

# Borland®

## Borland Delphi: an OASIS for the U.S. Department of Labor

*"In only seven months, we created an imaging system that can process 5 million documents per year, amounting to three terabytes of data, with zero tolerance for data loss. A document associated with a worker's compensation claim can be retrieved for viewing within seconds of a request"*

—Pete van Helden, Project Manager of OASIS, U.S. Department of Labor.

*"One of the key factors in our success was that Delphi always came through"*

—Dick Maley, Programmer Advanced Delphi Systems.

### U.S. Department of Labor

The Office of Workers' Compensation Programs (OWCP) for the U.S. Department of Labor administers four major disability compensation programs that provide wage replacement benefits, medical treatment, vocational rehabilitation, and other benefits for certain workers who experience work-related injury or occupational disease. The Division of Federal Employees' Compensation (DFEC) is one of these programs, and it created an imaging system to more efficiently and securely manage the large number of documents required to process claims filed under this program. Given the range of functionality required, programmers for the department chose Borland® Delphi™ as the development environment for this project, which came to be known as OASIS (OWCP Automated System for Imaging Services). DFEC's standard development environment, Delphi allowed for far greater flexibility than that which could be achieved by using other development environments.

### Situation

Historically, the Division of Federal Employees' Compensation had handled millions of paper documents each year. Maintaining this system required large amounts of time, large amounts of space, left the division susceptible to document loss, and left the agency vulnerable in the event of fire or other catastrophic disaster.

The DFEC needed a system that could process 5 million documents annually, service 1,100 employees in 13 locations across the United States, and be completed in seven months within a budget of about \$20 million.

Once the DFEC determined that alternative development environments were inadequate for the proposed solution, the office programmers chose Borland Delphi for the development of six multifunctional, high-productivity client applications.

### Solution

The Delphi solution created an environment whereby callers can quickly get questions answered and where all data can be stored electronically, largely by making it possible to produce clients with capabilities that far exceed that which could be achieved in a purely thin-client paradigm.

When a worker or physician calls with a question, the proper document can be called up in seconds to provide an answer without even putting the caller on hold.

With the Delphi application, all case files can be backed up and stored in multiple locations, eliminating the need to save and store the hard-copy source material.

The accuracy of claim records has substantially improved with the new Delphi-derived solution, which provides for the rapid indexing and retrieval of documents.

Even though the imaging system built with Delphi was developed under a tight, seven-month deadline, users reported a virtually problem-free startup.

# Delphi™

case study

## The OWCP Situation

Until recently, claims records for the Federal Employees' Compensation Program were maintained in paper files containing all of the information needed to support a claim. As these files could grow quite large, locating them—much less individual documents within them—could be time-consuming. Sometimes, files had to be mailed across the country to different DFEC offices; if files in transit were lost or damaged, workers had to reconstruct them, which was a long and difficult process. Managers worried about any of a number of disasters striking an office and making it impossible to reconstruct the 150,000 or so files that conceivably could be destroyed. In addition, the DFEC was spending considerable money leasing storage facilities for all these documents.

DFEC had seven months to develop an imaging application for 1,100 users to manage documents. Congressional scrutiny was significant. The project included development of six clients, a middle-tier server, a document capture process, a database, a document storage system, and a system for synchronizing data among the 13 DFEC offices.

## The Borland Solution

The development team made full use of the Borland Delphi environment to create a unique solution that electronically automates the document storage and retrieval process.

The rapid prototyping capabilities of Delphi also made a big contribution by making it possible for users to provide input throughout its developmental duration. User contribution during this prototyping phase for the final system was one of the key factors in its ultimate success.

The Delphi team—an experienced group of expert developers—completed development for all clients and the server and messaging infrastructure within the specified seven-month time frame. Simultaneously, the systems of 1,100 users across the nation were upgraded to integrate with the OASIS system. Typically, after such a complex application is deployed, administrators field numerous reports of bugs discovered by users; thanks to exhaustive testing and consistent collection of user feedback throughout the project, the phone didn't even ring after the Delphi solution was implemented!

The new OASIS imaging system uses a middle-tier server based on IBM® WebSphere®, with business logic written in Enterprise JavaBeans™ (EJB™). Six distinct client applications were written in Delphi: an end-user application for viewing, indexing, and workload management; an administrative application for quality assurance; a kiosk application for viewing by visiting agencies; a bronzing application; an end-user training application; and an application that binds single-page TIF files into multiple-page TIF files.

Currently, case files are stored on hard disk drives, but eventually they will migrate to optical disks on jukeboxes as storage needs increase. Each of the 13 OWCP sites has its own message server and database and is synchronized using the IBM MQSeries® messaging system. The OWCP headquarters site, as well as an offsite backup location, stores all of the data, thus allowing hard-copy records to be discarded.

With the Delphi solution, new claim files are now scanned and indexed into the OASIS imaging system. Entire case files or only parts of a case file now can be called up within seconds using any of a number of identifiers. With the claim file, examiners can quickly find an individual document based on its category and subject (medical report/diagnostic test, document author date, date document received, etc.). The time savings are impressive: before the Borland Delphi solution was implemented, examiners had to physically search through these files, which can easily grow to include more than 1,000 pages. The Borland Delphi imaging system continues to lower administrative costs by conserving the time of highly paid skilled examiners and increasing the number of cases each is able to handle.

### Technology

Application	OASIS imaging system
Tool	Borland® Delphi™
Other Tools Evaluated	Java™
Database server	Informix™
Number of users	1100
Development Team Size	2 Delphi developers (16 developers total)
Development time	7 months

# Borland®

100 Enterprise Way  
Scotts Valley, California 95066-3249  
www.borland.com | 831-431-1000



Made in Borland®. Copyright © 2001 Borland Software Corporation. All Borland product names are trademarks or registered trademarks of Borland Software Corporation in the United States and other countries. Offices in: Australia, Brazil, Canada, France, Germany, Hong Kong, Japan, Latin American Headquarters in U.S.A., The Netherlands, Singapore, Taiwan, and United Kingdom. Java and all Java-based marks are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries. CORBA and ORB are trademarks or registered trademarks of Object Management Group, Inc. in the U.S. and other countries. • 12189