as the most prolific tester.





1 Your program raises an exception on your end-user's computer

2 An error report arrives at the web service

3 You use SmartAssembly to view error reports...

4 ...and reports are downloaded to your SQL Server or a .mdb file

5 You fix the bug

6 And release a new, more stable version

Know what you don't know

The value of Automated Error Reporting in SmartAssembly is upending assumptions and enabling software development teams to receive detailed information about exceptions they didn't know existed.

"Developers don't know quite how many exceptions occur on their end-users' machines," says Davies. "They assume that it's very few; they are always wrong. And, they assume they cannot get detailed information, such as local variable values, but they can."

Skeptical?

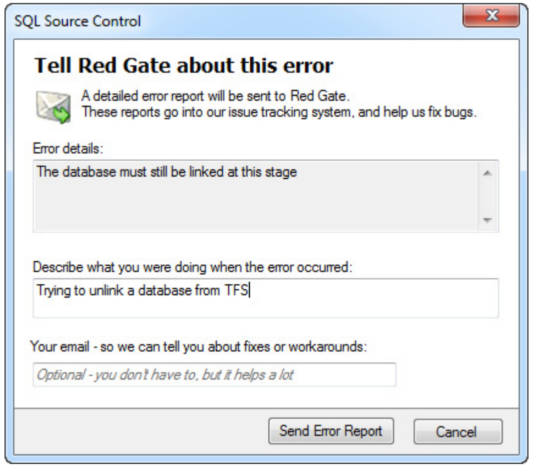
Your developers can try it out for themselves by downloading a 14-day free trial version of SmartAssembly.

How SmartAssembly works

Reports at the push of a button

SmartAssembly makes it easy for customers to tell your development team when things are going wrong with your applications. When an exception occurs, a simple exception message is displayed to the customer, enabling reporting at the click of a "Send error report" button.

There is no longer any need for time-consuming email exchanges to try and work out why an exception occurred. All of the necessary, detailed information is automatically collected, without disturbing the customer.



SQL Source Control

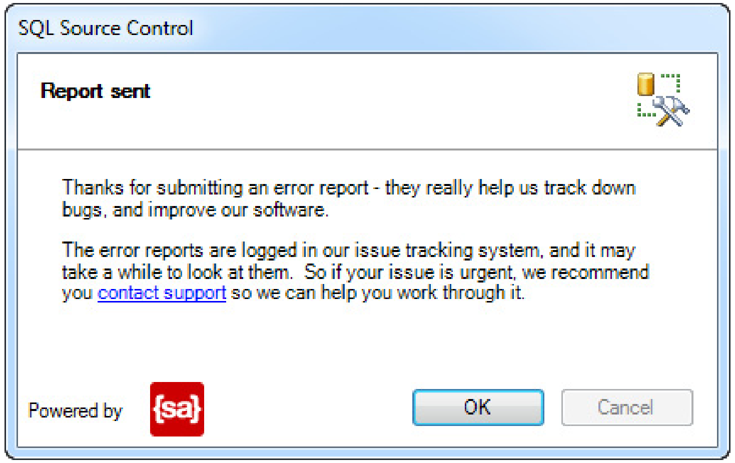
Tell Red Gate about this error

A detailed error report will be sent to Red Gate. These reports go into our issue tracking system, and help us fix bugs.

Error details:
The database must still be linked at this stage

Describe what you were doing when the error occurred:
Trying to unlink a database from TFS

Your email - so we can tell you about fixes or workarounds:
Optional - you don't have to, but it helps a lot



SQL Source Control

Report sent

Thanks for submitting an error report - they really help us track down bugs, and improve our software.

The error reports are logged in our issue tracking system, and it may take a while to look at them. So if your issue is urgent, we recommend you contact support so we can help you work through it.

The alternatives to SmartAssembly

Email or forum reporting

User response tends to be slow, many problems go unreported, and it is difficult to gather precise and accurate information on exceptions.

Writing your own Automated Error Reporting software

A major undertaking that will require on-going maintenance and will likely distract you from your core strengths.

Windows Error Reporting (WER)

Not designed specifically for .NET code, does not report the values of local variables, and you need to have a Class 3 VeriSign certificate to sign-up and use the service.

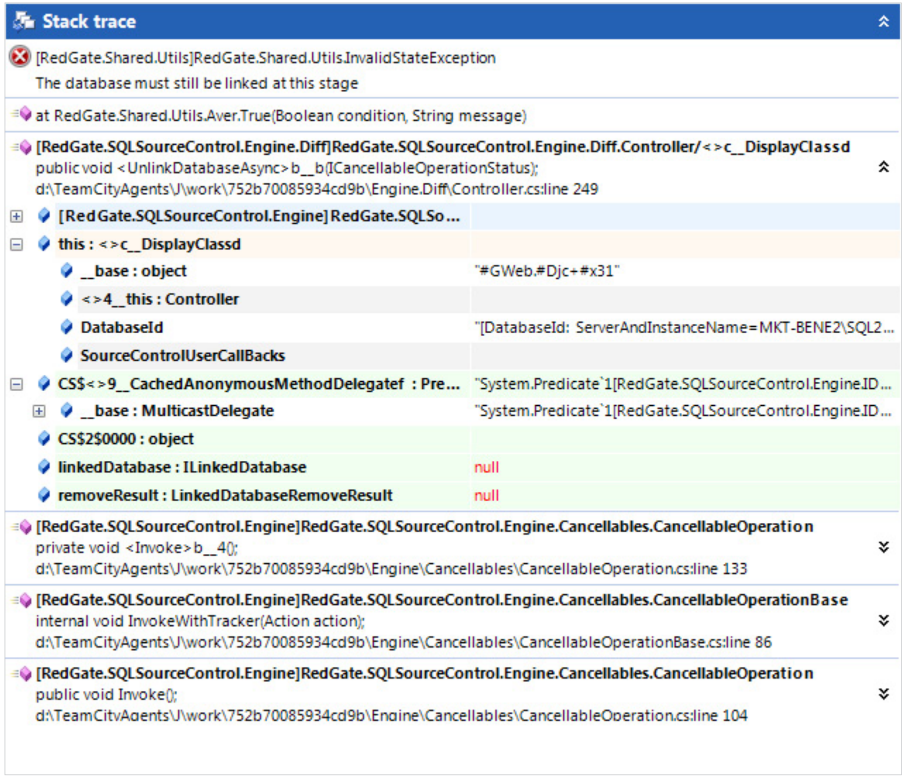
Other off-the-shelf solutions

Difficult or impossible to find a comprehensive solution. Typical problems are systems that are time-consuming to implement, don't have a fully featured viewer for prioritizing errors, and do not provide syncing to bug tracking systems. Many solutions don't provide local variable values, making it more difficult to debug and deliver fast fixes.

Do nothing

Bliss for a while, then the searing hangover.

Highly customizable



```
Stack trace
[RedGate.Shared.Utils]RedGate.Shared.Utils.InvalidStateException
The database must still be linked at this stage
at RedGate.Shared.Utils.Aver.True(Boolean condition, String message)
[RedGate.SQLSourceControl.Engine.Diff]RedGate.SQLSourceControl.Engine.Diff.Controller/<>c__DisplayClassd
public void <UnlinkDatabaseAsync>b__b(ICancellableOperationStatus);
d:\TeamCityAgents\J\work\752b70085934cd9b\Engine.Diff\Controller.cs:line 249
⊞ [RedGate.SQLSourceControl.Engine]RedGate.SQLSo...
⊟ this : <>c__DisplayClassd
    __base : object                                   "#GWeb.#Djc+#x31"
    <>4__this : Controller
    DatabaseId                                        "[DatabaseId: ServerAndInstanceName=MKT-BENE2\SQL2...
    SourceControlUserCallBacks
⊟ CS$<>9__CachedAnonymousMethodDelegatef : Pre...    "System.Predicate`1[RedGate.SQLSourceControl.Engine.ID...
  ⊞ __base : MulticastDelegate                        "System.Predicate`1[RedGate.SQLSourceControl.Engine.ID...
  CS$2$0000 : object
  linkedDatabase : ILinkedDatabase                    null
  removeResult : LinkedDatabaseRemoveResult           null
[RedGate.SQLSourceControl.Engine]RedGate.SQLSourceControl.Engine.Cancellables.CancellableOperation
private void <Invoke>b__4();
d:\TeamCityAgents\J\work\752b70085934cd9b\Engine\Cancellables\CancellableOperation.cs:line 133
[RedGate.SQLSourceControl.Engine]RedGate.SQLSourceControl.Engine.Cancellables.CancellableOperationBase
internal void InvokeWithTracker(Action action);
d:\TeamCityAgents\J\work\752b70085934cd9b\Engine\Cancellables\CancellableOperationBase.cs:line 86
[RedGate.SQLSourceControl.Engine]RedGate.SQLSourceControl.Engine.Cancellables.CancellableOperation
public void Invoke();
d:\TeamCityAgents\J\work\752b70085934cd9b\Engine\Cancellables\CancellableOperation.cs:line 104
```

A typical error report consists of a full stack trace and details about the exception context, including values of all the local variables.

The exception dialog can be customized to allow additional information to be packaged with the exception report. Customizations could include a log file or a screenshot taken at the time of the error, or asking end-users for contact information so you can notify them when a fix is released.

Five reasons to
Use Automated Error Reporting

1 It offers an efficient way to get customer feedback, massively reducing, or even eliminating, the to-and-fro between your customers and your developers.

2 It gives you insight into which bugs are the most severe or frequent, allowing you to prioritize bug fixes based on facts, not guesswork.

3 It enables you to deliver faster fixes for your customers and meet – **or even surpass** – their expectations.

4 You can lower support costs by identifying and fixing the issues your end-users are actually experiencing.

5 Before you release an application, you are able to gather lots of early user feedback. If you're using Agile development methodologies, Automated Error Reporting helps you go through rapid iterations based on customer feedback, allowing you to build – **and ship** – a robust application as quickly as possible.

SmartAssembly and JIRA working together

If you track bugs with JIRA, SmartAssembly Sync for JIRA can save even more time during the error-fixing process and provide a big boost for customer support.

SmartAssembly Sync for JIRA allows you to import SmartAssembly error reports into JIRA automatically. This means you can sort and manage all of your bugs from one central place: JIRA.

If end-users provide their email address when submitting the report, JIRA can be used to trigger an automatic email to the end-user who reported the bug. If there's already a workaround, JIRA notifies the user that there's a fix available and how to obtain it.

Combining best of breeds

"We relied on Atlassian's JIRA for providing us with a best-of-breed issue tracker, build server, and code repository browser, and now we have established an error reporting facility using Red Gate's SmartAssembly," says Mads H. Nissen, CTO of Puzzlepart.

"Our goal is to produce apps for the SharePoint platform. The software from these two vendors helps us speed up the process of creating high-quality business apps."

The SmartAssembly-JIRA integration provides the key link for a total solution. Better error reporting, faster fixes, and satisfied customers **– it's a nice proposition all around.**

> **"Knowing the frequency of problems (especially immediately after a release) is extremely helpful in prioritizing and triaging bugs that are reported internally. Additionally, having the context of where those errors occurred, including debugging information, really gives you that leap forward to start troubleshooting and diagnosing the issue."**
> Ed Blankenship, Microsoft MVP and Consultant

Automated Error Reporting resources

Automated Error Reporting in Red Gate's SmartAssembly software.

What's Automated Error Reporting and how does it work?

Harnessing the most powerful test rig in existence.

Capturing screenshots for Automated Error Reporting.

Customize Automated Error Reporting in SmartAssembly.

For me, SmartAssembly is all about remaining connected with your clients. If there is an error (as there inevitably will be) I can get in touch with them before they are driven to contact me. It's great for QA and they will love you for it.
Jonathan Evans, Vice President, Barclays Capital

The power of two: **Automated Error Reporting and obfuscation**

SmartAssembly combines Automated Error Reporting with obfuscation. An error reporting system that isn't aware of obfuscation is likely to be rendered useless by the obfuscation itself. Because SmartAssembly is both an obfuscator and error reporter, the obfuscation has no effect on the quality of error reports.