

# entwickler

Software, Systems & Development

magazin

CD-INHALT: Alle Infos auf Seite 3

www.entwickler-magazin.de

Juli/August

4.09

## Intrex Xtreme 4.5

Intranets, Portale und Webanwendungen einfach selbst erstellen



## Google Android SDK

Tools und APIs für Mobile-Developer

### Software

Subversion 1.6.2

Emacs 22.3

Windows Installer 4.5 SDK

GLScene

### Weitere Artikel

## ► Groovy-Workshop

Artikelverwaltung mit Groovy



## ► Tracing mit .NET

So minimieren Sie den Aufwand bei der Fehlersuche

Datenträger enthält Info- und Lehrprogramme gemäß §14 JuSchG

# Restart & Recovery



## Zuverlässigkeit für Ihre Anwendungen

## Elegant und mächtig

Reprint for fecher e.Kfm  
Translated from Entwickler Magazin 4.09

## Das leisten Schema-Sprachen bei der Prüfung von XML-Dokumenten

## Tools & IDEs für Sparfüchse

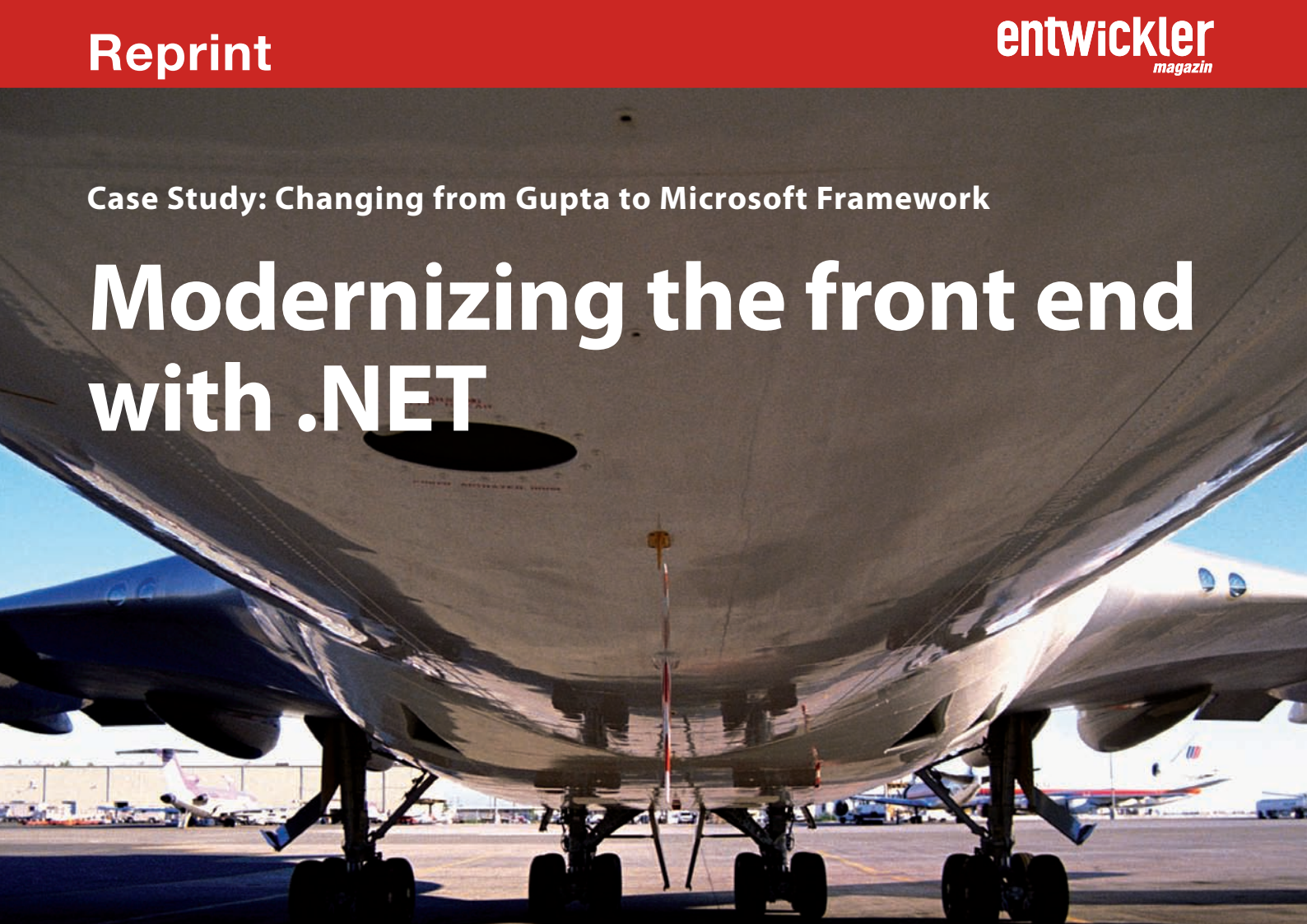
Günstiges Equipment für Entwickler

# Subversion 1.6

## Die neuen Features im Überblick

Case Study: Changing from Gupta to Microsoft Framework

# Modernizing the front end with .NET



Telco enterprises throughout the world are using the BSCS Customer Care & Billing Software of the LHS Group in Frankfurt to introduce new services or to bill on the basis of their tariff models. The client server solution, which has been offered since 1991, is operated not only by Web Clients but also via a Windows front end, which has been increasingly refined over the years and which in turn was implemented in the 4GL environment of Gupta. In order to open the user interface for modern presentation standards such as Microsoft Office Style or Windows Presentation Foundation, the company decided in 2007 to switch platforms to the .NET framework of Microsoft.

by **Günter Hofmann**

Telco enterprises throughout the world are using the BSCS Customer Care & Billing Software of the LHS Group in Frankfurt to introduce new services or to bill on the basis of their tariff models. The client server solution, which has been

offered since 1991, is operated not only by Web Clients but also via a Windows front end, which has been increasingly refined over the years and which in turn was implemented in the 4GL environment of Gupta. In order to open the user interface for modern presentation standards such as Microsoft Office Style or Windows Presentation Foundation, the company

decided in 2007 to switch platforms to the .NET framework of Microsoft.

A special feature of the BSCS solution was the numerous individual adaptations that it has undergone over the years for customers on all continents with quite different needs. Thus one of the decisive considerations in selection of the new platform was how simply such adaptations would be in

the future. Three alternatives for shifting the technology away from Gupta were available to LHS: Java, .NET and Web Clients. The argument for Java was that numerous Java developers were already on the in-house staff, and for .NET it was the better support by virtue of the large Microsoft community. In turn, Web Clients had the advantage that the application can be directly called up in the browser, without installation of software on the individual workstations of the users. The ultimate decision was for .NET, since it was amenable to automated porting of the SAL code of the Gupta application. For the other alternatives, it would have been necessary to develop completely new versions of the BSCS front end with its twelve individual modules in order to switch to the respective platform. The software specialists did not believe that it would be practical for the developers to spend considerable time restoring the status quo in the new environment while postponing their own further developments for years.

Concerning the choice of a suitable porting provider, the question of a reference visit was decisive. Only one of the consulting and software firms under consideration wanted the LHS team to speak with a user that had already successfully achieved porting (in this case the Fecher Co. of Rodgau). In a discussion with the customer in question, a private bank in Frankfurt, the developers convinced themselves that automated porting with the Ice Porter tool used for the purpose actually functioned in practice for complex projects.

### Limiting the migration risk

As a further decision aid for LHS, the enterprise was given the opportunity to analyze the application code in a test porting. In doing so, from March to April 2007, the porting specialists also determined the financial costs and the anticipated duration of the project. "Porting of the front end proved to be much less expensive than any new development," pronounced Jörn Hinrichsen, the LHS Project Manager. The esti-

mate of 100 man-days as the effort needed for porting was also reasonable. On the basis of its analysis, fecher recommended that some minor corrections and cleanup be done prior to porting, in order to avoid freezing the code any longer than absolutely necessary. The code slimmed down in this way would then run so much faster through the Ice Porter tool. In May 2007, LHS signed its agreement to the fixed-price bid and began preparation of the source code. When the parties met in July for the kick-off meeting, the project was divided into two phases. In the first phase, fecher was to port the six most critical BSCS modules immediately, after which the LHS developers could already resume further development, while fecher was taking care of the remaining six modules.

In total, around 650 windows with approximately 15,000 visual objects, 5,000 functions and 500 classes were converted into C# code, and also some embedded reports of the Gupta-internal Report Builder were ported to the more universal Crystal

## The three dependences of application developers

In selecting the application technology to be used, software developers always find themselves depending on the corresponding manufacturers. Three main areas can be distinguished (Fig. 1):

- \* The programming environment (programming language or 4GL) determines how the application logic and usually also the logic of the front end will be designed and further developed. Whereas proprietary techniques such as Gupta Team Developer, ADABAS/Natural or Visual Basic used to dominate in this area, .NET and Java as well as web standards such as PHP and Ajax are now being increasingly adopted. A platform switch usually means complete new development, and even automated migration offers are not successful without time-consuming manual follow-up work.
- \* In many applications the database not only is used for data storage but also contains important parts of the application code, for example in stored procedures or triggers. Such code is written in proprietary programming languages such as PL/SQL or T-SQL, and it impedes problem-free switching of the database platform. However, gaps in the SQL standard and corresponding manufacturer-specific expansions of instruction sets, syntax and data types also make porting from one platform to another a laborious undertaking.

- \* In addition to screen masks [= dialog boxes and windows], printed reports represent the most important "output" of applications. Although the generation of reports was still part of the formerly widespread 4GL tools, every developer must now make his own decision as to a reporting tool unless he wishes to program every list by hand. With the most widely used tools such as Crystal Reports, List & Label or Microsoft Reporting Services, even users can create their own forms, lists and analyses. In this way considerable effort is rapidly invested, but for lack of standards it cannot be transferred directly to a different reporting tool.

For these reasons software developers should not make light of the choice of framework, database and reporting platform. These represent long-range investments, and it is hardly possible any longer to reverse the wrong decision. If a switch cannot be avoided, migration specialists can help. Of course, they should be able to demonstrate adequate experience with both the source and destination platforms. Anyone who has doubts should not shy away from contacting cited reference customers and asking them extensively about their experiences.



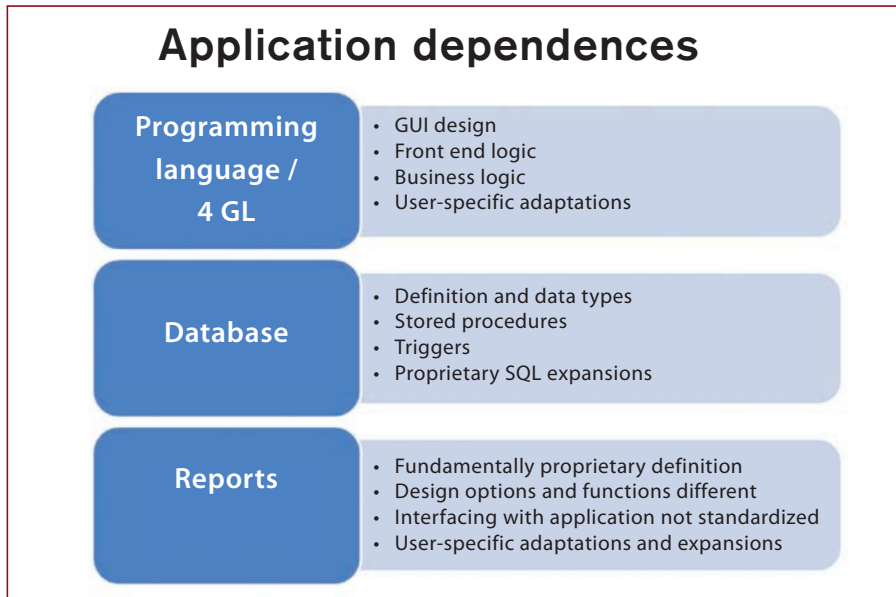


Fig. 1: Application dependences that must be considered in choosing a platform

Reports. Besides the enormous scope of the project, the consulting and software firm found itself faced with some further challenges. For example, it was necessary to ensure secure interfacing with an external Oracle database as well as support for the multilingual capacity of the application. Specifically, this capacity should be able to switch the dialog language dynamically and in the process adapt the layout of the screen dialogs automatically from right-reading to left-reading languages such as Arabic. Of course, the most imperative goal was to port all existing functionalities and make them run stably before new features could be developed. For example, remaining errors from the first porting phase had to be completely eliminated before delivery of the code to the developers and before the start of the second phase.

### Quality assurance by multistage test scenarios

All in all, the software tests accounted for the largest part of the total effort. The fecher team undertook the first round of tests before delivering the ported code to LHS. For this purpose, the function of the new .NET application was compared with recordings of previously prepared test cases of the old Gupta application. In a further round of tests, LHS itself ran extensive system tests to determine whether the new code

would be successful. Since the Gupta code was changed over one-to-one, the four staff members permanently assigned to the tests were able to compare the two applications with one another directly. At times, even as many as 15 developers participated in the tests, in order to satisfy the imposed quality demands. The developer team was able to eliminate most of the discovered errors on its own, but fecher still had to perform some manual touching up because of the large number of test cases. Nevertheless, the project management appeared to be more than satisfied with the quality of the code, since it was clear that an application of this magnitude cannot be switched overnight by pushing a button. The decisive factor was a fully functional C# code, which was able to achieve everything at least as well as the old Gupta application.

The project manager and his team took advantage of the waiting time during the first phase of the project to familiarize themselves with the .NET world. Although the Java-trained developers had gained little experience with C# up to that time, two one-week workshops organized by fecher and to some extent conducted by it were sufficient for the purpose. When the six main modules were finally accepted in October 2007, it became possible to begin modernization of the user interface

of BSCS in Visual Studio. For this purpose, a .NET expert was initially borrowed from fecher, in order to give the developers practical support in their first steps. In parallel, the second phase of the project was begun with porting of the remaining six modules. Now that all fundamental problems had already been solved, this proved, as expected, to be easier than before, and so porting was fully completed at the end of January 2008, after only six months.

### Goal achieved

In summary, by virtue of the switch to the .NET framework, a basis that would be secure for many years was successfully created for further development. In addition, because this framework is based purely on Microsoft technology, it will be possible to react more flexibly to future versions of Windows. At present, the LHS development team is optimizing the user interface on the basis of the most recent Windows standards.

*Günter Hofmann works as Head of Software Services for fecher e.Kfm.*

### Links & References

- [1] <http://www.fecher.eu/>
- [2] <http://www.lhsgroup.com/>

*good people  
good software*

**fecher. e. Kfm.**  
Seestraße 2-4  
63110 Rodgau  
fon: (06106) 605-0  
fax: (06106) 605-200  
[www.fecher.eu](http://www.fecher.eu)