Storing Data in Amazon Cloud with Delphi

Paweł Głowacki

Embarcadero Technical Lead for Developer Tools

April 2016

# Executive Summary

This is a whitepaper on storing data in Amazon Simple Storage Service (S3) with Delphi.

# Table of Contents

[Executive Summary - 1 -](#_Toc449722857)

[Table of Contents - 2 -](#_Toc449722858)

[Introduction - 3 -](#_Toc449722859)

[Setting Up the Cloud Storage - 3 -](#_Toc449722860)

[Setup AWS account - 3 -](#_Toc449722861)

[Create S3 bucket - 3 -](#_Toc449722862)

[Setup Access Control - 6 -](#_Toc449722863)

[Building Multi-Device App - 13 -](#_Toc449722864)

[Setup Your Development Environment - 13 -](#_Toc449722865)

[Create New Delphi Multi-Device Project - 13 -](#_Toc449722866)

[Work with Amazon S3 in Delphi Code - 13 -](#_Toc449722867)

[References - 18 -](#_Toc449722868)

[About Author - 19 -](#_Toc449722869)

# Introduction

Applications need to store data. They can use local files, traditional local and remote SQL databases or specialized web services.

Sometimes we do not need the flexibility of the Structured Query Language, popularly known as SQL. Maybe we just want to store plain files.

In a desktop application it easy to access a remote database, but in mobile apps it is more typical to access data through online services.

In this article we are going to see how to use Amazon Web Services Simple Storage Service (AWS S3) in a multi-device Embarcadero Delphi application.

We are going to go through the steps of building a Delphi multi-device app that would allow to store and read arbitrary text strings in Amazon Simple Storage Service.

# Setting Up the Cloud Storage

## Setup AWS account

The first step is to make sure that we have an account in the Amazon Web Services. If you do not have an account, you can create one at <http://aws.amazon.com/free/>

In the first year you can use certain services for free to try them out. In case of AWS Simple Storage Service this allows for 5 GB of free storage and 20,000 Get and 2000 Put requests to be used for one year from creating an AWS account.

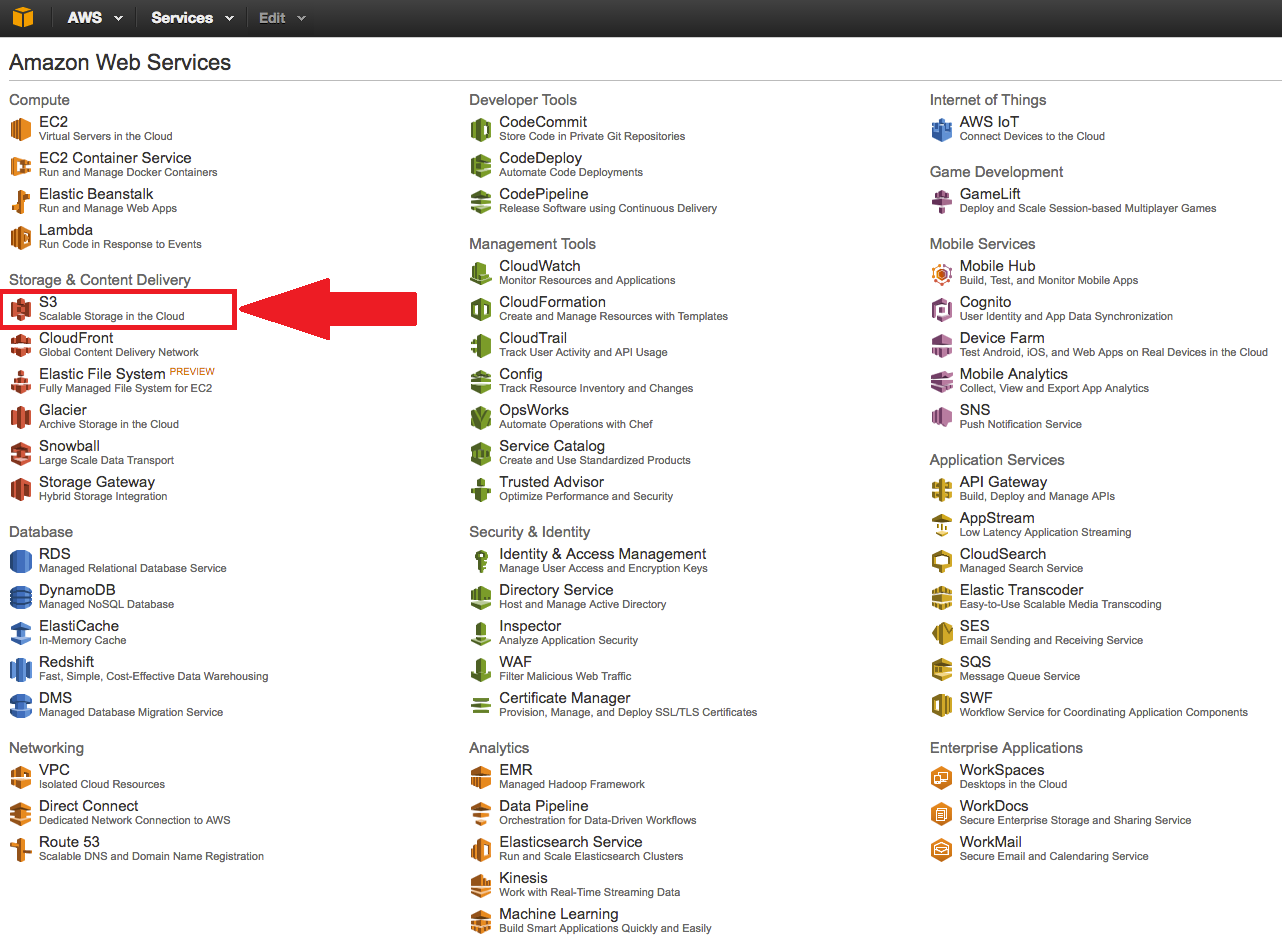
## Create S3 bucket

After signing up to the Amazon Web Services you have now access to AWS Management Console web page where you can manage different services.

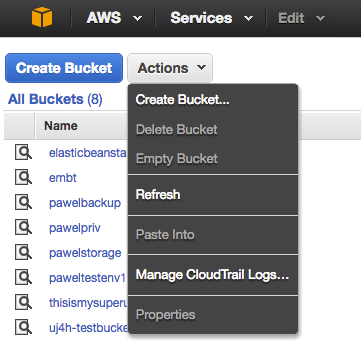
AWS Simple Storage Service let you store arbitrary amounts of data in a secure, reliable way. All data objects are organized into buckets. Single user can have up to 100 buckets, but the amount of objects stored in buckets is not limited. An individual object can have up to 5TB size. Within a bucket you can further organize files in folder structure. Every AWS object may be accessed via HTTP URL or programmatically from an app or a web site. Bucket names need to be unique and lower case.

More info on AWS S3 can be found in the online documentation at

<https://aws.amazon.com/documentation/s3/>

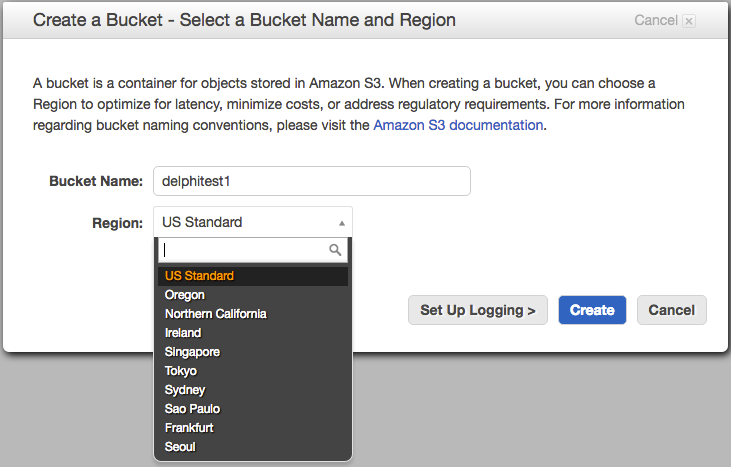


From AWS Management Console we can jump to the Simple Storage Service (S3) management page.

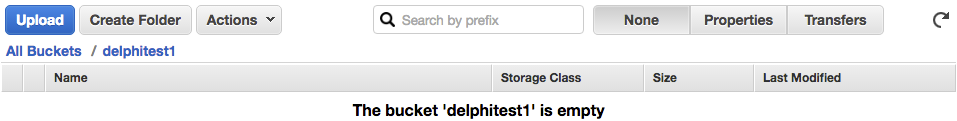


In the S3 page we can see the list of all existing buckets in our account and manage them.

Let’s create a new bucket called “delphitest1”in US-Standard AWS region.



Initially our bucket is empty. We can now upload files to the bucket and then access them either application code or via an URL.



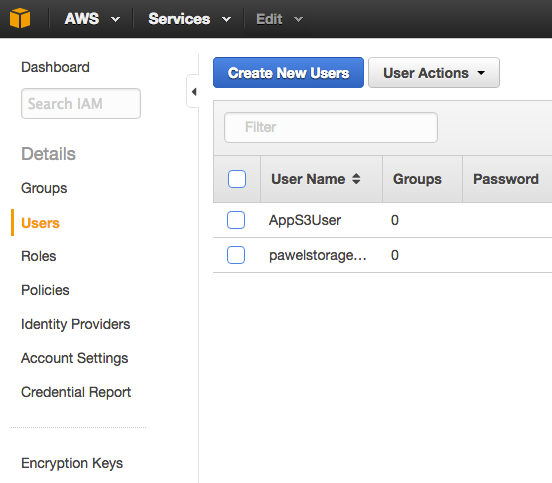
The next step is setup access control to this new bucket.

## Setup Access Control

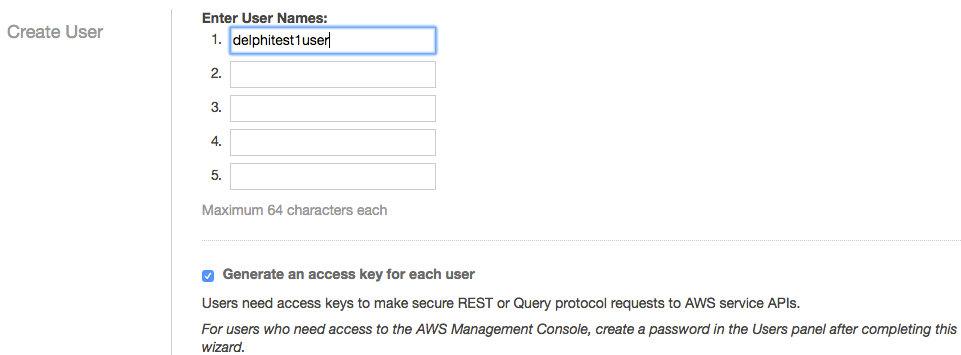
The multi-device app that we are going to build to interact with the “delphitest1” bucket in the AWS S3 is going to use RAD Studio Cloud API to make HTTP requests to AWS S3 APIs. This request needs to contain special security credentials in the form of account name and key.

We can use AWS “Identity and Access Management” (IAM) service to create a new user that has only access rights granted to work with a specific S3 bucket. This user will have access to a specific bucket inside of the AWS S3 without any additional rights.

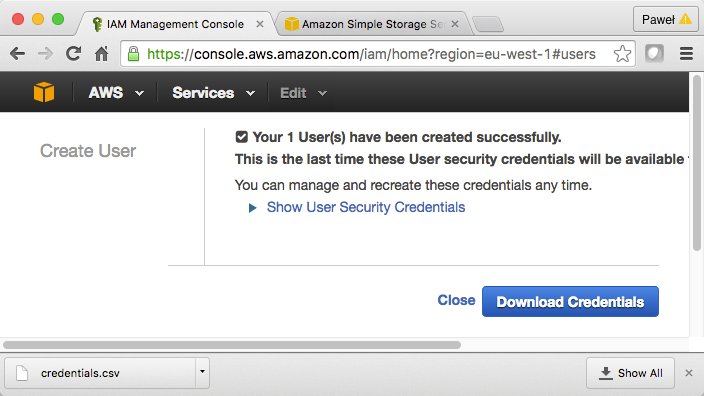
In the AWS Management Console navigate to “Identity and Access Management” page to create a new user.



In the “Create New Users” dialog it is possible to define up to five users at the same time.

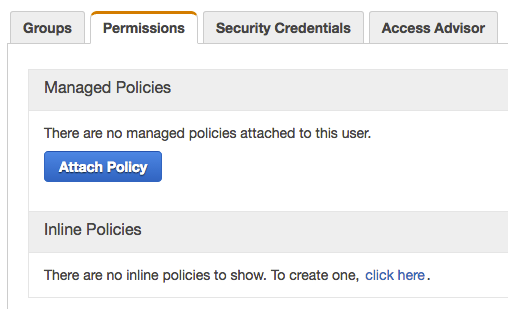


After a user is created there is the notification page shown with one-time opportunity to show on the web page or download new user access name and key to be used in web request to AWS APIs.

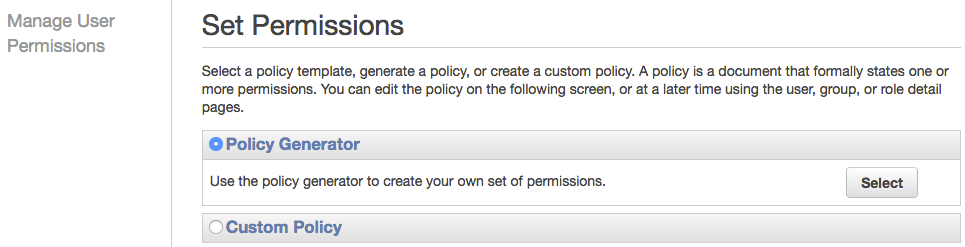


Make sure you save the downloaded “credentials.csv” file in a secure location.

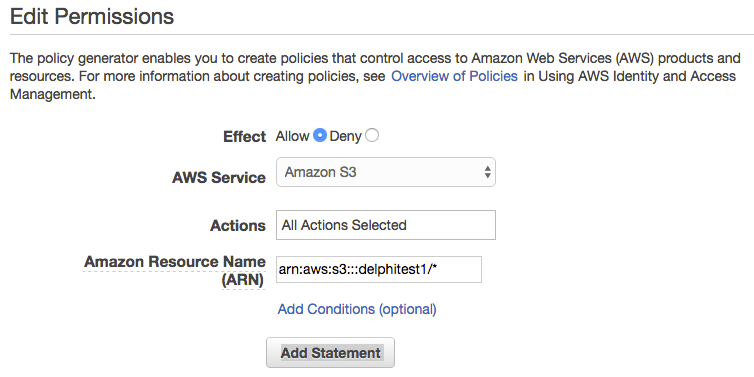
The next step is grant to the newly created user rights to access S3 bucket that we have created earlier. In the “Users” view on the IAM management page, click on “delphitest1user” and click on “click here” link to create an inline policy.



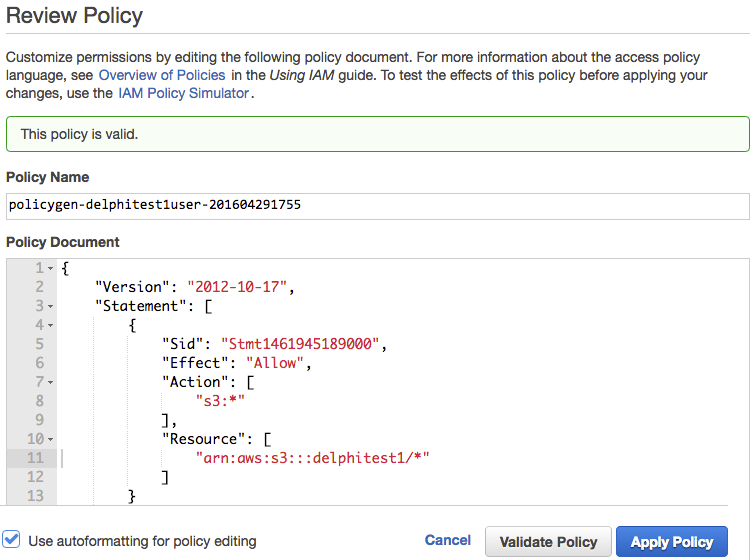
Click on “Select” button in the “Policy Generator” section.



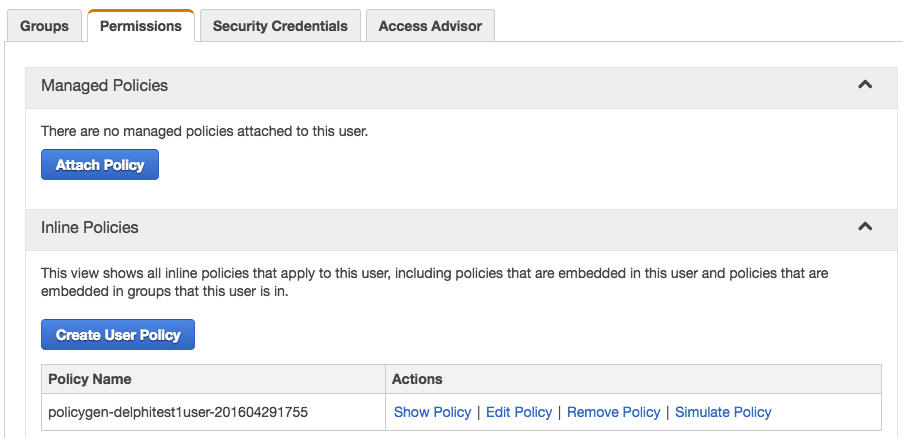
Make sure that “Allow” option is selected, choose “Amazon S3” from the list of services, check “All Actions” in the “Actions” drop down and enter “delphitest1” bucket in the format compatible with “Amazon Resource Notation” (ARN) based on AWS documentation (<http://docs.aws.amazon.com/AmazonS3/latest/dev/s3-arn-format.html>).



Click on “Add Statement” to create new S3 access policy to the IAM user. On the next screen we can review the raw JSON of generated policy, validate and apply it clicking on “Apply Policy” button.



The new policy is now listed as one of the inline policies of user “delphitest1”.



Now we can start working on the application that will be storing and reading data from the “delphitest1” AWS S3 bucket.

# Building Multi-Device App

## Setup Your Development Environment

Make sure that you have your development environment installed, configured and ready to be used. If you do not have Delphi installed, you can download it from <https://www.embarcadero.com/products/delphi>

If you are after mobile development, you will also need to follow up the configuration steps to be able to deploy your app to a mobile device. You can deploy to either Android or iOS device from within the IDE. Embarcadero Delphi IDE is a Windows program, so you will need to have Windows installed. If you intend to deploy to iOS, you also need a Mac machine, an account on <https://developer.apple.com/> and your device provisioned in XCode.

In Embarcadero Delphi you can build natively compiled apps from the same project source files for Android, iOS, Windows and Mac.

## Create New Delphi Multi-Device Project

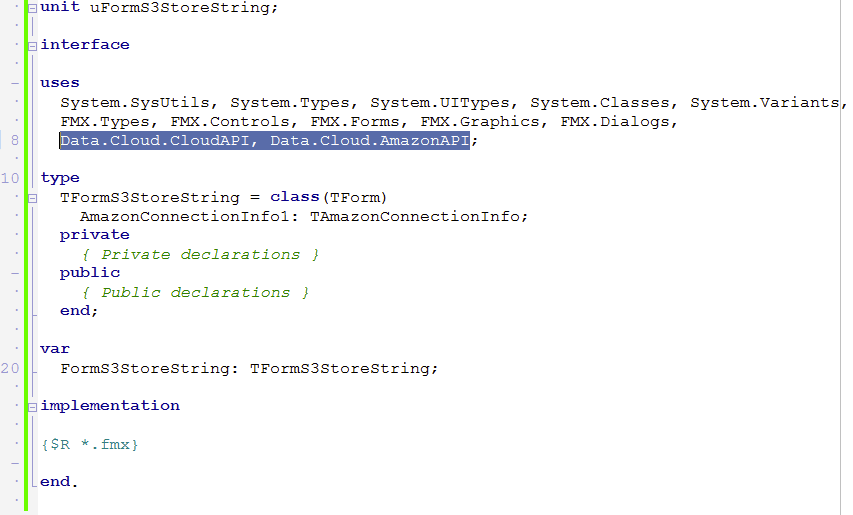
Select “File -> New -> Multi-Device Application – Delphi” from the main menu of Delphi IDE to create a new Delphi project. Click on “File -> Save All” and save the application main form and the project itself in a convenient location.

## Work with Amazon S3 in Delphi Code

Embarcadero Delphi contains a dedicated “Cloud” category in the Tool Palette where you can find two connection components. One for working with Amazon Web Services and for Microsoft Azure cloud services.

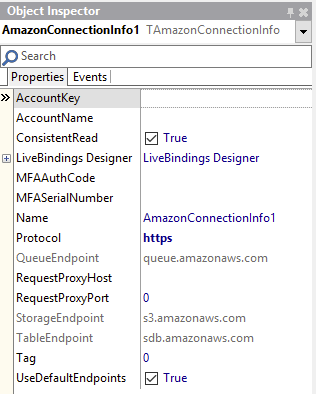
The functionality of the application will be very simple. When end user clicks on a button, then the contents of an edit box will be uploaded to Amazon Simple Storage Service to the bucket we have defined earlier. Clicking on the second button will download the string that will be displayed in a message box.

Double-click on the “TAmazonConnectionInfo” component in the Tool Pallette to add it to the form. Click on “Save All”. If you switch to Code Editor, you will realize that two units have been also added to form’s unit “uses” clause: “Data.Cloud.CloudAPI” and “Data.Cloud.AmazonAPI”.



***Note:*** *In production code, we would probably want to keep the Amazon connection component in a dedicated Data Module.*

Select connection info component in the Object Inspectore and enter downloaded AWS credential strings into “AccountKey” and “AccountName” properties.



You may also want to set “Protocol” property to “https”.

Drop an edit component on the form and enter some text into its “Text” property, for example “Hello from Delphi AWS S3 app!”.

Drop two buttons on the form and change their “Text” properties to respectively “Upload” and “Download”.

We are going to define two constants in the implementation section of our form’s unit. One for the bucket name and one for the object name.

**const**

BUCKET\_NAME = 'delphitest1';

OBJ\_NAME = 'testobj1';

Double-click on the “Upload” button and enter the following code in the body of the generated “OnClick” event handler.

**procedure** TFormS3StoreString.btnUploadClick(Sender: TObject);

**var** s3: TAmazonStorageService; strstr: TStringStream;

**begin**

s3 := TAmazonStorageService.Create(AmazonConnectionInfo1);

strstr := TStringStream.Create(Edit1.Text);

**try**

**if** s3.UploadObject(BUCKET\_NAME, OBJ\_NAME, strstr.bytes) **then**

ShowMessage('Uploaded OK')

**else**

ShowMessage('Upload ERROR');

**finally**

strstr.Free;

s3.Free;

**end**;

**end**;

Now double-click on the “Download” button and enter the following code.

**procedure** TFormS3StoreString.btnDownloadClick(Sender: TObject);

**var** s3: TAmazonStorageService; strstr: TStringStream;

**begin**

s3 := TAmazonStorageService.Create(AmazonConnectionInfo1);

strstr := TStringStream.Create();

**try**

**if** s3.GetObject(BUCKET\_NAME, OBJ\_NAME, strstr) **then**

ShowMessage(strstr.DataString)

**else**

ShowMessage('Failed to get object');

**finally**

strstr.Free;

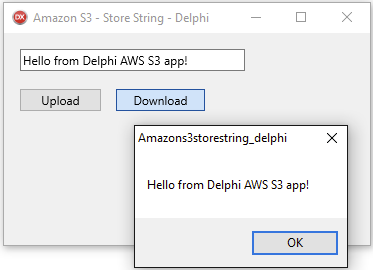
s3.Free;

**end**;

**end**;

Click on “Save All” and on “Run” to run the application.

Click on “Upload” button to upload the contents of the edit to the AWS Simple Storage Service bucket. Click on the “Download” button to verify that the same text is successfully downloaded and displayed in the message box!



That’s it! We have created a Delphi application that uploads and downloads data to and from Amazon Web Services Simple Storage Service (S3).

# References

* RAD Studio Home Page:   
  <http://www.embarcadero.com/products/rad-studio>

* C++Builder Home Page:   
  <http://www.embarcadero.com/products/cbuilder>
* Delphi Home Page:   
  <http://www.embarcadero.com/products/delphi>
* Amazon Web Services Free-Tier: <http://aws.amazon.com/free>
* Amazon Web Services Simple Storage Services (S3): <http://aws.amazon.com/s3>
* Amazon Web Services Identity and Access Management (IAM): <http://aws.amazon.com/iam>

# About Author



**Paweł Głowacki** is Embarcadero's European Technical Lead for Developer Tools. Previously, Paweł spent over 7 years working as a senior consultant and trainer for Delphi within Borland Education Services and CodeGear. As well as working with Embarcadero customers across the region, he also represents Embarcadero internationally as a conference and seminar speaker. For more information check out Paweł's technical blog at <http://blogs.embarcadero.com/pawelglowacki>



Embarcadero Technologies, Inc. is the leading provider of software tools that empower application developers and data management professionals to design, build, and run applications and databases more efficiently in heterogeneous IT environments. Over 90 of the Fortune 100 and an active community of more than three million users worldwide rely on Embarcadero’s award-winning products to optimize costs, streamline compliance, and accelerate development and innovation. Founded in 1993, Embarcadero is headquartered in San Francisco with offices located around the world. Embarcadero is online at [www.embarcadero.com](http://www.embarcadero.com/).