ntrocuction to **Appercept AWS SDK for Delphi**

Presented by Richard Hatherall



Appercept AWS SDK for Delphi General Features

- AWS HTTP client layer built on System.Net.HttpClient
- Compatible with all Delphi-supported target platforms
- Direct mapping of all features for implemented services
- Honours AWS configuration:
 - Shared files
 - Environment variables
 - Platform services



- Automatic credential resolution
- Automatic credential refreshing
- Scoped exceptions
- Automatic retries (based on policy)
- Convenient utility classes help you to:
 - perform complex or asynchronous actions
 - build complex data structures
- IDE-integrated AWS configuration manager
- Components for FMX and VCL



Appercept AWS SDK for Delphi Supported Services

- Application Integration
 - Amazon Simple Notification Service (SNS)
 - Amazon Simple Queue Service (SQS)
- Business Applications
 - Amazon Simple Email Service (SES)
- Cryptography & PKI
 - AWS Key Management Service (KMS)
- Machine Learning
 - Amazon Polly



- Amazon Textract
- Amazon Transcribe
- Amazon Translate
- Security, Identity & Compliance
 - Amazon Cognito
 - AWS Secrets Manager
 - AWS Security Token Service (STS)
- Storage
 - Amazon Simple Storage Service (S3)



AWS SDK Configuration First Launch

- Configuration wizard on first launch
- Configure the essentials to work with AWS



Appercept AWS SDK Setup

Welcome

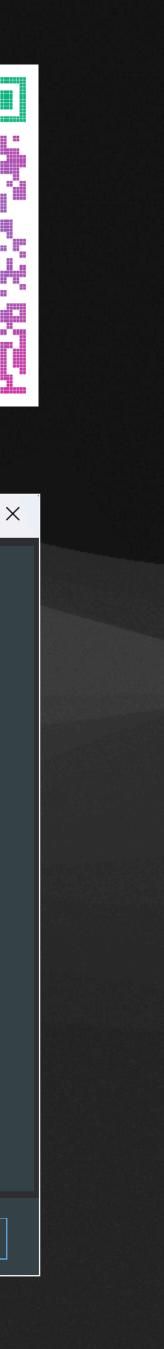
Configure the AWS SDK Register for updates (optional)

Welcome to the Appercept AWS SDK for Delphi

This wizard will guide you through setting up your AWS SDK and getting you started as smoothly as possible.

<u>C</u>ancel

<u>B</u>ack



AWS SDK Configuration IDE Integrated Configuration Manager

- Edit your AWS configuration profiles
- Choose an "IDE profile" to be used for debugging
- Stores configuration in your user profile `.aws` folder – where the official AWS SDKs load their configuration



	-		-	
5	<u>ר ו</u>	nt	\sim	na
	$\mathbf{\nabla}$	Dι	IU	112

✓ IDE

Default Folders

- Compiling and Running
 Component Toolbar
 Environment Variables
 File Association
- Desktop and Layout
- Welcome Page
- Project Upgrading
- LiveBindings
- Saving and Recovering
- Getlt Package Manager
- > User Interface
- > Editor
- > Language
- $\sim\,$ Third Party
 - TestInsight
 - Appercept AWS SDK
- > Version Control
- > Modeling
- > Deployment
- > Debugger

Appercept AWS SDK

	Profiles ☆ appercept		Properties					
				Кеу		Value		
	☆ default ★ demo							
	☆ nobuckets ☆ onlylogs					Remove	Add	
				Crede	entials			
				Static	Assume Role	e Advanced		
				Access	Key ID:			
				Secret /	Access Key:			0
				Session	n Token:			
	★ Default		Nev	N				
					Save	Cancel	Help	



Basic Service Usage Example

```
1 program ListS3Buckets;
 2
3 {$APPTYPE CONSOLE}
 4
5 {$R *.res}
 6
7 uses
     // 1. Add service unit to "uses" clause.
 8
    AWS.S3,
 9
    System.SysUtils;
10
11
12 var
    // 2. Define a reference to the service client interface.
13
    S3: IS3Client;
14
15
16 begin
17
     try
18
       // 3. Create a service client.
      S3 := TS3Client.Create;
19
20
21
      // 4. Make a request and store the response.
       var Response := S3.ListBuckets;
22
23
24
      // 5. Inspect the response.
       for var BucketName in Response.Buckets do
25
        Writeln(Format('%s %s', [
26
          FormatDateTime('yyyy-mm-dd hh:nn:ss', BucketName.CreationDate),
27
28
          BucketName.Name
29
        ]));
30
31
       Readln;
32
     except
33
      // 6. Don't forget to handle service exceptions.
34
35
      on E: ES3Exception do
        Writeln(E.ClassName, ': ', E.Message);
36
37
       on E: Exception do
38
        Writeln(E.ClassName, ': ', E.Message);
39
40
    end;
41 end.
```

🖾 C:\Users\rhatherall\Documen 🛛 🗡 $+ \vee$ 2023-07-25 11:04:50 appercept-demo-1 2023-07-25 11:05:14 appercept-demo-2 2023-07-25 11:05:31 appercept-demo-3

Requests Each action accepts arguments via a request object.

```
program Requests;
 3 {$APPTYPE CONSOLE}
 5 {$R *.res}
 6
  uses
     AWS.S3,
     System.SysUtils;
 9
10
11 var
     S3Client: IS3Client;
12
13
14 begin
15
     try
16
       // Create an S3 client.
17
       S3Client := TS3Client.Create;
18
19
       // Create a request object.
20
       var Request: IS3GetObjectRequest := TS3GetObjectRequest.Create(
21
       'a-bucket',
22
        'my-object'
23
       );
24
       // Call action with request object.
25
       S3Client.GetObject(Request);
26
27
       // Or use a convenient overload.
28
       S3Client.GetObject('a-bucket', 'my-object');
29
30
     except
31
       on E: Exception do
32
         Writeln(E.ClassName, ': ', E.Message);
33 end;
34 end.
```



Responses Each action returns a response.

```
program Responses;
 3 {$APPTYPE CONSOLE}
 5 {$R *.res}
 6
 7 uses
     AWS.S3,
 8
     System.SysUtils;
 9
10
11 var
12
     S3Client: IS3Client;
13
14 begin
15
     try
16
       // Create an S3 client.
       S3Client := TS3Client.Create;
17
18
19
       // Store the response.
20
       var Response := S3Client.GetObject('a-bucket', 'my-object');
21
22
       // Check if the response is successful.
23
       if Response.IsSuccessful then
24
       begin
25
        // Do something with the response.
26
       end;
27
28
     except
29
       on E: Exception do
         Writeln(E.ClassName, ': ', E.Message);
30
31 end;
32 end.
```



Exceptions On errors, actions raise exceptions scoped to the service client.

```
program Exceptions;
 3 {$APPTYPE CONSOLE}
 5 {$R *.res}
 7 uses
     AWS.S3,
     System.SysUtils;
10
11 var
     S3Client: IS3Client;
12
13
14 begin
15
     try
       // Create an S3 client.
16
17
       S3Client := TS3Client.Create;
18
19
       try
20
         var Response := S3Client.GetObject('a-bucket', 'my-object');
21
       except
22
         // A specific S3 exception.
23
         on E: ES3NoSuchKey do
24
           Writeln('Object not found!');
25
26
         // Any S3 excepion.
27
         on E: ES3Exception do
           Writeln(E.ClassName, ': ', E.Message);
28
29
       end;
30
31
     except
32
       on E: Exception do
33
         Writeln(E.ClassName, ': ', E.Message);
34 end;
35 end.
```



Retries Retries are automatic for "retryable" errors.

The behaviour can be altered through options.

```
program Retries;
 3 {$APPTYPE CONSOLE}
 5 {$R *.res}
 6
 7 uses
     AWS.S3,
 8
     System.SysUtils;
 9
10
11 var
     Options: IS30ptions;
12
     Client: IS3Client;
13
14
15 begin
16
     try
       Options := TS30ptions.Create;
17
18
       // Specify a retry limit.
19
20
       // Defaults to 3.
       Options.RetryLimit := 10;
21
22
       // Specify a duration in milliseconds as the base delay.
23
       // Defaults to 300.
24
       Options.RetryBaseDelay := 500; // Half a second.
25
26
27
       // Enable "jitter" with either "equal" or "full".
28
       /// Defaults to "none".
       Options.RetryJitter := 'full';
29
30
31
       Client := TS3Client.Create(Options);
32
       try
33
         var Response := Client.ListBuckets;
34
       except
         on E: ES3SlowDown do
35
           Writeln('Too many slow down responses!');
36
37
       end;
38
39
     except
    on E: Exception do
        Writeln(E.ClassName, ': ', E.Message);
41
42 end;
43 end.
```



Utilities Many utilities to assist you...

1 program BuildIAMPolicy; 2 3 {\$APPTYPE CONSOLE} 4 5 {\$R *.res} 6 7 uses AWS.IAM.PolicyDocument, 8 System.SysUtils; 9 10 11 begin 12 try Writeln(13 14 TIAMPolicyDocument.Build 15 .BeginStatement 16 .Sid('AllowListingContents') 17 .Action(['s3:ListBucketMultipartUploads', 18 's3:ListBucketVersions', 19 20 's3:ListBucket', 21 's3:ListMultipartUploadParts' 22 1) .Resource('arn:aws:s3:::*') 23 24 .EndStatement 25 .BeginStatement 26 .Sid('AllowListingBuckets') 27 .Action('s3:ListAllMyBucket') 28 .Resource('*') 29 .EndStatement 30 .Document 31 .ToString 32); 33 34 Readln; 35 except on E: Exception do 36 37 Writeln(E.ClassName, ': ', E.Message); 38 end; 39 end.

"Version": "2012-10-17", "Statement": [3 4 5 "Sid": "AllowListingContents", 6 "Action": ["s3:ListBucketMultipartUploads", "s3:ListBucketVersions", 8 9 "s3:ListBucket", 10 "s3:ListMultipartUploadParts" 11], 12 "Resource": [13 "arn:aws:s3:::*" 14 15 }, 16 "Sid": "AllowListingBuckets", 17 "Action": [18 19 "s3:ListAllMyBucket" 20], "Resource": "*" 21 22 23 24 }

1 {

2











ntrocuction to **Appercept AWS SDK for Delphi**

https://github.com/appercept/aws-sdk-delphi-samples https://www.appercept.com Support@appercept.com @ApperceptHQ

Presented by Richard Hatherall





