## 2. Building the sample EBP project for Azure

You can view an example of the steps required in video form here <u>https://youtu.be/ouCPWu4QkBs</u> and here <u>https://youtu.be/RmbbWhSpDZY</u>

Download and install Eclipse for J2EE from here : <u>http://www.eclipse.org/downloads/</u> Log into paas.heirloomcomputing.com and download our plugin, following instructions there for installation and license placement

Download Sample Source bundle from here: Open Eclipse



Right click the Project Explorer and select New->Project...:

Choose COBOL Project and click Next:

000	New Project		
Select a wizard Create a COBOL Project			\$
Wizards:			
type filter text			
<ul> <li>Java Project from Existing Ant</li> <li>Plug-in Project</li> <li>General</li> <li>Eclipse Modeling Framework</li> <li>EJB</li> <li>Elastic COBOL</li> <li>COBOL Project</li> <li>Elastic Transaction Platform</li> <li>Java</li> <li>Java EE</li> <li>JavaScript</li> <li>JAXB</li> <li>JPA</li> </ul>	Buildfile		
?	Next >	Cancel Finis	h

Name the project 'ListTable', select 'Default Project' under Project Type and click Finish:

000	New
COBOL Pro	ject
Create a ne	
Project na	me: ListTable
🗹 Use de	fault location
Location:	/Users/ianw/Documents/workspace-azure/ListTable Browse
	Choose file system: default \$
Defer	Project type Project directories
► Sampl ► Tutori Diagn	es als ostic
Details	
This p	roject template represents the default project setup.
?	< Back Next > Cancel Finish

Expand the ListTable project:



From your source bundle drag the ListTable.cbl file into the cobol\_source folder and choose to 'Copy files' and click OK:

000	File Operation
Select how files s	hould be imported into the project:
<ul> <li>Copy files</li> </ul>	
🔵 Link to file	s
🗹 Create	link locations relative to: PROJECT_LOC +
Configure Drag a	nd Drop Settings
?	Cancel OK

Expand the intermediate\_java\_code folder and note that Java has been generated from the COBOL:



Drag the file test.jcl into the workdir folder and sqljdbc41.jar into the workdir folder, again choosing to 'Copy files' :

![](_page_4_Picture_2.jpeg)

Right click the ListTable Project and choose properties, then click 'Java Build Path':

![](_page_5_Picture_0.jpeg)

Click 'Add JARS' and expand the ListTable project to the workdir, select sqljdbc41.jar and click OK:

O   O     JAR Selection	1
Choose the archives to be added to the build path:	
type filter text	
▼ 😕 ListTable	er
🕨 🧁 .executable	
cobol_source	
intermediate_java_code	
V Coverage workdir	
🐳 sqljdbc41.jar	
💽 test.jcl	L
X .classpath	ł
ecobol_project	-
Project     PeroteSystemsTempEiles	t
	d
	Γ
	ł
	ł
Cancel	Ī
	-

Click 'Java Compiler' , check 'Enable project specific settings' and choose 1.6 and the Compiler compliance level:

type filter text	Java Compiler	↓ ⇒ +
▶Resource Builders	✓ Enable project specific settings	Configure Workspace Setting
COBOL Compiler Settings COBOL Runtime Settings		
Java Build Path Use compliance from execution environment on the 'Java Build Path'		
Java Compiler	Compiler compliance level:	1.6 ‡
Java Editor Javadoc Location	✓ Use default compliance settings	
Project Facets Project References	Generated .class files compatibility:	1.6 ‡
Refactoring History	Source compatibility:	1.6 ‡
Task Repository	Disallow identifiers called 'assert':	Error \$
Task Tags ▶Validation	Disallow identifiers called 'enum':	Error \$
	<ul> <li>Add source file name to generated class file (us</li> <li>Preserve unused (never read) local variables</li> <li>Inline finally blocks (larger class files, but impro</li> <li>Store information about method parameters (us</li> <li>When selecting 1.6 compliance, make sure to ha (currently 1.8). Configure the <u>'Installed JREs'</u> and on the <u>'Java Build Path'</u>.</li> </ul>	ed by the debugger) oved performance) sable via reflection) ave a compatible JRE installed and activated l' <u>Execution Environments</u> ', or change the JRE
		Restore Defaults Apply

Click OK, then click Yes to perform a project rebuild.

Open the cobol\_source folder , double click ListTable.cbl to bring up the source code. Edit the jdbc driver information with the jdbc connection string you copied from the Azure SQL database properties. Remember to add in the user and password:

000	🕒 workspace-azure - Java EE - ListTable/cobol_source/ListTable.cbl - Eclipse
🖆 • 🖩 🕼 📮 🗽 🕨 💷 📾 🗱 🖧 👼 🧝	◎♥◎♥ ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ● ●
Project Explorer 🛛 📄 🔄 🗢 🗖 🗖	📓 ListTable.cbl 🕱 🗖 🗖
<ul> <li>▼ Collisitation</li> <li>Collisitation</li> <li>Colli</li></ul>	1       1       2       3       4       5       6       7       8         1       \$set sourceformat "free"       2       IDENTIFICATION DIVISION.         3       PROGRAM-ID. ListTable.         4       ENVIRONMENT DIVISION.         5       INPUT-OUTPUT SECTION.         6       DATA DIVISION.         7       WORKING-STORAGE SECTION.         8       01 emp-group-item.         9       05 uname pic x(16).         10       05 phone PIC 9(9).         11       01 cursor-switch pic x value 'n'.         12       88 end-of-cursor value 'y'.         13       EXEC SQL INCLUDE SQLCA END-EXEC.         15       16         17       MAIN SECTION 01.
∰ sqljdbc41.jar	18⊕ BEGIN. 19 20 EXEC SQL CONNECT TO 21 'jdbc:sqlserver://onldemosrv.database.windows.net:1433;database-onldemo;user-ianw@o 22 23 DRIVER 'com.microsoft.sqlserver.jdbc.SQLServerDriver' 24 END-EXEC. 25 26 IF SQLCODE NOT = 0 27 DISPLAY "FAILED TO CONNECT TO DATABASE " SQLCODE " " SQLERRM 28 UPON SYSERR 29 GO TO FIN 30 END-IF. 31 32 EXEC SQL declare csr1 cursor for

Save ListTable.cbl. Right click the source code and choose 'Run As->1 COBOL Application' :

![](_page_9_Picture_0.jpeg)

If you have set up the database and table you should see a SQL Code 100 in the console window:

![](_page_9_Picture_2.jpeg)

This confirms the program will work correctly after it is exported and run under EBP.

To export this application so it is ready to deploy to EBP later on, right click the ListTable project and choose 'Export':

	Export	
elect		-7-
Package Elastic Co	OBOL project resources for deployment	25
Select an export v	wizard:	
type filter text		
🕨 🧁 General		
🕨 🧁 EJB		
🔻 🗁 Elastic COE	BOL	
🧧 Elastic C	COBOL Deploy Wizard	
😰 Elastic T	Transaction Platform Deploy Wizard	
🕨 🗁 Install		
🕨 🗁 Java		
🕨 🗁 Java EE		
🕨 🗁 Plug-in De	velopment	
Remote Sy	stems	
Run/Debug	9	
Tasks		
Team		
Web		
Web Servic	es.	

Choose Elastic COBOL Deploy Wizard and click Next:

000	COBOL Deploy Wizard
Archive Package Specification Define what resources to package	, ē
Resources to export:	
[ListTable] listtable\$Wrk.class [ListTable] listtable.class [ListTable] listtable.java.smap	Add Remove
Deployment Options	~
• Traditional O Cloud	
Select the export destination:	
JAR or WAR file: /Users/ianv	//ListTable.jar ▼ Browse
Specify the deployment license	e location:
	Browse
Options: Compress the contents of the arc Overwrite existing files without w	nive arning
?	< Back Next > Cancel Finish

File in the name of the file to export (seen here is /Users/ianw/ListTable.jar) and click 'Add..' at the top to add the SQL driver to the export. Open the workspace/project and workdir folder and select the sqljdbc41.jar, then click OK:

Plea	se select resources to add to the list	
	<ul> <li>Workspace</li> <li>ListTable</li> <li>executable</li> <li>cobol_source</li> <li>copylib</li> <li>intermediate_java_code</li> <li>java_source</li> <li>listing</li> <li>resources</li> <li>workdir</li> <li>RemoteSystemsTempFiles</li> </ul>	<ul> <li>✓ math sqljdbc41.jar</li> <li>□ math sqljdbc41.jar</li> <li>□ math sqljdbc41.jar</li> </ul>
?	)	Cancel OK

This will add the driver to the export list:

Resources to export:	
[ListTable] listtable\$Wrk.class [ListTable] listtable.class	Add
[ListTable] listtable.java.smap	Remove
[ListTable] Workdir/Sqljdbc41.jar	
	T

Click Next:

000 cc	BOL Deploy Wizard
Archive Packaging Options	
Define required runtime elements and ot	her options
Select runtime elements required by app	lication:
Elastic COBOL Runtime Elements	<ul> <li>Core Runtime Elements (Required</li> <li>Printing API</li> <li>File Protocol Support</li> <li>Graphical Screen Section Support</li> <li>JCurses Support (Platform Depend</li> <li>XML Support</li> <li>Environment Support</li> </ul>
${box{ }}$ Include the selected runtime elem	ents within the export archive
Select the settings to use for launching th	e application:
Run Configuration: ListTable.cbl	\$
Select the class of the application entry pe	pint:
Main Class: listtable	Browse
Other Options (JAR only): Copy JAR dependencies from the E Generate script-based application Generate HTML-based application Generate WAR package from	lastic COBOL directory to the export directory launcher stub launcher stub n JAR and HTML stub
Use Java plugin	
? < B	ack Next > Cancel Finish

Click Finish, the jar file will be created. Minimize the ListTable project and close the ListTable.cbl source file:

![](_page_14_Picture_2.jpeg)