

# **CICS<sup>®</sup> TS support for the WebSphere<sup>®</sup> XD Compute Grid**

*CICS SupportPac CN11 - Installation and User's Guide*

*Version 1.1 – December 2011*

**Authors:**

Daniel Millwood – IBM CICS Services and Technologies, CICS Development

Mark Cocker – IBM CICS Technical Strategy and Planning, CICS Development

IBM United Kingdom Ltd.

Hursley Park. Winchester. SO21 2JN. UK

---

# Contents

Chapter 1. Overview.....	3
Traditional z/OS batch applications and CICS.....	3
SupportPac CN11.....	3
Software requirements.....	4
Restrictions.....	4
Chapter 2. Installing the SupportPac.....	5
Downloading and copying the SupportPac files.....	5
Create the PDS datasets.....	6
Creating the sample VSAM files.....	6
Configuring the JVM server profile.....	7
Adding resources in the CICS CSD.....	7
Defining the tables in DB2.....	8
Gathering WebSphere Application Server information.....	8
Configuring the SupportPac.....	9
Updating CICS.....	10
Chapter 3. Getting started.....	11
Starting the SupportPac.....	11
Running the SupportPac sample.....	11
Understanding the xJCL.....	12
Understanding the CatalogManagerStockUpdate job step.....	13
Batching.....	14
Locking.....	14
Stopping the SupportPac.....	14
Troubleshooting.....	15
Feedback.....	15
Chapter 4. References.....	17
Chapter 5. Notices.....	18
License.....	18
Trademarks.....	18

---

## Chapter 1. Overview

IBM® WebSphere® Extended Deployment (XD) Compute Grid provides services for job scheduling and management, execution control, and monitoring of Java transactional batch, compute-intensive and non-Java workloads. WebSphere XD Compute Grid can schedule work in JEE and JSE runtimes, including WebSphere and stand-alone Java environments on z/OS and other platforms.

This SupportPac provides the infrastructure for WebSphere XD Compute Grid to:

- Schedule Java job steps in CICS
- Provide programming interfaces to process input and output files
- Checkpoint
- Restart

For more information about WebSphere XD Compute Grid refer to <http://www.ibm.com/software/webservers/appserv/extend/computegrid/>.

---

## Traditional z/OS batch applications and CICS

Batch processing is traditionally used for reconciling and reporting transactions at the end of the day, week, month, quarter, and so on. It is used in many business activities from payments processing to billing, stock control, credit scoring, interest calculations, and more.

Batch processes sometimes require exclusive access to resources, which makes them more efficient to run, and they typically share business logic. Batch processing is usually accomplished through automated processes that remove resources in CICS, run the batch process, and restore the resources in CICS. This batch window, as it is frequently called, can be several hours long during which time the CICS applications are not available or are only partially available which can be disruptive to business operations.

WebSphere XD Compute Grid helps to resolve this problem by running batch and CICS applications simultaneously. With Compute Grid, CICS resources and business logic are shared. Compute Grid provides the following features:

- Dispatching, managing, run control, and monitoring facilities for jobs. In addition, Compute Grid works with other schedulers such as the IBM® Tivoli® Workload Scheduler.
- Higher throughput to address larger volumes of data by processing jobs in parallel across multiple CICS regions.
- Data locking, where updates are synchronized at configurable checkpoints and checkpoints include positions in input and output resources.
- Failure and recovery scenarios, where you can roll back updates, restore the last checkpoint, restore cursors to input and output resources, and retry a job step.

---

## SupportPac CN11

The SupportPac includes a sample batch application called CatalogManagerStockUpdate that can be used in conjunction with the CICS Catalog Manager sample.

The envisioned scenario is that when new stock arrives, a batch program is run that updates stock levels in the Catalog Manager stock file. As the Catalog Manager application needs to be available 24x7, this update needs to be performed while online processing continues. Therefore, the choice is made to create the batch application using the CICS support for the Compute Grid batch programming model. This allows the batch program to run alongside the existing online Catalog Manager application.

In the scenario, input data about the new stock that has arrived is stored in a KSDS

file that is defined to CICS. The sample application browses each record in the file, and for each record, it then reads and updates the stock levels for the matching stock in the catalog manager stock file.

---

## Software requirements

To use SupportPac CN11, you must have the following versions of software and service installed:

- IBM CICS Transaction Server for z/OS V4.1 or higher
  - In addition CICS TS V4.1 requires the following service fixes to support the JVM server technology: [PM08649](#), [PM08661](#), [PM11157](#), [PM11791](#), and [PM32731](#).
- IBM DB2 for z/OS V8.1 or higher.
  - In addition, if using DB2 V8.1 the IBM Data Server Driver for JDBC and SQLJ version 3.62 provided by [PM32360](#) is required.
  - In addition, if using DB2 V9.1 or higher the IBM Data Server Driver for JDBC and SQLJ version 3.62 provided by [PM32360](#) or version 4.12 provided by [PM32361](#) is required.
- IBM WebSphere Extended Deployment (XD) Compute Grid V8.0.0.0 or higher.
  - In addition, we recommend applying the following service fix: PM54139.

A trial version is available from:

<http://www.ibm.com/software/webservers/appserv/extend/computegrid/>

The operating system and software requirements for WebSphere XD Compute Grid are available from:

<http://www-01.ibm.com/support/docview.wss?uid=swg27021569>

The Information Center for WebSphere XD Compute Grid is available from:

<http://publib.boulder.ibm.com/infocenter/wasinfo/cgwas80/index.jsp>

---

## Restrictions

This SupportPac has the following restrictions:

- Jobsteps written in CICS should not issue SYNCPOINT or SYNCPOINT ROLLBACK commands. The SupportPac will issue syncpoint or roll back on behalf of the jobstep during checkpoint processing.
- The Compute Grid Job Management Console cannot perform the following action on jobs executing in CICS; Stop.
- The following Batch Data Stream patterns provided with Compute Grid do not work with jobs executing in CICS:
  - JDBCReaderPattern
  - JDBCWriterPattern

---

## Chapter 2. Installing the SupportPac

Audience: experienced CICS system programmers who are responsible for installing and customizing CICS. Knowledge of downloading files from the internet, uploading files to zFS and MVS, and UNIX commands is assumed. For information about the options for the ftp, cp, and pax commands, see the *z/OS UNIX System Services Command Reference, SA22-7802*.

The following tasks guide you through downloading, installing, and configuring the SupportPac. These tasks assume that the software requirements are installed and that you are using WAS ND, although other versions of WebSphere Application Server are supported.

### Downloading and copying the SupportPac files

From a workstation:

1. Go to the CICS SupportPacs website at <http://www.ibm.com/support/docview.wss?rs=1083&uid=swg27007241>
2. Click **View all: By Category**.
3. Click **CN11**.
4. Download CN11.zip by clicking the appropriate link under **Download package**.
5. Providing you agree to the terms and conditions shown, click **I agree**.
6. Save the file CN11.zip to your workstation.
7. Extract the contents of CN11.zip to a local directory. The directory should contain the following files:

CN11.pdf	– This Installation and User's Guide
CN11.pax.Z	– SupportPac CN11 code
CN11SamplexJCL.xml	– xJCL code for the sample batch job
Licenses/Lic_*.txt	– License files in various languages
8. Upload the CN11.pax.Z file in binary to your z/OS system. For example, enter the following, where *mvs.mysite.com* is your z/OS system TCP/IP host name and *install\_directory* is the path and directory for the SupportPac files on z/OS UNIX.

```
ftp mvs.mysite.com
ftp> mkdir install_directory
ftp> cd install_directory
ftp> binary
ftp> put CN11.pax.Z
ftp> quit
```

From a z/OS UNIX shell:

1. Extract the CN11.pax.Z file, for example:

```
cd install_directory
pax -rvf CN11.pax.Z
```

2. Ensure that the appropriate permissions and attributes are set for the directories, subdirectories, and the files that are contained in them.

The following directories are in *install\_directory*:

CN11	
CN11/pipelines	– Pipeline configuration
CN11/jars	– Java archive files
CN11/workdir	– JVM server working directory

CN11/datasets – MVS data sets in XMIT format  
CN11/src – Java source file for the sample.

CICS needs read, write and execute permissions to the JVM server working directory and its joblog subdirectory. CICS needs to be able to read files in the pipeline configuration and the Java archive files directories.

## Create the PDS datasets

The SupportPac supplies MVS data sets compressed in XMIT format, which must be extracted. Throughout this document, enter the MVS high-level qualifier to be used for the SupportPac data sets where you see *INSTALL\_HLQ*.

1. From a z/OS UNIX shell copy the .XMIT files to MVS data sets; for example:

```
cd install_directory/CN11/datasets
```

```
cp -F bin -P "RECFM=FB,LRECL=80" ./CN11.JCL.XMIT  
"//'INSTALL_HLQ.CN11.JCL.XMIT'"
```

```
cp -F bin -P "RECFM=FB,LRECL=80" ./CN11.COBOL.XMIT  
"//'INSTALL_HLQ.CN11.COBOL.XMIT'"
```

```
cp -F bin -P "RECFM=FB,LRECL=80" ./CN11.LOAD.XMIT  
"//'INSTALL_HLQ.CN11.LOAD.XMIT'"
```

2. From a z/OS TSO shell, extract the data sets; for example:

```
RECEIVE INDATASET('INSTALL_HLQ.CN11.JCL.XMIT')
```

When prompted with the message "Enter restore parameters..." enter:  
DA('INSTALL\_HLQ.CN11.JCL')

```
RECEIVE INDATASET('INSTALL_HLQ.CN11.COBOL.XMIT')
```

When prompted with the message "Enter restore parameters..." enter:  
DA('INSTALL\_HLQ.CN11.COBOL')

```
RECEIVE INDATASET('INSTALL_HLQ.CN11.LOAD.XMIT')
```

When prompted with the message "Enter restore parameters..." enter:  
DA('INSTALL\_HLQ.CN11.LOAD')

3. From a z/OS TSO shell, remove the .XMIT data sets; for example:

```
DELETE 'INSTALL_HLQ.CN11.LOAD.XMIT'  
DELETE 'INSTALL_HLQ.CN11.JCL.XMIT'  
DELETE 'INSTALL_HLQ.CN11.COBOL.XMIT'
```

4. Ensure that the CICS default user ID has access to the *INSTALL\_HLQ.CN11.LOAD* data set.

These data sets are created:

*INSTALL\_HLQ.CN11.JCL* – JCL to configure CICS and DB2

*INSTALL\_HLQ.CN11.LOAD* – CICS program load modules

*INSTALL\_HLQ.CN11.COBOL* – Sample COBOL program source and copybooks

## Creating the sample VSAM files

To create the input and output files that are required by the SupportPac sample application, update and submit the *INSTALL\_HLQ.CN11.JCL* (CN11SAMP) JCL job:

1. Change all instances of <*INSTALL\_HLQ*> to the MVS high-level qualifier for the SupportPac.
2. Change all instances of <*VOL*> to the disk volume where you want to create the datasets.

3. Review the JCL job card and update as appropriate.
4. Save your changes and submit the job. The job should complete with a return code of 0.

The following data sets are created:

```

INSTALL_HLQ.CN11.CN11IN           – Sample VSAM input file
INSTALL_HLQ.CN11.CN11IN.DATA
INSTALL_HLQ.CN11.CN11IN.INDEX
INSTALL_HLQ.CN11.CN11OUT         – Sample VSAM output file
INSTALL_HLQ.CN11.CN11OUT.DATA
INSTALL_HLQ.CN11.CN11OUT.INDEX

```

## Configuring the JVM server profile

The SupportPac runs in the CICS JVM server environment and requires a JVM server profile configured appropriately for your environment, including adding JDBC support and appending the SupportPac .jar files to the class path.

1. Copy the CICS supplied sample DFHJVMAX and give it the name CN11PROF. Customize CN11PROF for your environment using advice in the CICS Information Center topic “JVM profiles: options and samples”.
2. Add support for JDBC provided by the IBM Data Server Driver for JDBC and SQLJ.
3. Add the following SupportPac .jar files to the Java class path, where *install\_directory* is the z/OS UNIX file system directory path that you created to unpack the installation files.

```

install_directory/CN11/jars/SupportPacCN11.jar
install_directory/CN11/jars/batchframework.jar
install_directory/CN11/jars/pgcccommon.jar
install_directory/CN11/jars/pgcruntime.jar
install_directory/CN11/jars/pgcstandalone.jar
install_directory/CN11/jars/SupportPacCN11Sample.jar

```

Note: For CICS TS 4.2, this is done by setting the properties:

```

JAVA_PIPELINE=YES
CLASSPATH_SUFFIX=jar1;jar2 etc

```

4. Set the WORK\_DIR parameter to the SupportPac provided workdir:  
WORK\_DIR=*install\_directory*/CN11/workdir

If WORK\_DIR is already set to a different directory, copy the files from *install\_directory*/CN11/workdir to the directory specified by WORK\_DIR.

5. Save your changes.

## Adding resources in the CICS CSD

The SupportPac must have resources installed in the CICS region. These resources are supplied in a group called CN11.

To create group CN11 update and submit the JCL job  
*INSTALL\_HLQ*.CN11.JCL(CSDDEF):

1. Change all instances of <CN11> to the SupportPac *install\_directory*, where *install\_directory* is the z/OS UNIX file system directory path that you created to unpack the installation files.
2. Change all instances of <CN11INST> to the SupportPac *INSTALL\_HLQ*.
3. Change <PortNumber> to the TCP/IP port number that CICS will listen on for inbound HTTP requests.
4. Change the PIPELINE resource SHELF directory as appropriate, for example:

/var/cicsts

5. Change <DB2NAME> to the name of the DB2 subsystem that will contain the SupportPac DB2 tables. If an appropriate DB2CONN resource is already defined in your CSD, the DEFINE statement here can be commented out or removed.
6. Change <YOUR\_CSD> to the data set name of your CICS CSD.
7. Review the JCL job card and STEPLIB statements and change them as appropriate.
8. Save your changes and submit the job. The job should complete with a return code of 4 because of warning message DFH5510 for the definition of transaction CN11 that starts with the letter C.

## Defining the tables in DB2

The SupportPac uses a JDBC connection from Java into DB2. Ensure that your DB2 system is installed and customized to allow connections from the IBM Data Server Driver for JDBC.

The SupportPac must have access to three DB2 tables to save job status, job step status, and checkpoint information.

To create these tables update and submit the JCL job *INSTALL\_HLQ.CN11.JCL(DB2SETUP)*:

1. Change <DB2INST> to the high-level qualifier for DB2, for example *SYS2.DB2.V910*
2. Change <DB2NAME> to the name of the DB2 subsystem that CICS is connecting to or, if you are not using DB2 V9, review and update (as appropriate) all instances of the following: *DSNTIA91* and *DSN910P2*
3. Review and update (as appropriate) the security permissions on the SQL GRANT statements and the JCL job card.
4. Save your changes and submit the job. The first time the job is run, it should complete with a return code of 8. If the job is run again, it should complete with a return code of 0. This is because the job first attempts to delete a database before recreating it and on first run, the delete fails as the database does not exist.

## Gathering WebSphere Application Server information

To configure the SupportPac, you need to find out the *WAS\_ND\_Cell*, *WAS\_ND\_Name*, *WAS\_ND\_Node*, *WAS\_ND\_Port*, and the fully qualified TCP/IP *WAS\_ND\_Host\_Name* to use for configuring the SupportPac.

To gather this information:

1. Start WAS ND. For example, you can use the Windows menu:

**Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.0 > Profiles > AppSrv01 > Start the server**

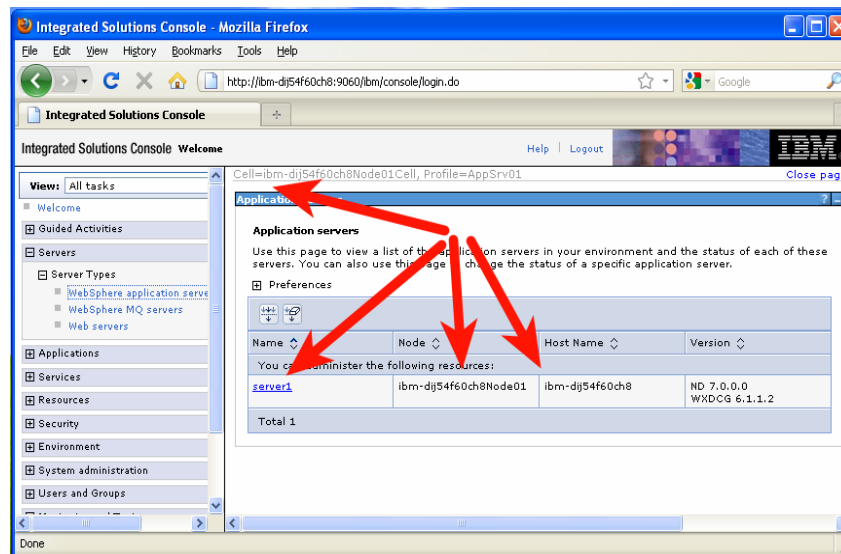
2. Start and log into the web browser administration console, for example, by using the Windows menu:

**Start > All Programs > IBM WebSphere > IBM WebSphere Application Server V8.0 > Profiles > AppSrv01 > Administrative console**

3. Expand **Servers** in the left column.
4. Expand **Server Types**.
5. Click **WebSphere application servers**. The top of the right column identifies the cell name. Take note of the *WAS\_ND\_Cell*, for example, *ibm-dij54f60ch8Node01Cell*



- A table of application servers is also shown in the right column. In this table locate the server that contains Compute Grid – this will have “WXDCG 8.0.0.1” or similar in the Version column.



- For the server also take note of:
  - WAS\_ND\_Name. For example: server1
  - WAS\_ND\_Node. For example: ibm-dij54f60ch8Node01
  - WAS\_ND\_Host\_Name. This must be the host name that can be resolved by CICS and might need to be fully qualified; for example: ibm-dij54f60ch8.hursley.ibm.com
- Click the identified server in the table, for example: server1. The server configuration is displayed. Under the Communications heading, expand the plus sign (+) next to **Ports**. Locate `WC_defaultHost` and take note of the WAS\_ND\_Port; for example, 9080.

## Configuring the SupportPac

The SupportPac requires a configuration file to define how to connect to Compute Grid and to define the job steps to be published.

- Edit the file `<WORK_DIR>/endpoint-config.xml`, where `<WORK_DIR>` is the working directory specified in the JVM server profile.
- Change all occurrences of `<CICSHOST>` to the TCP/IP host name on which CICS accepts HTTP requests. This host name must be resolved by the WebSphere XD Compute Grid and might need to be fully qualified.
- Change `<CICSPORT>` to the TCP/IP port number on which CICS accepts HTTP requests.
- Change `<CICSAPLID>` to the CICS region APPLID.
- Change `<CGHOST>` to the value for the WAS\_ND\_Host\_Name.
- Change `<CGPORT>` to the value for the WAS\_ND\_Port.
- Change `<CGCELL/NODE/SERVER>` to the WAS\_ND\_Cell/WAS\_ND\_Node/WAS\_ND\_Name, for example:
 

```
ibm-dij54f60ch8Node01Cell/ibm-dij54f60ch8Node01/server1
```
- Save your changes.

## Updating CICS

The SupportPac requires CICS to have TCP/IP enabled, for the JVM server profile to be accessible, and for the resources defined in the DFHJAVA and CN11 CSD groups to be installed and enabled. This can be achieved a number of ways, for example:

1. Update your CICS SIT parameters to include:  
TCPIP=YES  
JVMPROFILEDIR=<jvmprofiledir>

where <jvmprofiledir> is the directory where you saved the customized CICS JVM server profile in a previous step.

2. Ensure the DB2 load libraries SDSNLOAD and SDSNLOAD2 are in the STEPLIB concatenation for the CICS region.
3. Install the CSD groups DFHJAVA and CN11.  
CEDA INSTALL GROUP(DFHJAVA)  
CEDA INSTALL GROUP(CN11)
4. Verify that the JVMSERVER resource is enabled, for example:  
CEMT INQUIRE JVMSERVER(CN11SERV)
5. Verify that the PIPELINE resource is enabled, for example:  
CEMT INQUIRE PIPELINE(CN11PIPE)
6. Verify that the DB2CONN resource is connected to DB2, for example:  
CEMT INQUIRE DB2CONN
7. Verify that the FILE resources are enabled, for example:  
CEMT INQUIRE FILE(CN11IN)  
CEMT INQUIRE FILE(CN11OUT)

---

## Chapter 3. Getting started

---

### Starting the SupportPac

From a CICS 3270 terminal, run transaction CN11. The transaction completes with message:

```
SupportPac initiated. Check JVM logs for startup progress
```

Transaction CN11 links to a program in the JVM server; the program loads the SupportPac configuration file, initializes the environment, and registers with Compute Grid. In addition a scheduled task periodically publishes available job steps to Compute Grid.

The dfhjvmout and dfhjvmerr files in the JVM server <WORK\_DIR> directory contain progress and error messages, respectively.

When transaction CN11 completes, the dfhjvmout file should confirm a successful initialization, for example:

```
Initiating SupportPac CN11
----0
pgc-services.properties Exists! loading it..
Loading: install_directory/CN11/workdir/endpoint-config.xml
----1
----2
----3
SupportPac CN11 Successfully Initiated
```

When the SupportPac registers with Compute Grid, the WebSphere Application Server SystemOut.log file will contain an information message:

```
Compute Grid Portable Endpoint: <CICSHOST>/<CICSAPPLID>
registered successfully!
```

---

### Running the SupportPac sample

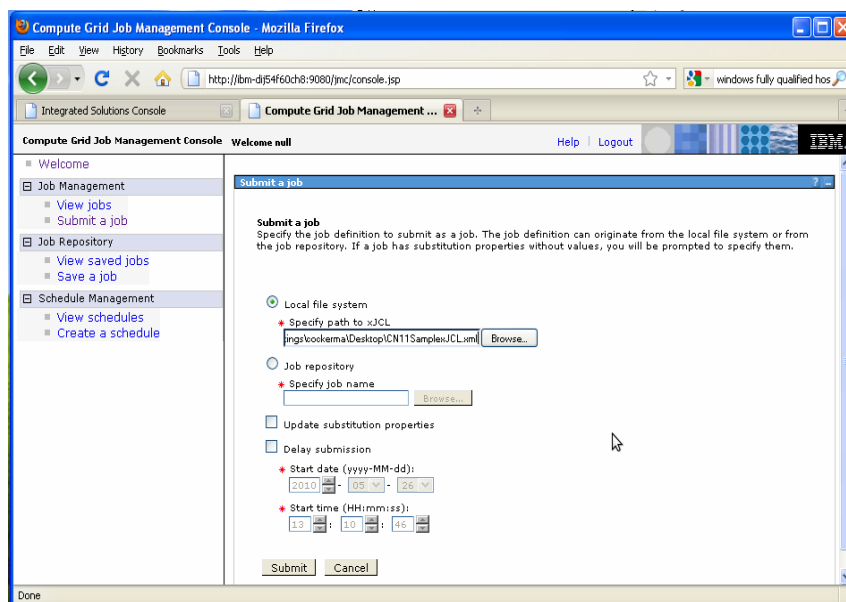
To run the SupportPac sample, submit the provided sample xJCL job:

1. Go to `http://<WAS_ND_Host_Name>:<WAS_ND_Port>/jmc/console.jsp` to display the Compute Grid Job Management Console.

For example:

```
http://ibmdij54f60ch8.hursley.ibm.com:9080/jmc/console.jsp
```

2. Click **Job Management > Submit a job.**
3. Click **Local file system > Browse**, and select the CN11SamplexJCL.xml file that was extracted from the CN11.zip file.



4. Click **Submit**. The web page displays a message: Successfully submitted the job definition CN11SamplexJCL.xml with job ID CN11Sample:xxxxx where xxxxx is the job ID. To view the job status, click **Job Management > View jobs** and search the table for the job ID.

See “Troubleshooting” on page 16 if you see an error message.

When the job has been submitted, Compute Grid sends an HTTP request to CICS to start the job. CICS analyzes the HTTP path and matches it against the path in the CN11URI URIMAP resource and starts the CN11PIPE pipeline to handle the request.

The pipeline starts the `com.ibm.cics.CN11.CN11Listener.SupportPac` Java class in the JVM server to process the HTTP request and run the sample job step as specified in the xJCL. The job step is implemented by the `com.ibm.cics.CN11.samples.CatalogManagerStockUpdate` Java class.

The source for this Java class is supplied in the file `install_directory/CN11/src/com/ibm/cics/CN11/samples/CatalogManagerStockUpdate.java`

When the job completes, the job state changes to Ended and the job log contains a success message, for example:  
Job CN11Sample:xxxxx ended normally

## Understanding the xJCL

The Compute Grid uses a job control language called xJCL, which is defined using XML notation. The file `CN11SamplexJCL.xml` contains the xJCL for the SupportPac sample.

Note the following XML elements in the file:

- A **job** including its name and default application name.
- A **checkpoint-algorithm** to use; in this sample, take a checkpoint every 5 iterations of the batch job step.
- A **results-algorithms** to use. For multiple batch job steps, this describes how to work out the return code for the overall job.
- A batch **job-step** to run, including the name of the job step and any properties to be passed to it.
- Two **batch-data-streams**: one for the input file and another for the output file. Each defines: 1) a **logical-name**, 2) **props** - a set of

properties in name/value pair notation, such as the file name and key length and starting key, and 3) **impl-class** - a Java implementation class.

```
<?xml version="1.0" encoding="UTF-8" ?>
  <job name="CN11Sample" default-application-name="CN11Sample"
xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
xsi:noNamespaceSchemaLocation="http://core.wcg.ertools.ibm.com/xsd/xJCL.xsd">
  <job-type>Batch</job-type>
  <jndi-name>com/ibm/cics/CN11/CICSCatalogManagerBatchController</jndi-name>
  <step-scheduling-criteria>
    <scheduling-mode>sequential</scheduling-mode>
  </step-scheduling-criteria>
  <checkpoint-algorithm name="chkpt">
    <classname>com.ibm.wsspi.batch.checkpointalgorithms.recordbased</classname>
    <props>
      <prop name="recordcount" value="5" />
    </props>
  </checkpoint-algorithm>
  <results-algorithms>
    <results-algorithm name="jobsum">
      <classname>com.ibm.wsspi.batch.resultsalgorithms.jobsum</classname>
    </results-algorithm>
  </results-algorithms>
  <substitution-props></substitution-props>
  <job-step name="IVTStep1">
    <classname>com.ibm.cics.CN11.samples.CatalogManagerStockUpdate</classname>
    <checkpoint-algorithm-ref name="chkpt"/>
    <batch-data-streams>
      <bds>
        <logical-name>inputStream</logical-name>
        <impl-class>com.ibm.cics.CN11.bds.impl.CICSVsamKsdsReaderImpl
        </impl-class>
        <props>
          <prop name="CICSFILE" value="CN11IN" />
          <prop name="KEYLENGTH" value="4" />
          <prop name="START" value="AAAAAA==" />
        </props>
      </bds>
      <bds>
        <logical-name>outputStream</logical-name>
        <impl-class>com.ibm.cics.CN11.bds.impl.CICSVsamKsdsRecordUpdaterImpl
        </impl-class>
        <props>
          <prop name="CICSFILE" value="CN11OUT" />
        </props>
      </bds>
    </batch-data-streams>
    <props>
      <prop name="debug" value="true"/>
    </props>
  </job-step>
</job>
```

See the Compute Grid Information Center for more information about xJCL elements.

## Understanding the CatalogManagerStockUpdate job step

The Sample job step is implemented by the CatalogManagerStockUpdate Java class. This class uses the Compute Grid batch programming model. The key components of the application are:

- For each step in the batch job, a Java class is written to implement the BatchJobStepInterface. This is where the business logic is performed.
- BatchDataStreams are used to access data. For the SupportPac sample, the data is in two VSAM files that must be opened in CICS before the job step runs.

The SupportPac supplies two implementations of BatchDataStreams that are used by the CatalogManagerStockUpdate application:

- CICSVsamKsdsReaderImpl allows the records in a VSAM KSDS file that is managed by CICS to be browsed.
- CICSVsamKsdsRecordUpdaterImpl allows a record in a VSAM KSDS file to be read for update by its primary key and then to be updated.

The CatalogManagerStockUpdate application has three key methods:

- createJobStep() looks up the BatchDataStreams.
- processJobStep() is called to read an input record and update the output record. It is called iteratively until all records have been read.
- destroyJobStep() is called at the end of the batch job step application to clean up any resources.

To demonstrate how a non-Java program could be called as part of a job step, the sample uses JCICS to call the COBOL program CATUPDAT, which performs the business logic that adjusts the stock and back-order levels based on the amount of new stock that arrived.

---

## Batching

The application must perform the updates to the catalog manager stock file in small batches. At the end of each batch, a checkpoint is taken and the updates are committed to the file. In the xJCL code that is provided, the checkpointing is done after every five calls to the application. The application processes one update per call. This approach ensures that the number of records locked by the CatalogManagerStockUpdate application remains small, so that the majority of the records are always accessible to the online catalog manager application.

This approach also means that if a failure occurs during the run of the CatalogManagerStockUpdate application, you can restart the application from the last checkpoint, rather than having to restart all the updates again.

---

## Locking

By performing the updates in small batches, the effect of locked records on the online catalog manager application can be minimized; however, in general, some thought still needs to be given to the design, to ensure that both the online application and the new batch application can handle the case where the record they need to access is temporarily held by another program.

In the case of the catalog manager application, browses and reads of the file will read uncommitted updates. When an order is placed, the record that is associated with the order is locked until a sync point occurs. If the record is already locked when the catalog manager application tries to lock it, the program will wait unless the lock is retained.

The design of the batch application CatalogManagerStockUpdate is similar. The record is locked to update it and the lock is held until a sync point occurs. If the record is locked by another program, the batch application waits to get the lock (unless the lock is retained). Both the CatalogManagerStockUpdate application and catalog manager application can each handle a record that is being locked by the other application.

Neither the catalog manager application nor the batch CatalogManagerStockUpdate application expect to see retained locks. The Catalog Manager does not allow an order to be placed if the record cannot be obtained. The batch application writes an error to the Java System error file.

---

## Stopping the SupportPac

After the SupportPac is started it continues to run in the JVM server. To stop the SupportPac, disable the JVM server; for example:

```
CEMT SET JVMSERVER(CN11SERV) DISABLED
```

To start the SupportPac again, first enable the JVM server and then run the CN11 transaction:

```
CEMT SET JVMSERVER(CN11SERV) ENABLED
```

```
CN11
```

---

## Troubleshooting

If the SupportPac sample does not work, check the following on z/OS:

- The CICS resources in CSD group CN11 are enabled.
- If transaction CN11 ABENDs with the 4088 code, check that the DB2CONN resource connection status is 'CONNECTED'.
- The CICS job log does not contain error messages.
- There are no error messages in the files  
<WORK\_DIR>\*.dfhjvmerr and <WORK\_DIR>\*.dfhjvmout
- If dfhjvmerr shows a Java exception with the message `SQLCODE = -923, SQLSTATE = 57015 error tokens = ACCESS;00E30301;00000800;DSNJCC`, ensure that your DB2 system is installed and customized to allow connections from the IBM Data Server Driver for JDBC.
- There are no error messages in the files  
<WORK\_DIR>/joblog/PortableGridContainerLog.\*
- The SupportPac configuration file is correct:  
<WORK\_DIR>/endpoint-config.xml
- The CICS system and the Compute Grid system can communicate with each other using HTTP.

Also check the following using a web browser:

- The job output, which is available in the Compute Grid Job Management Console.
- The WAS ND System.out and System.err files do not contain errors. You can view these files from the Integrated Solutions Console; click: **Servers > Server Types > WebSphere application servers > your server > Logging and tracing > JVM Logs > Runtime > View**

---

## Feedback

What do you think of this SupportPac? We welcome your comments by email to [suppacs@uk.ibm.com](mailto:suppacs@uk.ibm.com), or on the WebSphere XD Compute Grid newsgroup:

<http://www.ibm.com/developerworks/forums/forum.jspa?forumID=1240&start=0>

---

## Chapter 4. References

- IBM CICS Transaction Server  
[ibm.com/software/http/cics/tserver/v42/](http://ibm.com/software/http/cics/tserver/v42/)
  - IBM WebSphere Extended Deployment Compute Grid  
[ibm.com/software/webservers/appserv/extend/computegrid/](http://ibm.com/software/webservers/appserv/extend/computegrid/)
- IBM Education Assistant presentations  
[publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wxd\\_v6/wxd/6.1/ComputeGrid.html?dmuid=20070905072459439555](http://publib.boulder.ibm.com/infocenter/ieduasst/v1r1m0/topic/com.ibm.iea.wxd_v6/wxd/6.1/ComputeGrid.html?dmuid=20070905072459439555)
- IBM WebSphere Application Server Network Deployment  
[ibm.com/software/webservers/appserv/was/network/](http://ibm.com/software/webservers/appserv/was/network/)
  - Batch Modernization on z/OS  
[redbooks.ibm.com/redpieces/abstracts/sg247779.html](http://redbooks.ibm.com/redpieces/abstracts/sg247779.html)
  - IBM Redbooks® publication  
Batch Processing with WebSphere Compute Grid: Delivering Business Value to the Enterprise, [redbooks.ibm.com/abstracts/REDP4566.html](http://redbooks.ibm.com/abstracts/REDP4566.html)



---

## Chapter 5. Notices

The provisions set out in the following two paragraphs do not apply in the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY, OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore this statement may not apply to you.

Information contained and techniques described in this publication have not been submitted to any formal IBM test and are distributed on an "AS IS" basis.

The use or implementation of any information contained and/or of any technique described in this document is the user's responsibility and depends on the user's ability to evaluate and integrate the information and/or technique into the user's operational environment. While IBM has reviewed each item for accuracy in a specific situation, IBM offers no guarantee or warranty that the same or similar results will be obtained elsewhere. Users attempting to adapt any technique described in this document to their own environments do so at their own risk.

The information contained in this publication could include technical inaccuracies or typographical errors.

Changes are periodically made to the information contained herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any reference in this publication to an IBM licensed program or another IBM product is not intended to state or imply that only IBM's program or other product may be used. Any functionally equivalent program that does not infringe applicable intellectual property rights may be used instead of the referenced IBM licensed program or other IBM product.

The user is responsible for evaluating and verifying the operation of the material supplied in conjunction with this publication in conjunction with other products, except those expressly designated by IBM.

International Business Machines Corporation may have patents or pending patent applications covering subject-matter described in this document. The furnishing of this document does not give you any license to any such patent. You can send license inquiries, in writing, to:

The IBM Director of Licensing  
International Business Machines Corporation  
North Castle Drive  
Armonk, NY 10504-1785  
U.S.A.

---

## License

The international license agreement for the SupportPac is available as a set of language-specific text files in the directory Licenses in the CN11.zip file. The SupportPac is provided "as-is" and does not include defect correction.

---

## Trademarks

The following terms are trademarks of International Business Machines

Corporation in the United States, or other countries, or both:

CICS	MVS	WebSphere
IBM	Tivoli	z/OS
DB2	System z	

Java and all Java-based trademarks and logos are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other product and service names might be trademarks of IBM or other companies.