

A Guide To The CBSA GitHub Folders And Their Contents.

CBSA (CICS Banking Sample Application) is a banking application developed in a mixture of COBOL, Java and Springboot. It is derived from its parent application Hursley bank. Hursley Bank has been used to test CICS TS over many years.

CBSA is a smaller more easily consumable version. CBSA started life as part of project Nazare, to showcase IBM products in a CI/CD pipeline environment. Since then, more and more teams have requested access to it as they seek to utilise it as part of their testing.

This latest version (CBSA V3) includes the addition of a restful API which can be driven via a z/OS Connect EE server. The APIs themselves can be driven natively or via the new Payments and Customer Services interfaces. These interfaces, written in Springboot, are supplied with source code so, should you wish to, you can build upon them and tailor them to meet your own needs.

All functionality provided previously in CBSA should still function. However, CBSA V3 replaces all previous versions and any repos containing previous versions of CBSA should be fully replaced with the current V3 folders and code.

CBSA V2 and all functionality relating to 'enacting Scenario 2' are supported in CBSA V3. For information: Scenario 2 was an exercise to make a change to the CBSA application (to increase the account number from 8 bytes in length to 9 bytes). However, it should be noted that 'enacting Scenario 2' was a one-off exercise specifically for project Nazare. Once the exercise has been completed the database and supporting files will all have a 9 byte account number. After completion, users will need to reinstall CBSA to revert the database and all files back to an 8 byte account number. No migration path between CBSA V2 and CBSA V3 is provided.

The CBSA repo folders are structured as follows:

/aarfiles – contains the aar (application archive) files that the zOS Connect EE server uses to implement the new Payment and Customer Services APIs.

/application-conf – contains property files used by DBB (DBB is used when enacting scenario 2/CBSA V2 to recompile changed source code. If using Nazare Scenario 2 in the provided zTrial environment, no further set up should be required and this information is for information only. If you need to get DBB working with CBSA outside of the Nazare zTrial environment, please refer to the separate DBB Installation Instructions in the **/doc** folder).

/asm – contains a single assembler module

/automation – contains Ansible automation installation scripts.

/bms – this folder has multiple uses:

- In Scenario 2/CBSA V2, the folder contains the BMS maps used by CBSA at the start of the enactment exercise. During the exercise some BMS will be amended and then this folder becomes the working directory for the BMS source code changes.
- In CBSA V3 the folder contains the supplied BMS maps for CBSA.

/bms_after – this folder is specifically for use by Scenario 2/CBSA V2. It contains the amended BMS maps i.e. all of the revised BMS map changes for Scenario 2 (after enactment). This is provided so that enacting Scenario 2 can be achieved quickly and easily.

/bms_before – this folder contains all of the BMS maps for CBSA V2. It is the starting point for all BMS maps used in Scenario 2/CBSA V2. It should have the same content as the **/bms** folder at installation time. However, the **/bms** folder content will change as Scenario 2 is enacted.

Should there be any kind of error when enacting Scenario 2, the user can restore all BMS maps back to a working state by simply copying the content of **/bms_before** into **/bms** (and recompiling the affected modules).

/buildjcl – this folder contains all of the JCL used by the underlying install jobs to compile, bind and link edit every COBOL program, and assemble every BMS map.

/buildOutput – this folder contains abend procedure information.

/cobol – this folder has multiple uses:

- In Scenario 2/CBSA V2, the folder contains the cobol source code used by CBSA at the start of the enactment exercise. During the exercise some cobol programs will be amended and then this folder becomes the working directory for the cobol source code changes.
- In CBSA V3 the folder contains the supplied cobol program source code for CBSA.

/cobol_after – this folder is specifically for use by Scenario 2/CBSA V2. It contains the amended cobol programs i.e. all of the revised cobol program source code for Scenario 2 (after enactment). This is provided so that enacting Scenario 2 can be achieved quickly and easily.

/cobol_before – this folder contains all of the cobol programs for CBSA V2. It is the starting point for all cobol program source code used in Scenario 2/CBSA V2. It should have exactly the same content as the **/cobol** folder at installation time. However, the **/cobol** folder content will change as Scenario 2 is enacted.

Should there be any kind of error when enacting Scenario 2, the user can restore all cobol programs back to a working state by simply copying the content of **/cobol_before** into **/cobol** (and recompiling the affected modules).

/com.ibm.cics.cip.bank.LibertyAPI.webUI - this folder has multiple uses:

- In Scenario 2/CBSA V2, the folder contains the java source code for the Liberty web UI used by CBSA at the start of the enactment exercise. During the exercise some java source code will be amended and then this folder becomes the working directory for the java source code changes.
- In CBSA V3 the folder contains the java source code for the Liberty web UI for CBSA.

/com.ibm.cics.cip.bank.LibertyAPI.webUI-AFTER - this folder is specifically for use by Scenario 2/CBSA V2. It contains the amended java source code i.e. all of the revised java source code for Scenario 2 (after enactment). This is provided so that enacting Scenario 2 can be achieved quickly and easily.

/com.ibm.cics.cip.bank.LibertyAPI.webUI-BEFORE - this folder contains all of the java source code for CBSA V2. It is the starting point for all java program source code used in Scenario 2/CBSA V2. It should have exactly the same content as the **/com.ibm.cics.cip.bank.LibertyAPI.webUI** folder at installation time. However the **/com.ibm.cics.cip.bank.LibertyAPI.webUI** folder will change as Scenario 2 is enacted.

Should there be any kind of error when enacting Scenario 2, the user can restore all java source code back to a working state by simply copying the content of **/com.ibm.cics.cip.bank.LibertyAPI.webUI-BEFORE** into **/com.ibm.cics.cip.bank.LibertyAPI.webUI** (and recompiling the affected modules).

/copylib – this folder has multiple uses:

- In Scenario 2/CBSA V2, the folder contains the copybook members used by / pulled into the CBSA source code at the start of the enactment exercise. During the exercise some copybook members will be amended and then this folder becomes the working directory for the copybook changes. The content of this folder is used by DBB.

- In CBSA V3 the folder contains the supplied copybook members used by / pulled into by the CBSA source code. The content of this folder can be used by DBB.

/copylib_after – this folder is specifically for use by DBB in Scenario 2/CBSA V2. It contains the amended copybooks i.e. all of the revised copybooks for Scenario 2 (after enactment). This is provided so that enacting Scenario 2 can be achieved quickly and easily.

/copylib_before – this folder contains all of the copybooks used by DBB for CBSA V2. It is the starting point for all copybooks used in Scenario 2/CBSA V2. It should have exactly the same content as the **/copylib** folder at installation time. However, the **/copylib** folder content will change as Scenario 2 is enacted.

Should there be any kind of error when enacting Scenario 2, the user can restore all copybooks back to a working state by simply copying the content of **/copylib_before** into **/copylib** (and recompiling the affected modules).

/Db2_jcl_install – this folder contains all of the jobs/JCL to create all of the Db2 artifacts used by CBSA.

/doc – this folder contains CBSA documentation

/dsect_before – this folder contains all of the copybooks used by CBSA JCL installation jobs for both CBSA V2 and CBSA V3.

/install – This is an Admin only folder, which can be ignored.

/jclInstall – This folder contains jobs to aid installation of CBSA for V2 and V3 (please refer to the CBSA Installation guide for more information).

/linkedit – This folder contains all of the linkedit entries used during compilation.

/reorg – This folder contains jobs that get executed as part of the enacting Scenario 2 exercise/CBSA V2.

/runtime_jars – This folder contains all of the run time jars for CBSA V2 and V3.

/sarfiles – contains the sar (Service archive) files that the zOS Connect EE server uses to implement the new Payment and Customer Services APIs in CBSA V3.

/warfiles – contains the warfiles used in CBSA V2 and V3.

/zosconnect_artefacts – This folder is new in CBSA V3 (and isn't used at all in CBSA V2). It contains two sub folders called **/apis** and **/services**.

These two sub folders contain z/OS Connect EE project artefacts. Utilising zOS Connect EE project artefacts is not necessary when using the Payment or Customer Services interfaces or using the restful APIs standalone (the aars and sars are sufficient for that). However, should you ever wish to amend an API (for example, to change the information being returned) or its associated Service, having the zos connect project artefacts available, means you can utilise z/OS Connect EE's built in API and Service editors to make any changes.

{If these are required; you can import the artefacts within the z/OS Connect EE Project Explorer view, simply right click and select to "Import" and select "import" from the list. Then under "General" click on the "Existing Projects into Workspace" and click Next. Locate the **/zosconnect_artefacts/apis** or **/zosconnect_artefacts/services** folder in the CBSA repo and check the "Copy projects into Workspace" option, and the artefacts will be pulled in to z/OS Connect EE. From there you can open the artefacts with zOS Connect EE's editors}.

/Z-OS-Connect-EE-Customer-Services-Interface – This folder is new in CBSA V3 (and isn't used at all in CBSA V2). It contains all of the artefacts for the Customer Services interface.

/Z-OS-Connect-EE-Payment-Interface – This folder is new in CBSA V3 (and isn't used at all in CBSA V2). It contains all of the artefacts for the Payment interface.

/zoseeserver – This folder contains the zOS Connect EE server.xml. Please refer to the CBSA Installation Guide for more information.

