

Borland® C++®

Case Study

Borland® C++Builder® supports Mexico's elections

Background: Improving the electoral process

In 1996, the Mexican government passed legislation designed to ensure a more democratic and transparent electoral process. One of the mandates was to create the autonomous Federal Electoral Institute (IFE), which was responsible for the organization of elections and for implementing a computerized system to calculate preliminary results. Officials named this initiative the Preliminary Electoral Results Program, or PREP.

The PREP development team was given quite a task: create an extremely secure and reliable system with enough redundancies that even a major interruption would allow it to continue operating, collecting, and publishing results promptly and accurately. Thereby, the system would allow the public access to real-time voting poll results—via the Internet—throughout election day.

In the 1997 elections for Mexico City's mayor and federal congressional members, the first such system was successfully executed. The decision was made to use the Internet to distribute the results, in real time, to the public and the national media—a worldwide first. The idea behind this was that the Internet had the ability to be the "grand auditor" by showing the exact location and results of the voting, in a short period of time. By publishing preliminary voting data quickly and widely, the IFE hoped to elude and avoid any potential attempt at deception.

After an exhaustive search, the IFE asked the Center for Research in Mathematics (CIMAT) to re-architect, develop, and test the Internet distribution system—with only about 15 days to spare before the elections. Thanks to the power afforded by Borland development tools—the Borland® Delphi™ development environment was used to generate Web pages with the ongoing voting information—CIMAT succeeded.

Challenge: Make the voting visible

Because its system for the 1997 elections was so successful, the IFE asked CIMAT to work on the systems for the 2000 presidential and congressional elections, which resulted in the election of Vicente Fox as President. The solution for 2000 was of even higher scope. These election results were to be published before a much wider audience by 26 different ISPs, several newspapers, and television networks, but it was unknown which platforms and technologies each would be

Borland®

“The organization and professionalism of the 2000 elections showed the world that Mexico has great respect for and commitment to its laws and its citizens. Borland technologies delivered the means to accomplish this task.”

*—Dr. Victor Guerra
Preliminary Electoral Results Program.*

utilizing. Because of this need for cross-platform integration and high performance, CIMAT decided that Java™ should be the programming language and that the Borland JBuilder® IDE, the leading Java development solution, was an ideal choice as it supports so many technologies and is so reliable. Borland C++Builder® was selected to develop and test the main application (before porting it to UNIX®) and as the back-end application, which monitored the distribution of the data from the individual voting polling locations to the Java-based application.

The solution: Start with C++, finish with Java™

The main focus of the programming team was on two key back-end processes: the validation, backup, and storage of preliminary results and the distribution of that information to the proper outlets. The first process accumulated voting data from balloting centers around the country with more than 3,000 POS keying terminals and 300 modems connecting in to transfer more than 300,000 certified electoral results. Using C++Builder, the development team quickly created an application that would run in conjunction with the BEA® Systems Tuxedo® transaction-processing engine to validate the data, store the data in an Oracle® database, and also replicate the data in a parallel site for redundancy and backup purposes.

Downstream from the database, the team again used C++Builder to construct an application that would conduct a SQL query every five minutes (to get the most updated election data), and then construct a set of TXT files. Those files were then transferred to a high-availability server that published the continuously updated—and verified— election results.

Mexican electoral law requires that a copy of the results be posted at each voting polling location. When the tallying was over, people could compare these results with the results that were being published by the ISPs, newspapers, and television networks via the application.

“The whole idea was that if we put the results on the Internet, many people could go see what was happening,” one team member says. By broadcasting the numbers to as many television, radio, and other media outlets as they could, the IFE would have multiple fallbacks in the event of computer hacking at any of the publishing locations.

“That was why we needed to send the results to so many locations,” says the team member, explaining the IFE’s “grand auditor” plan for using the Internet. “If a hacker modified the results or blocked access at one location, the others could be checked for the correct results.”

Case Study

To execute this plan on the network, the development team had to figure out a technical solution that would be secure, fast in operation, and have the ability to run on multiple types of host and client platforms. They found their answer in Java and used JBuilder to get them there.

The key to the information dispersal, the team said, was the development of a small Java servlet that the IFE could send out on frequent, regular intervals to all the broadcast points. The servlet, along with a few other necessary Java applets, would then unzip the encrypted text files sent with the voting results, and publish them to whatever Web server or other host software each broadcast point employed.

“One of the main reasons we wrote the programs as Java servlets was because we couldn’t depend on there being the same software or hardware at each [broadcast] location,” says the lead programmer. “We were sure that with servlets and text files, we could do the job anywhere.”

In creating the small servlet, the programmers added another layer of security to the voting process; instead of sending just the voting data every time there was new information, the IFE actually broadcasted the entire servlet application as well. That way, even if a hacker had managed to corrupt a broadcast site’s data, the correct data would be published at the next refresh, as the refresh would also contain a new version of the servlet.

Results: Credibility, for now and the future

According to CIMAT, the assorted Web sites supported by the IFE’s system received more than 50 million hits, many during the critical hours of the election—from 9 p.m. the night of voting until 1 or 2 a.m. the next day.

Each election year, the project team recommit itself to the same goal of improving the system’s efficiency, speed, and precision, while reducing its costs. Indeed, the team already is testing with Borland Kylix™ and with CORBA® technologies, to support a planned migration to Linux® running on PC platforms.

With the release of C++Builder 6 and its support for cross-platform development, team members have the freedom to choose between platforms without having to retrain themselves or rebuild their applications. The Borland CLX™ component library enables an application to be written for Windows® with either C++Builder or Borland Delphi and then recompiled to Linux with Kylix—all within the intuitive, high-productivity RAD environment Borland development tools provide.

“We like to create [applications] in a graphical way, and Borland tools make it quite easy to create, test, and debug,” says the team. “We don’t think we could do all that with traditional development tools for Linux.”

Case Study

Technology

Application	Voting tabulation, verification, and broadcast system
Tools	Borland® C++ Builder® and JBuilder®
Operating Systems	Windows® 2000, Linux®
Hardware platforms	Solaris™ (internal); various Web servers, including Apache™ for clients
Database server	Oracle®
Number of users	300+ voting tabulation sites (input); 30+ client information distribution sites (output)
Development team	4 developers

Made in Borland® Copyright © 2003 Borland Software Corporation. All rights reserved. All Borland brand and product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. All other marks are the property of their respective owners. Corporate Headquarters: 100 Enterprise Way, Scotts Valley, CA 95066-3249 • 831-431-1000 • www.borland.com • Offices in: Australia, Brazil, Canada, China, Czech Republic, Finland, France, Germany, Hong Kong, Hungary, India, Ireland, Italy, Japan, Korea, Mexico, the Netherlands, New Zealand, Russia, Singapore, Spain, Sweden, Taiwan, the United Kingdom, and the United States. • 13310.1