Feature Abstract by First Appearance in SDK – Security Features

Abstract	SDK V7.0	SDK V7R1	SDK V8.0	SDK V11.0
Addition of **PlatformAccessControl.checkMyPermission**	N/A	N/A	SR4FP2	GA
OpenJCEPlus cryptographic provider for z/OS	N/A	N/A	N/A	GA

Java Security – Feature Descriptions (order matches order of appearance in tables)

- Addition of **PlatformAccessControl.checkMyPermission** - This access control method is used to check whether a specific user has permission to a resource. A similar method, checkPermission, requires the current user to either have READ access to BPX.SERVER or to have superuser status. However, checkMyPermission() only requires that the current user has a task-level Access Control Environment Element (ACEE).
- OpenJCEPlus cryptographic provider for z/OS - The OpenJCEPlus cryptographic provider is an implementation of the Java™ Cryptography Extensions (JCE) APIs, which include, for example: ciphers, signatures, message digests, MACs and HMACs, secure random number generation, and key generation. OpenJCEPlus is a z/OS-specific provider supported in 64-bit releases of IBM Semeru Runtime Certified Edition for z/OS, Version 11.0.

Feature Abstract by First Appearance in SDK – Legacy Features

Abstract	SDK V6.0.0	SDK V6.0.1	SDK V7.0
Introduction of JZOS to SDK	SR1	GA	GA
JZOS SMF record write support	SR1	GA	GA
JZOS Rauditx support	SR1	GA	GA
JZOS additional system services	SR3	GA	GA
JZOS implement IDCAMS	SR3	GA	GA
z/OS DFSORT API Support	SR3	GA	GA
Add support to JZOS for reading Extended Address Volume (EAV) Data Set Control Blocks (DSCBs)	N/A	GA	GA
Add support to JZOS for z/OS logstream read/browse	N/A	GA	GA
Add support to JZOS for native job submission	N/A	GA	GA

Updated 21 October 2021

Feature Abstract by First Appearance in SDK – Legacy Features (cont)

Abstract	SDK V6.0.0	SDK V6.0.1	SDK V7.0
JZOS Add support for Workload Manager (WLM) services	N/A	GA	GA
JZOS Add IO interfaces for z/OS logstreams	N/A	GA	GA
JRIO Deprecation and migration to JZOS	N/A	GA	GA
JZOS ZFile Enqueue Support	N/A	GA	GA
JZOS Sequential Dataset I/O Performance Improvement	N/A	SR1	SR1
JZOS: zCompression wrapper (CMPSC compression wrapper)	N/A	N/A	SR3

Updated 21 October 2021

Feature Abstract by First Appearance in SDK – Legacy Features (cont)

Abstract	SDK V7.0	SDK V7.0.1	SDK V8.0
JZOS WLM Query Contention wrapper	N/A	GA	GA
Java Data Access Accelerator – JZOS exploitation	N/A	GA	GA
Ability to force an abnormal termination in JZOS	N/A	GA	GA
JZOS API to support USS file tagging in Java	N/A	SR2FP10	GA
JZOS JES Symbol Support and Preliminary Support for User Job Correlators	N/A	SR2FP10	GA
Provide JMX based performance logging to SMF for the JZOS launcher	N/A	SR2FP10	GA
Removal of JRIO component from z/OS Java SDK	N/A	N/A	GA
Augment JZOS launcher SMF logging data with CPU consumption data	N/A	SR3FP20	SR1
Augment JZOS launcher SMF logging data with native thread ID	N/A	SR3FP20	SR2

Updated 21 October 2021

Feature Abstract by First Appearance in SDK – Legacy Features (cont)

Abstract	SDK 7.0	SDK V7.0.1	SDK V8.0	SDK V11.0
JZOS Supported CLASSPATH length increased to 150,528 bytes	N/A	N/A	SR5	GA
JZOS Add support for ISGENQ RNL= option to JZOS Enqueue class	N/A	N/A	SR5FP20	GA
JZOS java.time.* nanosecond support in JUtil	N/A	N/A	SR5FP20	GA
Provide consistent behavior between CLASSPATH wildcarding in Java command and JZOS launcher	N/A	N/A	SR5FP20	GA
zOS batch Cobol/Java mix: redirect java stdout/stderr into COBOL SYSOUT	N/A	N/A	SR5FP20	GA
JZOS Enqueue Try-with-Resources Support	N/A	N/A	SR5FP25	GA
JZOS ZFile Unlock Support	N/A	N/A	SR5FP35	GA
JZOS Query Virtual Server (QVS) support	N/A	N/A	N/A	GA

Java Batch – Legacy Feature Descriptions (order matches order of appearance in tables)

- Introduction of JZOS to SDK – An aggregation of legacy services support through Java was first introduced into the SDK at this time.
- JZOS SMF record write support – Feature provided Java APIs for z/OS SMF record writing for applications needing to create the z/OS SMF log records from a Java application.
- JZOS Rauditx support – Feature provided Java APIs which enable Java applications to record security relevant events to the z/OS System Management Facility (SMF) repository. Specifically, this API enables the creation of security SMF 83 type 5 records that enable Java applications to write SMF 83 records in a controlled fashion.
- JZOS additional system services - Feature provided additional system services to Java on z/OS that allow customers to: 1) use currently available system serialization on z/OS resources via ENQ/DEQ (ISGENQ) , 2) access system-wide symbols, 3) provide new services for getting clock values using STCK and STCKE instructions, and 4) redirect all output streams into a single DD output statement.
- JZOS implement IDCAMS – Feature provided a Java API to allow applications to use the currently available system utility functions for creating and deleting VSAM datasets.
- z/OS DFSORT API Support – Feature provided a Java API to spawn a child process to run an executable program that invokes DFSORT, and which supports multiple concurrent requests.
- Add support to JZOS for reading Extended Address Volume (EAV) Data Set Control Blocks (DSCBs) – Prior to this feature, the JZOS API for the OBTAIN Macro returned only Format-1 DSCB information. This feature enhanced JZOS to provide support for extended access volumes (Format-8 and Format-9 DSCBs), introduced in z/OS V1.10.

Java Batch – Legacy Feature Descriptions (order matches order of appearance in tables)

- Add support to JZOS for z/OS logstream read/browse – Feature provided an API to read existing logstreams from Java (in addition to being able to write to logstreams, function which already existed in prior versions of the SDK).
- Add support to JZOS for native job submission – Feature provided an API in JZOS for submitting jobs to the internal reader and returning the Job ID to the caller.
- Add support for Workload Manager (WLM) services - Java applications all run against the default RMF profile. This feature added a set of Java APIs to create / join / leave / delete a WLM work unit which allow Java processes to be tracked discretely by WLM.
- Add IO interfaces for z/OS logstreams – Feature enabled z/OS Logstreams to be writable using standard Java interfaces (java.io.OutputStream).
- JRIO Deprecation and migration to JZOS - The record I/O support in the JZOS component is designated as the strategic implementation, supplanting JRIO in future SDKs. JRIO has been available and widely used since being introduced in SDK1.4.2 SR6. In SDK 7, JRIO classes are marked deprecated and generate compiler warnings for customers using these classes, indicating that these classes are no longer considered strategic implementations.
- JZOS ZFile Enqueue Support – Feature provided a new ZFile constructor that has the same capabilities as the existing ZFile constructor with two additional flags, FLAG_DISP_SHR and FLAG_PDS_ENQ, to automate the serialization support of PDS members (previously, this could be done explicitly using JZOS ENQ/DEQ support).

Java Batch – Legacy Feature Descriptions (order matches order of appearance in tables)

- JZOS Sequential Dataset I/O Performance Improvement – Feature implemented peformance improvements on BSAM reads and writes such that dataset I/O through JZOS now more closely approximates the performance of similar COBOL I/O to these datasets.
- JZOS: zCompression wrapper (CMPSC compression wrapper) – Feature provides Java APIs for invocation of compression function using the z/Architecture CMPSC hardware Compression instruction, rather than using software-based compression. The APIs support CMPSC, GZIP (from java.util.zip), and INFLATE/DEFLATE (from java.util.zip).
- JZOS WLM Query Contention wrapper - Adds support for Workload Manager (WLM) services by providing a Java binding for the SYSEVENT QRYCONT macro which debuts in z/OS V2R1.
- Java Data Access Accelerator – JZOS exploitation - SDK 7.1 introduced a Data Access Accelerator solution which includes a Java class library that provides a set of primitive data access/manipulation functions and JIT compiler which acts in concert with the Data Access Accelerator library, replacing known methods with generated inline hardware instructions when compiling Java methods that include calls to these special functions. The JZOS component integrates exploitation of DAA into the COBOL copy book record parsing infrastructure.
- Ability to force an abnormal termination in JZOS – Prior to SDK 7.1, when an abend occurs in a JZOS Batch Launcher job, JZOS exception handling takes effect and sets a return code greater than zero for the termination of the step. In such error situations, the normal JES mechanisms should take effect in the current step (dataset disposition). Previously it was possible to set a non-zero return code, but a non-zero return code is not an “abnormal termination” error situation for JES, and job step failure mechanisms were not invoked. A new optional environment variable for the JZOS Batch Launcher is available to force the Batch Launcher into abnormal termination. From an application design perspective, this new behavior of JZOS will match the traditional batch approach/principles.

Java Batch – Legacy Feature Descriptions (order matches order of appearance in tables)

- JZOS API to support USS file tagging in Java - Added new APIs for accessing z/OS USS specific file attributes, such as extended attribute flags (Shared Library, No Shareas, APF Authorized, Program Controlled), user and auditor audit flags, change time and reference time, file format, file tagging, and security label.
- JZOS JES Symbol Support and Preliminary Support for User Job Correlators - The JES Symbol Service, introduced in the z/OS V2R1 release, provides a single view of JCL and JES symbols. The service allows applications to create, update, delete, or extract JES symbols. This JZOS feature adds JES Symbol support APIs to enable applications to create, update, delete, and extract JES symbols.
- Provide JMX based performance logging to SMF for the JZOS launcher – Enables JZOS batch launcher to write a z/OS SMF record containing z/OS Java runtime performance information before JVM shutdown using existing JMX beans capabilities. The z/OS Java runtime performance information to be written to the SMF record includes JVM name, JVM creation time, JVM uptime, JVM Garbage Collection policy and Garbage Collector details, Peak thread count, and live thread count.
- Removal of JRIO component from z/OS Java SDK - The JRIO component of z/OS Java SDKs was deprecated in SDK V6.0.1. In accordance with the Statement of Direction in the SDK V7.0 Announcement, this component has been withdrawn from SDK V8. For SDK V8 and later releases, use the record I/O facilities in the JZOS component in lieu of the JRIO facilities.
- Augment JZOS launcher SMF logging data with CPU consumption data, native thread ID - SMF record 121.1 updated to add precise and accountable CPU resource consumption information for each of these four thread categories: Application, System-JVM (including GC and JIT). GC, JIT. In addition to summary information, individual per-thread detail may be optionally configured to include Java thread ID (including native z/OS thread ID), Thread name, Thread category, Total CPU time.

Java Batch – Legacy Feature Descriptions (order matches order of appearance in tables)

- JZOS Supported CLASSPATH length increased to 150,528 bytes – The length of the supported CLASSPATH in JZOS has been increased from 16k bytes to 150,528 bytes.
- JZOS Add support for ISGENQ RNL= option to JZOS Enqueue class - Updated the JZOS Enqueue class to allow a user to set the RNL value to either YES or NO. The RNL value as specified by the user will then be passed to the ISGENQ macro when it is invoked.
- JZOS java.time.* nanosecond support in JUtil – In Java 1.8, Java is able to handle finer granularity using java.time.* classes. The JZOS java.time class has been enhanced to provide this nanosecond support.
- Provide consistent behavior between CLASSPATH wildcarding in Java command and JZOS launcher - Consistent behavior between CLASSPATH support for wildcards in both Java command invocation and JZOS batch launcher invocation improves usability and fidelity of function across the product. For traditional Unix developers not intimately familiar with z/OS JCL and dataset concatenations, eases configuration where JZOS jobs require hundreds of specific jars files to be appended to CLASSPATH.
- zOS batch Cobol/Java mix: redirect java stdout/stderr into COBOL SYSOUT – Provides the option of merging output from batch jobs which invoke a combination of Java and COBOL language executables onto COBOL SYSOUT.
- JZOS Enqueue Try-with-Resources Support - Added an AutoCloseable implementation to com.ibm.jzos.Enqueue to ensure that enqueues for resources are released automatically at the end of a try-with-resources block.
- JZOS ZFile Unlock Support – Provides unlocked LE methods fread and flocate in ZFile.
- JZOS Query Virtual Server (QVS) support - A Java wrapper for IWMQVS-Query Virtual Server returning the machine type and machine capacity for registration of pricing data.