



APUSIC
固若长城
睿比世界

错误码说明文档

金蝶Apusic应用服务器V10

版权所有 © 深圳市金蝶天燕云计算股份有限公司2026。保留所有权利。

版权声明

本档所涉及的软件著作权、版权等知识产权已依法进行了注册，由金蝶天燕云计算股份有限公司合法拥有。受《中华人民共和国著作权法》《计算机软件保护条例》《知识产权保护条例》和相关国际版权条约、法律、法规以及其它知识产权法律和条约的保护。未经授权许可，不得非法使用。

免责声明

本档包含的版权信息由金蝶天燕云计算股份有限公司合法拥有，受法律的保护，金蝶天燕云计算股份有限公司对本档可能涉及到的非金蝶天燕云计算股份有限公司的信息不承担任何责任。在法律允许的范围内，您可以查阅并仅能够在《中华人民共和国著作权法》规定的合法范围内复制和打印本档。任何单位和个人未经金蝶天燕云计算股份有限公司书面授权许可，不得使用、修改、再发布本档的任何部分和内容，否则将被视为侵权，金蝶天燕云计算股份有限公司有依法追究其责任的权利。

本档如有更新，不另行通知。对本档中的问题您可向金蝶天燕云计算股份有限公司告知或查询。未经本公司明确授予的任何权利均予保留。

商标声明

 是深圳市金蝶天燕云计算股份有限公司向中华人民共和国国家商标局申请注册的注册商标，注册商标专用权由金蝶天燕合法拥有，受法律保护。未经金蝶天燕的书面许可，任何单位及个人不得以任何方式或理由对该商标的任何部分进行使用、复制、修改、传播、抄录或与其它产品捆绑使用销售。凡侵犯金蝶天燕商标权的，金蝶天燕将依法追究其法律责任。本档提及的其他所有商标或注册商标，由各自的所有人拥有。

目录

- 1 更新历史
- 2 前言
- 3 概述
 - 3.1 产品介绍
 - 3.2 字符说明
 - 3.3 默认路径
- 4 错误码说明
 - 4.1 AS-ACDEPL-00103 Java Web Start services started for the app client {0} (contextRoot: {1})
 - 4.2 AS-ACDEPL-00105 Error starting the adapter to serve static system-level content
 - 4.3 AS-ACDEPL-00108 Config file for client JNLP not found: {0}
 - 4.4 AS-ACDEPL-00109 The config file for client JNLP {0} exists but could not be read.
 - 4.5 AS-ACDEPL-00111 Client JNLP document {0} refers to the static resource {1} that does not exist or is not readable
 - 4.6 AS-ACDEPL-00112 The following extensions or libraries are referenced from the manifest of {0} but were not found where indicated: {1}; ignoring and continuing
 - 4.7 AS-ACDEPL-00114 The ApplicationSignedJARManager for a nested app client deployer helper is unexpectedly null
 - 4.8 AS-ACDEPL-00115 The custom JNLP document {0} in a stand-alone app client incorrectly refers to a JAR {1}
 - 4.9 AS-ACDEPL-00101 Java Web Start services not started for the app client {0}; its developer has marked it as ineligible
 - 4.10 AS_ACDEPL_00102 Java Web Start services not started for the app client {0}; the administrator has disabled Java Web Start support for it
 - 4.11 AS_ACDEPL-00104 Java Web Start services stopped for the app client {0}
 - 4.12 AS-NAMING-00001 Failed to load {0} using CommonClassLoader
 - 4.13 AS-NAMING-00002 Exception during name lookup : {0}
 - 4.14 AS-NAMING-00003 Exception occurred when instantiating LocalSerialContextProviderImpl: {0}
 - 4.15 AS-NAMING-00004 Error during CosNaming.unbind for name {0}: {1}
 - 4.16 AS-NAMING-00005 Naming binding already exists for {0} in namespace {1}
 - 4.17 AS-NAMING-00006 Exception in NamingManagerImpl copyMutableObject(): {0}
 - 4.18 AS-NAMING-00008 Fall back to INITIAL_CONTEXT_FACTORY {0}

- 4.19 AS-CONCURRENT-00001 Task [{0}] has been running on thread [{1}] for {2} seconds, which is more than the configured hung task threshold of {3} seconds in [{4}].
- 4.20 AS-CONCURRENT-00002 Unable to setup or reset runtime context for a task because an invalid context handle is being passed
- 4.21 AS-CONCURRENT-00003 Unable to bind {0} to JNDI location [{1}]
- 4.22 AS-CONCURRENT-00004 Unable to deploy {0}
- 4.23 AS-RAR-05001 Failed to get the thread-pool [{0}] for resource adapter [{1}].
- 4.24 AS-RAR-05002 Failed to get the default thread-pool for resource adapter [{0}].
- 4.25 AS-RAR-05003: An error occurred during instantiation of the Work Manager class [{0}] for resource adapter [{1}]
- 4.26 AS-RAR-05004 The Work named [{0}], progress [{1}].
- 4.27 AS-RAR-05005 Resource adapter association failed
- 4.28 AS-RAR-05006 Unable to load Work Context class {0}
- 4.29 AS-RAR-05007: Cannot specify both Execution Context [{0}] as well as Transaction Context [{1}] for Work [{2}] execution. Only one can be specified
- 4.30 AS-RAR-05008 Duplicate Work Context for type [{0}]
- 4.31 AS-RAR-05009 Application server cannot handle the following Work Context : {0}
- 4.32 AS-RAR-05010: Setting custom Work Context class [{0}] using most specific supported Work Context class [{1}]
- 4.33 AS-RAR-05011 Unable to set Security Context
- 4.34 AS-RAR-05012 Unsupported callback {0} during credential mapping
- 4.35 AS-CORE-JAVAEE-0001 Web Container not installed
- 4.36 AS-CORE-JAVAEE-0002 Done with starting {0} container
- 4.37 AS-CORE-JAVAEE-0003 Unable to start container (no exception provided)
- 4.38 AS-CORE-JAVAEE-0004 Unable to start container {0}
- 4.39 AS-DEPLOYMENT-04017 Error in deleting file {0}
- 4.40 AS-DEPLOYMENT-04018 Deployment manager load failure. Unable to find {0}
- 4.41 AS-DEPLOYMENT-04019 Deployment manager load failure. Unable to find {0}
- 4.42 AS-DEPLOYMENT-00001 Ignore {0} in archive {1}, as WLS counterpart runtime xml {2} is present in the same archive
- 4.43 AS-DEPLOYMENT-00002 Exception caught: {0}
- 4.44 AS-DEPLOYMENT-00003 {0} module [{1}] contains characteristics of other module type: {2}

- 4.45 AS-DEPLOYMENT-00004 Exception caught: {0} for the subarchive indicated by the path: {1}
- 4.46 AS-DEPLOYMENT-00005 Exception caught during annotation scanning
- 4.47 AS-DEPLOYMENT-00006 Adding {0} since {1} is annotated with {2}
- 4.48 AS-DEPLOYMENT-00007 Adding {0} since it is implementing {1}
- 4.49 AS-DEPLOYMENT-00008 Inconsistent type definition. {0} is neither an annotation nor an interface
- 4.50 AS-DEPLOYMENT-00009 The exception {0} occurred while examining the jar at file path: {1}
- 4.51 AS-DEPLOYMENT-00010 No classloader can be found to use
- 4.52 AS-DEPLOYMENT-00011 Error in annotation processing: {0}
- 4.53 AS-DEPLOYMENT-00012 Cannot load {0} reason : {1}
- 4.54 AS-DEPLOYMENT-00013 An exception was caught during library jar processing: {0}
- 4.55 AS-DEPLOYMENT-00014 Could not load {0}
- 4.56 AS-DEPLOYMENT-00015 Unsupported deployment descriptors element {0} value {1}
- 4.57 AS-DEPLOYMENT-00016 Unknown port-component-name {0} port, all sub elements will be ignored
- 4.58 AS-DEPLOYMENT-00017 DOLUtils: converting EJB to web bundle id {0}
- 4.59 AS-DEPLOYMENT-00020 Application validation failed for application: {0}, jndi-name: {1}, resource adapter name: {2} is wrong
- 4.60 AS-DEPLOYMENT-00022 For administered-object resource: {0}, there is no application part in its resource adapter name: {1}.
- 4.61 AS-DEPLOYMENT-00023 For connection-factory resource: {0}, there is no application part in its resource adapter name: {1}.
- 4.62 AS-DEPLOYMENT-00024 For jms-connection-factory resource: {0}, there is no application part in its resource adapter name: {1}
- 4.63 AS-DEPLOYMENT-00025 For jms-destination resource: {0}, there is no application part in its resource adapter name: {1}
- 4.64 AS-DEPLOYMENT-00026 JNDI lookup failed for the resource: Name: {0}, Lookup: {1}, Type: {2}
- 4.65 AS-DEPLOYMENT-00027 Resource Adapter not present: RA Name: {0}, Type: {1}
- 4.66 AS-DEPLOYMENT-00028 Skipping resource validation
- 4.67 AS-DEPLOYMENT-02015 Skipped processing for module {0} as its module type was not recognized

- 4.68 AS-DEPLOYMENT-02016 Error occurred
- 4.69 AS-EJB-00001 [{0}]: trimLru(), resetting head and tail
- 4.70 AS-EJB-00002 [{0}]: Exception in backingStore.remove({{1}})
- 4.71 AS-EJB-00003 [{0}]: passivateEJB(), Exception caught ->
- 4.72 AS-EJB-00004 [{0}]: Cannot load from BACKUPSTORE FOR Key: {{1}}
- 4.73 AS-EJB-00005 [{0}]: Exception while loading from backup session: {{1}}
- 4.74 AS-EJB-00006 [{0}]: Error while loading from backup session: {{1}}
- 4.75 AS-EJB-00007 [{0}]: Exception during backingStore.passivateSave({{1}})
- 4.76 AS-EJB-00008 [{0}]: Iterator(), resetting head.IPrev
- 4.77 AS-EJB-00009 [{0}]: Exiting TrimTimedoutBeans() because current cache state: {{1}}
- 4.78 AS-EJB-00010 [{0}]: TrimTimedoutBeans(), resetting head.IPrev
- 4.79 AS-EJB-00011 [{0}]: Exiting TrimUnSortedTimedoutBeans() because current cache state: {{1}}
- 4.80 AS-EJB-00012 [SFSBContainer] Exception while initializing SessionSynchronization methods
- 4.81 AS-EJB-00013 [SFSBContainer] Exception while loading checkpoint info
- 4.82 AS-EJB-00014 Exception creating ejb object : {{0}}
- 4.83 AS-EJB-00015 Exception creating ejb local object {{0}}
- 4.84 AS-EJB-00016 Couldn't update timestamp for: {{0}}; Exception: {{1}}
- 4.85 AS-EJB-00017 Cannot register bean for checkpointing
- 4.86 AS-EJB-00018 Error during checkpoint ({{0}}. Key: {{1}}) {{2}}
- 4.87 AS-EJB-00019 sfsb checkpoint error. Name: {{0}}
- 4.88 AS-EJB-00020 sfsb checkpoint error. Key: {{0}}
- 4.89 AS-EJB-00021 Exception in afterCompletion : {{0}}
- 4.90 AS-EJB-00022 1. passivateEJB() returning because containerState: {{0}}
- 4.91 AS-EJB-00023 Extended EM not serializable. Exception: {{0}}
- 4.92 AS-EJB-00024 Error during passivation: {{0}}; {{1}}
- 4.93 AS-EJB-00025 Error during passivation of {{0}}
- 4.94 AS-EJB-00026 sfsb passivation error. Key: {{0}}
- 4.95 AS-EJB-00028 Error during Stateful Session Bean activation for key {{0}}
- 4.96 AS-EJB-00029 [{0}]: Error during backingStore.shutdown()
- 4.97 AS-EJB-00030 [{0}]: Error during onShutdown()
- 4.98 AS-EJB-00031 [{0}]: Error while undeploying ctx. Key: {{1}}
- 4.99 AS-EJB-00032 Cannot add idle bean cleanup task

- 4.100 AS-EJB-00033 Got exception during removeExpiredSessions (but the reaper thread is still alive)
- 4.101 AS-EJB-00034 Error during checkpoint(, but session not destroyed)
- 4.102 AS-EJB-00035 Error during checkpoint
- 4.103 AS-EJB-00036 TopLevel AvailabilityService.getAvailabilityEnabled: [{0}]
- 4.104 AS-EJB-00037 TopLevel EjbAvailabilityService.getAvailabilityEnabled: [{0}]
- 4.105 AS-EJB-00038 Global AvailabilityEnabled: [{0}], application AvailabilityEnabled: [{1}]
- 4.106 AS-EJB-00039 Exception while trying to determine availability-enabled settings for this app
- 4.107 AS-EJB-00040 StatefulContainerBuilder AvailabilityEnabled [{0}] for this application
- 4.108 AS-EJB-00041 StatefulContainerBuilder.buildStoreManager() storeName: [{0}]
- 4.109 AS-EJB-00042 Could not instantiate backing store for type [{0}]
- 4.110 AS-EJB-00043 StatefulContainerbuilder instantiated store: {0}, with ha-enabled [{1}], and backing store configuration: {2}
- 4.111 AS-EJB-00044 Error while adding idle bean passivator task
- 4.112 AS-EJB-00045 Error while adding idle bean removal task
- 4.113 AS-EJB-00046 Error while removing idle beans for [{0}]
- 4.114 AS-EJB-00047 Error while removing expired beans for [{0}]
 - 4.114.1 AS-EJB-00048 Passivation-capable value of stateful session bean [{0}] is false, it should not have any PrePassivate nor PostActivate configuration, but you have configuration at [{1}]
- 4.115 AS-EJB-00049 Cannot find stateful session bean [{0}] in memory, and will not read it from disk because current stateful session bean passivation-capable value is false
- 4.116 AS-EJB-00050 Cache is shutting down, {0} stateful session beans will not be restored after restarting since passivation is disabled
- 4.117 AS-EJB-00051 Disabling high availability for the stateful session bean {0}, as its marked non passivable
- 4.118 AS-EJB-00052 Internal Error
- 4.119 AS-EJB-00053 The feature {0} requires Full Java EE Profile to be supported
- 4.120 AS-EJB-00054 Portable JNDI names for EJB {0}: {1}
- 4.121 AS-EJB-00055 Apusic-specific (Non-portable) JNDI names for EJB {0}: {1}
- 4.122 AS-EJB-00056 A system exception occurred during an invocation on EJB {0}, method: {1}
- 4.123 AS-EJB-00057 Error while creating enterprise bean context for {0} during jacc callback

- 4.124 AS-EJB-00058 Attempt to override reserved ejb interface method [{0}] in [{1}].
Override will be ignored
- 4.125 AS-EJB-00059 Bean class for ejb [{0}] does not define a method corresponding to [{1}]
interface method [{2}]
- 4.126 AS-EJB-00060 keepstate is true and will not create new auto timers during
deployment
- 4.127 AS-EJB-00061 Failed to initialize the interceptor
- 4.128 *AS-EJB-00062 [*BaseContainer*] Could not create MonitorRegistryMediator. [{0}]**
- 4.129 AS-EJB-00063 Exception in backingStore.size()
- 4.130 AS-JMS-CORE-00001 JMS Service Connection URL is : {0}
- 4.131 AS-JMS-CORE-00002 ADDRESSLIST in setJmsServiceProvider : {0}
- 4.132 AS-JMS-CORE-00003 Addresslist : {0}
- 4.133 AS-JMS-CORE-00004 End point determines destination name, Res name: {0}, JNDI
name: {1} descriptor name : {2}
- 4.134 AS-JMS-CORE-00005 Successfully set Master broker on JMSRA to {0}
- 4.135 AS-JMS-CORE-00006 Failed to set Master broker on JMSRA to {0} cause {1}
- 4.136 AS-JMS-CORE-00007 Successfully set Cluster brokerlist to {0}
- 4.137 AS-JMS-CORE-00008 Failed to set Cluster brokerlist to {0} cause {1}
- 4.138 AS-JMS-CORE-00009 Failed to shut down Grizzly NetworkListener : {0}
- 4.139 AS-JMS-CORE-00010 Error occurs when shutting down JMSRA : {0}
- 4.140 AS-JMS-CORE-00011 Invalid RMI registry port
- 4.141 AS-JMS-CORE-00012 No such method {0} in the class {1}
- 4.142 AS-JMS-CORE-00013 Connector Resource could not be closed
- 4.143 AS-JMS-CORE-00014 rardeployment.mcfcreation_error {0}
- 4.144 AS-JMS-CORE-00015 Exception while getting configured RMI port : {0}
- 4.145 AS-JMS-CORE-00016 Failed to start Grizzly proxy for MQ broker
- 4.146 AS-JMS-CORE-00017 Failed to create addresslist due to the exception : {0}
- 4.147 AS-JMS-CORE-00018 Error executing method {0} of the class {1}
- 4.148 AS-JMS-CORE-00019 MDB destination not specified
- 4.149 AS-JMS-CORE-00020 Failed to validate endpoint
- 4.150 AS-JMS-CORE-00021 Cannot obtain master broker
- 4.151 AS-JMS-CORE-00022 Error while loading connector resources during recovery : {0}
- 4.152 AS-JMS-CORE-00023 Exception in reading mdb-container configuration : [{0}]
- 4.153 AS-JMS-CORE-00024 MQ Resource adapter upgrade started

- 4.154 AS-JMS-CORE-00025 MQ Resource adapter upgrade completed
- 4.155 AS-JMS-CORE-00026 Upgrading a MQ resource adapter failed : {0}
- 4.156 AS-JMS-CORE-00027 Check for a new version of MQ installation failed : {0}
- 4.157 AS-ORB-00001 Exception occurred when resolving {0}
- 4.158 AS-ORB-00002 No Endpoints selected in com.sun.appserv.iioop.endpoints property.Using {0}:{1} instead
- 4.159 AS-ORB-00003 Problem with membership change notification. Exception occurred : {0}
- 4.160 AS-ORB-00004 Could not find an endpoint to send request to
- 4.161 AS-ORB-00005 Unknown host: {0} Exception thrown : {1}
- 4.162 AS-ORB-00006 No Endpoints selected in com.sun.appserv.iioop.endpoints property. Using JNDI Provider URL {0} instead
- 4.163 AS-ORB-00007 Exception : {0} thrown for bad provider URL String: {1}
- 4.164 AS-WSSSECURITY-00001 Request processing failed
- 4.165 AS-WSSSECURITY-00002 SEC2002: Container-auth: wss: Error validating request
- 4.166 AS-WSSSECURITY-00003 SEC2003: Container-auth: wss: Error securing response
- 4.167 AS-WSSSECURITY-00004 SEC2004: Container-auth: wss: Error securing request
- 4.168 AS-WSSSECURITY-00005 SEC2005: Container-auth: wss: Error validating response
- 4.169 AS-WSSSECURITY-00006 SEC2006: Container-auth: wss: Not a SOAP message context
- 4.170 AS-WSSSECURITY-00007 EJB Webservice security configuration Failure
- 4.171 AS-WSSSECURITY-00008 Servlet Webservice security configuration Failure
- 4.172 AS-WSSSECURITY-00009 BASIC AUTH username/password http header parsing error for {0}
- 4.173 AS-WSSSECURITY-00010 Servlet Webservice security configuration Failure
- 4.174 AS-WSSSECURITY-00011 Following exception was thrown:
- 4.175 AS-WEB-ADMIN-00001 Unable to register StatsProvider {0} with Monitoring Infrastructure. No monitoring data will be collected for {1} and {2}
- 4.176 AS-WEB-ADMIN-00002 Current server config is null
- 4.177 AS-WEB-ADMIN-00003 The acceptor threads must be at least 1
- 4.178 AS-WEB-ADMIN-00004 Listener {0} could not be created, actual reason: {1}
- 4.179 AS-WEB-ADMIN-00005 A default virtual server is required. Please use --default-virtual-server to specify this value
- 4.180 AS-WEB-ADMIN-00006 --defaultVS and --default-virtual-server conflict. Please use only --default-virtual-server to specify this value

- 4.181 AS-WEB-ADMIN-00007 Attribute value (default-virtual-server = {0}) is not found in list of virtual servers defined in config
- 4.182 AS-WEB-ADMIN-00008 Http Listener named {0} already exists
- 4.183 AS-WEB-ADMIN-00009 Port [{0}] is already taken for address [{1}], please choose another port
- 4.184 AS-WEB-ADMIN-00010 Network Listener named {0} already exists.
- 4.185 AS-WEB-ADMIN-00011 Protocol {0} has neither a protocol nor a port-unification configured
- 4.186 AS-WEB-ADMIN-00012 {0} create failed
- 4.187 AS-WEB-ADMIN-00013 The specified protocol {0} is not yet configured
- 4.188 AS-WEB-ADMIN-00014 Failed to create http-redirect for {0}: {1}
- 4.189 AS-WEB-ADMIN-00015 An http element for {0} already exists. Cannot add duplicate http
- 4.190 AS-WEB-ADMIN-00016 An http-redirect element for {0} already exists. Cannot add duplicate http-redirect
- 4.191 AS-WEB-ADMIN-00017 {0} protocol already exists. Cannot add duplicate protocol
- 4.192 AS-WEB-ADMIN-00018 Failed to create protocol {0}
- 4.193 AS-WEB-ADMIN-00019 {0} create failed: {1}
- 4.194 AS-WEB-ADMIN-00020 {0} create failed. Given class is not a ProtocolFilter: {1}
- 4.195 AS-WEB-ADMIN-00021 {0} create failed. Given class is not a ProtocolFinder: {1}
- 4.196 AS-WEB-ADMIN-00022 {0} transport already exists. Cannot add duplicate transport
- 4.197 AS-WEB-ADMIN-00023 Failed to create transport {0}
- 4.198 AS-WEB-ADMIN-00024 Please use only networklisteners
- 4.199 AS-WEB-ADMIN-00025 Virtual Server named {0} already exists
- 4.200 AS-WEB-ADMIN-00026 {0} create failed
- 4.201 AS-WEB-ADMIN-00028 Specified http listener, {0}, doesn't exist
- 4.202 AS-WEB-ADMIN-00029 {0} delete failed
- 4.203 AS-WEB-ADMIN-00030 {0} Network Listener doesn't exist
- 4.204 AS-WEB-ADMIN-00031 Deletion of NetworkListener {0} failed
- 4.205 AS-WEB-ADMIN-00032 {0} http-redirect doesn't exist
- 4.206 AS-WEB-ADMIN-00033 Deletion of http {0} failed
- 4.207 AS-WEB-ADMIN-00034 Deletion of http-redirect {0} failed
- 4.208 AS-WEB-ADMIN-00035 {0} protocol doesn't exist
- 4.209 AS-WEB-ADMIN-00036 {0} protocol is being used in the network listener {1}

- 4.210 AS-WEB-ADMIN-00037 Deletion of Protocol {0} failed
- 4.211 AS-WEB-ADMIN-00038 {0} delete failed: {1}.
- 4.212 AS-WEB-ADMIN-00039 No {0} element found with the name {1}
- 4.213 AS-WEB-ADMIN-00040 {0} transport is being used in the network listener {1}
- 4.214 AS-WEB-ADMIN-00041 Deletion of Transport {0} failed
- 4.215 AS-WEB-ADMIN-00042 {0} transport doesn't exist
- 4.216 AS-WEB-ADMIN-00043 {0} delete failed
- 4.217 AS-WEB-ADMIN-00044 Specified virtual server, {0}, doesn't exist
- 4.218 AS-WEB-ADMIN-00045 Specified virtual server, {0}, can not be deleted because it is referenced from http listener, {1}
- 4.219 AS-WEB-ADMIN-00046 Monitoring Registry does not exist. Possible causes are 1) Monitoring is not turned on or at a lower level 2) The corresponding container (web, ejb, etc.) is not loaded yet
- 4.220 AS-WEB-UTIL-00001 Resource "{0}" is missing
- 4.221 AS-WEB-UTIL-00002 Failed tracking modifications of "{0}" : {1}
- 4.222 AS-WEB-UTIL-00003 WebappClassLoader.findClassInternal({0}) security exception: {1}
- 4.223 AS-WEB-UTIL-00004 Security Violation, attempt to use Restricted Class: {0}
- 4.224 AS-WEB-UTIL-00005 Class {0} has unsupported major or minor version numbers, which are greater than those found in the Java Runtime Environment version {1}
- 4.225 AS-WEB-UTIL-00006 Unable to load class with name [{0}], reason: {1}
- 4.226 AS-WEB-UTIL-00007 The web application [{0}] registered the JDBC driver [{1}] but failed to unregister it when the web application was stopped. To prevent a memory leak, the JDBC Driver has been forcibly unregistered
- 4.227 AS-WEB-UTIL-00008 JDBC driver de-registration failed for web application [{0}]
- 4.228 AS-WEB-UTIL-00009 Exception closing input stream during JDBC driver de-registration for web application [{0}]
- 4.229 AS-WEB-UTIL-00010 This web container has not yet been started
- 4.230 AS-WEB-UTIL-00011 Failed to check for ThreadLocal references for web application [{0}]
- 4.231 AS-WEB-UTIL-00012 Unable to determine string representation of key of type [{0}]
- 4.232 AS-WEB-UTIL-00013 Unknown
- 4.233 AS-WEB-UTIL-00014 Unable to determine string representation of value of type [{0}]
- 4.234 AS-WEB-UTIL-00015 The web application [{0}] created a ThreadLocal with key of type [{1}] (value [{2}]). The ThreadLocal has been correctly set to null and the key will be removed

by GC

- 4.235 AS-WEB-UTIL-00016 The web application [{0}] created a ThreadLocal with key of type [{1}] (value [{2}]) and a value of type [{3}] (value [{4}]) but failed to remove it when the web application was stopped. Threads are going to be renewed over time to try and avoid a probable memory leak
- 4.236 AS-WEB-UTIL-00017 Failed to find class sun.rmi.transport.Target to clear context class loader for web application [{0}]. This is expected on non-Sun JVMs
- 4.237 AS-WEB-UTIL-00018 Failed to clear context class loader referenced from sun.rmi.transport.Target for web application [{0}]
- 4.238 AS-WEB-UTIL-00019 Removed [{0}] ResourceBundle references from the cache for web application [{1}]
- 4.239 AS-WEB-UTIL-00020 Failed to clear ResourceBundle references for web application [{0}]
- 4.240 AS-WEB-UTIL-00021 Illegal JAR entry detected with name {0}
- 4.241 AS-WEB-UTIL-00022 Unable to validate JAR entry with name {0}
- 4.242 AS-WEB-UTIL-00023 Unable to create {0}
- 4.243 AS-WEB-UTIL-00024 Unable to delete {0}
- 4.244 AS-WEB-UTIL-00025 Unable to read data for class with name [{0}]
- 4.245 AS-WEB-UTIL-00026 Unable to purge bean classes from BeanELResolver
- 4.246 AS-WEB-UTIL-00027 extra-class-path component [{0}] is not a valid pathname
- 4.247 AS-WEB-UTIL-00028 The clearReferencesStatic is not consistent in context.xml for virtual servers
- 4.248 AS-WEB-UTIL-00029 class-loader attribute dynamic-reload-interval in sun-web.xml not supported
- 4.249 AS-WEB-UTIL-00030 Property element in sun-web.xml has null 'name' or 'value'
- 4.250 AS-WEB-UTIL-00031 Ignoring invalid property [{0}] = [{1}]
- 4.251 AS-WEB-UTIL-00032 The xml element should be [{0}] rather than [{1}]
- 4.252 AS-WEB-UTIL-00033 This is an unexpected end of document
- 4.253 AS-WEB-UTIL-00034 Unexpected type of ClassLoader. Expected: java.net.URLClassLoader, got: {0}
- 4.254 AS-WEB-UTIL-00035 Unable to load class {0}, reason: {1}
- 4.255 AS-WEB-UTIL-00036 Invalid URLClassLoader path component: [{0}] is neither a JAR file nor a directory

- 4.256 AS-WEB-UTIL-00037 Error trying to scan the classes at {0} for annotations in which a ServletContainerInitializer has expressed interest
- 4.257 AS-WEB-UTIL-00038 Ignoring [{0}] during Tag Library Descriptor (TLD) processing
- 4.258 AS-WEB-UTIL-00039 Unable to determine TLD resources for [{0}] tag library, because class loader [{1}] for [{2}] is not an instance of java.net.URLClassLoader
- 4.259 AS-WEB-CORE-00001 Configuration error: Must be attached to a Context
- 4.260 AS-WEB-CORE-00002 Authenticator[{0}]: {1}
- 4.261 AS-WEB-CORE-00003 Exception getting debug value
- 4.262 AS-WEB-CORE-00004 Unexpected error forwarding or redirecting to login page
- 4.263 AS-WEB-CORE-00005 Started
- 4.264 AS-WEB-CORE-00006 Stopped
- 4.265 AS-WEB-CORE-00007 Process session destroyed on {0}
- 4.266 AS-WEB-CORE-00008 Process request for "{0}"
- 4.267 AS-WEB-CORE-00009 Principal {0} has already been authenticated
- 4.268 AS-WEB-CORE-00010 Checking for SSO cookie
- 4.269 AS-WEB-CORE-00011 SSO cookie is not present
- 4.270 AS-WEB-CORE-00012 Checking for cached principal for {0}
- 4.271 AS-WEB-CORE-00013 Found cached principal {0} with auth type {1}
- 4.272 AS-WEB-CORE-00014 No cached principal found, erasing SSO cookie
- 4.273 AS-WEB-CORE-00015 Associate sso id {0} with session {1}
- 4.274 AS-WEB-CORE-00016 Registering sso id {0} for user {1} with auth type {2}
- 4.275 AS-WEB-CORE-00017 Looking up certificates
- 4.276 AS-WEB-CORE-00018 No certificates included with this request
- 4.277 AS-WEB-CORE-00019 No client certificate chain in this request
- 4.278 AS-WEB-CORE-00020 Cannot authenticate with the provided credentials
- 4.279 AS-WEB-CORE-00021 Unable to determine target of zero-arg dispatcher
- 4.280 AS-WEB-CORE-00022 Unable to acquire RequestDispatcher for {0}
- 4.281 AS-WEB-CORE-00023 Unable to acquire RequestDispatcher for {0} in servlet context {1}
- 4.282 AS-WEB-CORE-00024 Error invoking AsyncListener
- 4.283 AS-WEB-CORE-00025 Asynchronous dispatch already in progress, must call ServletRequest.startAsync first
- 4.284 AS-WEB-CORE-00026 Must not call AsyncContext.addListener after the container-initiated dispatch during which ServletRequest.startAsync was called has returned to the

container

- 4.285 AS-WEB-CORE-00027 Must not call AsyncContext.setTimeout after the container-initiated dispatch during which ServletRequest.startAsync was called has returned to the container
- 4.286 AS-WEB-CORE-00028 The connector has already been initialized
- 4.287 AS-WEB-CORE-00029 Error registering connector
- 4.288 AS-WEB-CORE-00030 Failed to instantiate HttpHandler
- 4.289 AS-WEB-CORE-00031 mod_jk invalid Adapter implementation: {0}
- 4.290 AS-WEB-CORE-00032 Protocol handler instantiation failed: {0}
- 4.291 AS-WEB-CORE-00033 The connector has already been started
- 4.292 AS-WEB-CORE-00034 Protocol handler start failed: {0}
- 4.293 AS-WEB-CORE-00035 Coyote connector has not been started
- 4.294 AS-WEB-CORE-00036 Protocol handler destroy failed: {0}
- 4.295 AS-WEB-CORE-00037 An exception or error occurred in the container during the request processing
- 4.296 AS-WEB-CORE-00038 HTTP listener on port {0} has been disabled
- 4.297 AS-WEB-CORE-00039 Error parsing client cert chain into array of java.security.cert.X509Certificate instances
- 4.298 AS-WEB-CORE-00040 No Host matches server name {0}
- 4.299 AS-WEB-CORE-00041 Cannot use this object outside a servlet's service method or outside a filter's doFilter method
- 4.300 AS-WEB-CORE-00042 Cannot set a null ReadListener object
- 4.301 AS-WEB-CORE-00043 Cannot set a null WriteListener object
- 4.302 AS-WEB-CORE-00044 Failed to skip {0} characters in the underlying buffer of CoyoteReader on readLine()
- 4.303 AS-WEB-CORE-00045 Stream closed
- 4.304 AS-WEB-CORE-00046 Already set read listener
- 4.305 AS-WEB-CORE-00047 Cannot set ReaderListener for non-async or non-upgrade request
- 4.306 AS-WEB-CORE-00048 Error in invoking ReadListener.onDataAvailable
- 4.307 AS-WEB-CORE-00049 The WriteListener has already been set
- 4.308 AS-WEB-CORE-00050 Cannot set WriteListener for non-async or non-upgrade request
- 4.309 AS-WEB-CORE-00051 Error in invoking WriteListener.onWritePossible

- 4.310 AS-WEB-CORE-00052 `getReader()` has already been called for this request
- 4.311 AS-WEB-CORE-00053 `getInputStream()` has already been called for this request
- 4.312 AS-WEB-CORE-00054 Unable to determine client remote address from proxy (returns null)
- 4.313 AS-WEB-CORE-00055 Unable to resolve IP address {0} into host name
- 4.314 AS-WEB-CORE-00056 Exception thrown by attributes event listener
- 4.315 AS-WEB-CORE-00057 Cannot call `setAttribute` with a null name
- 4.316 AS-WEB-CORE-00058 Unable to determine canonical name of file [{0}] specified for use with `sendfile`
- 4.317 AS-WEB-CORE-00059 Unable to set request character encoding to {0} from context {1}, because request parameters have already been read, or `ServletRequest.getReader()` has already been called
- 4.318 AS-WEB-CORE-00060 Attempt to re-login while the user identity already exists
- 4.319 AS-WEB-CORE-00061 `changeSessionId` has been called without a session
- 4.320 AS-WEB-CORE-00062 Cannot create a session after the response has been committed
- 4.321 AS-WEB-CORE-00063 Invalid URI encoding; using HTTP default
- 4.322 AS-WEB-CORE-00064 Invalid URI character encoding; trying `ascii`
- 4.323 AS-WEB-CORE-00065 Request is within the scope of a filter or servlet that does not support asynchronous operations
- 4.324 AS-WEB-CORE-00066 `ServletRequest.startAsync` called again without any asynchronous dispatch, or called outside the scope of any such dispatch, or called again within the scope of the same dispatch
- 4.325 AS-WEB-CORE-00067 Response already closed
- 4.326 AS-WEB-CORE-00068 `ServletRequest.startAsync` called outside the scope of an `async` dispatch
- 4.327 AS-WEB-CORE-00069 The request has not been put into asynchronous mode, must call `ServletRequest.startAsync` first
- 4.328 AS-WEB-CORE-00070 Request already released from asynchronous mode
- 4.329 AS-WEB-CORE-00071 Unable to perform error dispatch
- 4.330 AS-WEB-CORE-00072 `Request.{0}` is called without multipart configuration. Either add a `@MultipartConfig` to the servlet, or a `multipart-config` element to `web.xml`
- 4.331 AS-WEB-CORE-00073 This should not happen-breaking background lock: `sess = {0}`

- 4.332 AS-WEB-CORE-00074 Must not use request object outside the scope of a servlet's service or a filter's doFilter method
- 4.333 AS-WEB-CORE-00075 Error during finishResponse
- 4.334 AS-WEB-CORE-00076 getWriter() has already been called for this response
- 4.335 AS-WEB-CORE-00077 getOutputStream() has already been called for this response
- 4.336 AS-WEB-CORE-00078 Cannot reset buffer after response has been committed
- 4.337 AS-WEB-CORE-00079 Cannot change buffer size after data has been written
- 4.338 AS-WEB-CORE-00080 Cannot call sendError() after the response has been committed
- 4.339 AS-WEB-CORE-00081 Cannot call sendRedirect() after the response has been committed
- 4.340 AS-WEB-CORE-00082 Null response object
- 4.341 AS-WEB-CORE-00083 Not allowed to call this javax.servlet.ServletContext method from a ServletContextListener that was neither declared in the application's deployment descriptor nor annotated with WebListener

1 更新历史

日期	手册版本	适用产品	更新说明
2026年5月	V10P11E08F01	AAS V10	新建

2 前言

本文档详细介绍使用金蝶Apusic应用服务器V10时可能遇到的问题。

3 概述

3.1 产品介绍

3.2 字符说明

下表说明本文档可能使用的相关字符。

符号	说明	示例	示例解释
[]	包含可选参数和命令选项	ls [-l]	-l选项不是必需的
{}	包含一组必需命令选项的选择	-d {y n}	使用 -d 选项时, 必须同时指定 y 参数或 n 参数
\${}	表示变量引用	\${com.sun.javaRoot}	引用com.sun.javaRoot变量的值
>	在图形用户界面中指示菜单项选择	【资源管理】 > 【JDBC】 > 【JDBC连接池】	从【资源管理】模块中选择【JDBC】模块, 再选择【JDBC连接池】模块

3.3 默认路径

下表说明本文档中使用的默认路径和文件名。

名称	说明	默认值
aas-install	安装Apusic应用服务器的目录	user's-home-directory/ApusicAS/aas
aas-install-parent	安装Apusic应用服务器的目录的父目录	user's-home-directory/ApusicAS
domain-root-dir	创建域的默认目录	aas-install/domains
domain-dir	存储域配置的目录	domain-root-dir/domain-name
instance-dir	服务器实例的目录	domain-dir/instance-name

4 错误码说明

本章汇总了使用金蝶Apusic应用服务器V10时可能出现的错误信息。

这些错误信息会详细说明服务器运行过程中发生的各类事件，例如配置错误、安全漏洞或服务器故障。通常还会提供相应的诊断和故障排除指南。

4.1 AS-ACDEPL-00103 Java Web Start services started for the app client {0} (contextRoot: {1})

原因：Java Web Start (JWS) 服务已为指定的应用客户端启动。该日志为信息级别 (INFO)，仅表示 JWS 服务已成功启动，供管理员确认客户端部署状态，无错误或警告含义。

处理方案：若无需使用 Java Web Start 功能（如客户端通过其他方式连接服务器），该日志可忽略，不影响应用核心功能。若客户端启动失败（如无法通过 JWS 下载），需结合其他日志（如错误级别的 AS-ACDEPL-xxxx 日志）排查，可能与客户端配置、权限或网络有关。

4.2 AS-ACDEPL-00105 Error starting the adapter to serve static system-level content

原因：发生意外的内部系统错误。表明用于处理“系统级静态资源”的适配器启动失败，可能影响服务器的部分基础功能（如管理控制台静态资源加载、内置系统页面访问等）。

处理方案：请查阅异常堆栈跟踪。

4.3 AS-ACDEPL-00108 Config file for client JNLP not found: {0}

原因：表明在处理 Java Web Start (JWS) 客户端时，未找到指定的 JNLP 配置文件（{0} 为缺失的文件路径占位符）。

处理方案：需通过日志定位具体缺失路径，优先尝试重新部署应用客户端以触发文件生成，同时检查配置完整性、文件权限和磁盘空间。若为自定义路径配置错误，需修正路径或重置为默认配置，确保 JNLP 文件能被服务器正确生成和访问。

4.4 AS-ACDEPL-00109 The config file for client JNLP {0} exists but could not be read.

原因：表明 Java Web Start (JWS) 客户端的 JNLP 配置文件存在 ({0} 为文件路径) ，但服务器无法读取该文件，导致客户端无法通过 JWS 方式正常启动。

处理方案：优先通过 `ls -l` 检查文件权限和所有者，确保 AAS运行用户有读取权限；若权限正确，进一步排查文件系统限制（如 SELinux、只读分区）或文件损坏问题。重新部署客户端是快速修复文件损坏的有效方式，可优先尝试。

4.5 AS-ACDEPL-00111 Client JNLP document {0} refers to the static resource {1} that does not exist or is not readable

原因：AAS在处理 Java Web Start (JWS) 客户端时输出的错误日志，表明 JNLP 配置文件 ({0}) 中引用的某个静态资源 ({1}) ，如 JAR 包、图片、配置文件等) 不存在或无法读取，导致客户端无法通过 JWS 正常下载和启动。

处理方案：需先通过日志定位具体资源和 JNLP 文件，再分情况处理：资源缺失则补充文件或修正路径，权限问题则修复文件 / 目录权限，配置不一致则调整 `context-root` 或 JNLP 路径。重新部署应用通常能解决大部分因部署不完整导致的资源缺失问题，可优先尝试。

4.6 AS-ACDEPL-00112 The following extensions or libraries are referenced from the manifest of {0} but were not found where indicated: {1}; ignoring and continuing

原因：AAS在部署应用时输出的警告级日志，表明应用的 JAR 包清单文件 (`MANIFEST.MF`) 中引用了某些扩展或依赖库，但服务器在指定位置未找到这些库。

处理方案：补充必要依赖、修正路径声明、清理冗余依赖。若应用运行正常，可暂时忽略；若出现类缺失错误，需优先解决依赖问题，避免影响功能可用性。

4.7 AS-ACDEPL-00114 The ApplicationSignedJARManager for a nested app client deployer helper is unexpectedly null

原因：AAS在部署嵌套式应用客户端 (nested app client 时输出的错误日志，表明负责验证签名 JAR 包的 `ApplicationSignedJARManager` 组件未正常初始化 (为 `null`) ，导致部署过程异常。该错误通常与应用客户端的签名验证、嵌套部署配置或服务器组件故障相关。

处理方案：需从应用签名有效性、EAR 包结构、服务器组件完整性三个方向排查。优先通过验证 JAR 签名、清理服务器缓存、重新打包应用等方式尝试解决；若涉及服务器组件损坏，重新安装AAS是最直接的修复手段。

4.8 AS-ACDEPL-00115 The custom JNLP document {0} in a stand-alone app client incorrectly refers to a JAR {1}

原因：AAS在部署 独立应用客户端（stand-alone app client）时输出的错误日志，表明客户端自定义的 JNLP 配置文件（{0}）中引用的 JAR 包（{1}）存在配置错误（如路径无效、与客户端实际部署的 JAR 不匹配），导致 Java Web Start（JWS）无法正确加载客户端依赖。

处理方案：需重点验证 JAR 路径的正确性、与实际部署文件的一致性，以及 JNLP 格式的规范性。修正路径引用并重新部署后，通常可解决该问题，确保 JWS 能正确加载客户端依赖的 JAR 包。

4.9 AS-ACDEPL-00101 Java Web Start services not started for the app client {0}; its developer has marked it as ineligible

原因：AAS输出的提示性日志，表明 指定的应用客户端（{0}）未启动 Java Web Start（JWS）服务，原因是开发者在客户端配置中明确标记该客户端“不适合”使用 JWS 方式启动。

处理方案：若需通过 JWS 启动客户端，需检查并修改部署描述符、注解或清单文件中的相关配置，启用 JWS 访问；若无需 JWS，可忽略此日志。

4.10 AS_ACDEPL_00102 Java Web Start services not started for the app client {0}; the administrator has disabled Java Web Start support for it

原因：在目标应用客户端（{0}）的部署配置中，「JWS 支持」被手动禁用（默认是启用的），导致 JWS 服务无法为其启动。

处理方案：优先通过控制台或命令行启用即可解决。若需长期禁用 JWS，可采用替代部署方式避免依赖 JWS 服务。

4.11 AS_ACDEPL-00104 Java Web Start services stopped for the app client {0}

原因：Java Web Start（JWS）服务停止或应用客户端部署异常。

处理方案：优先通过「重启服务」「检查部署状态」「清理缓存」解决；若持续报错，结合日志定位具体原因（权限、依赖、配置或环境问题）。

4.12 AS-NAMING-00001 Failed to load {0} using CommonClassLoader

原因：AAS在启动或处理命名服务（Naming Service，如 JNDI 服务）时输出的错误日志，表明服务器的 CommonClassLoader（公共类加载器）无法加载指定的类或资源（{0} 为无法加载的类名或资源路径），可能导致命名服务初始化失败或相关功能（如 JNDI 查找）异常。

处理方案：需从类 / 资源完整性、类加载冲突、权限路径、服务器模块完整性四个方向排查。优先通过补充缺失的 JAR 包、修复权限、解决冲突类等方式处理；若涉及核心模块损坏，重新安装AAS是最可靠的解决手段。

4.13 AS-NAMING-00002 Exception during name lookup : {0}

原因：在处理 JNDI（Java Naming and Directory Interface）命名服务查询时输出的错误日志，表明在尝试查找某个 JNDI 名称（如 EJB 组件、数据源、环境变量等）时发生异常，{0} 为具体的异常信息（如 NameNotFoundException 或连接超时等）。该错误会导致依赖 JNDI 查找的功能（如 EJB 调用、数据库连接）失败。

处理方案：需根据 {0} 中的具体异常信息定位原因：名称错误则修正名称，服务未启动则重启服务，资源依赖问题则修复资源配置，权限问题则调整安全策略。通过 管理控制台的 JNDI 浏览器确认名称存在性，是快速排查名称正确性的有效手段。

4.14 AS-NAMING-00003 Exception occurred when instantiating LocalSerialContextProviderImpl: {0}

原因：AAS在初始化 本地命名服务提供者（LocalSerialContextProviderImpl）时输出的错误日志，表明该核心组件实例化失败，{0} 为具体的异常原因（如类缺失、依赖故障、配置错误等）。这会导致 AAS 的 JNDI 命名服务初始化失败，进而影响整个服务器的启动或资源访问（如 EJB、数据源等无法通过 JNDI 查找）。

处理方案：需根据 {0} 中的异常信息，从核心 JAR 包完整性、配置文件正确性、类加载冲突、权限资源等方面排查。优先通过替换损坏的 naming.jar、修复 domain.xml 配置、清理缓存等方式处理；若问题持续，重新安装 AAS是解决服务器核心组件故障的可靠手段。

4.15 AS-NAMING-00004 Error during CosNaming.unbind for name {0}: {1}

原因：AAS在执行 JNDI 名称解绑（unbind）操作时输出的错误日志，表明在尝试从命名服务（CosNaming，CORBA 命名服务的实现）中移除某个 JNDI 名称（{0}）时发生异常，{1} 为具体的错误原因（如名称不存在、权限不足、连接异常等）。该错误通常发生在应用卸载、资源删除或服务器关闭阶段，可能导致命名服务中残留无效绑定，影响后续资源注册。

处理方案：需根据 {1} 中的具体异常定位原因：名称不存在可忽略，权限问题需调整安全配置，服务异常需重启服务，资源锁定需先停止依赖应用。清理命名服务缓存或重建域是解决数据损坏的有效手段，确保命名空间状态一致。

4.16 AS-NAMING-00005 Naming binding already exists for {0} in namespace {1}

原因：AAS在注册 JNDI 名称（绑定操作）时输出的错误日志，表明尝试向命名空间（{1}）中注册的 JNDI 名称（{0}）已存在，导致新的绑定操作失败。这通常发生在应用部署、资源创建或服务初始化阶段，会影响新资源的注册或应用的正常部署

处理方案：需通过管理工具定位冲突来源，解除重复绑定或修改名称确保唯一性。清理缓存和重启服务器可解决部署残留导致的冲突，而在多应用或集群环境中，制定统一的 JNDI 命名规范（如包含应用前缀）是预防此类错误的键。

4.17 AS-NAMING-00006 Exception in NamingManagerImpl copyMutableObject(): {0}

原因：AAS在通过 NamingManagerImpl 的 `copyMutableObject()` 方法处理 JNDI 绑定的可变对象复制时输出的错误日志，表明对象复制过程中发生异常（{0} 为具体异常信息，如序列化失败、类转换错误等）。该错误通常影响 JNDI 资源的访问或传递（如远程调用中对象的复制），可能导致资源获取失败或数据不一致

处理方案：需根据 {0} 中的异常类型，针对性修复对象的序列化能力、类构造函数、状态有效性或类加载逻辑。确保对象可序列化且结构简单、状态有效，是避免此类错误的键。若问题持续，需检查自定义对象的复制逻辑是否与 AAS命名服务的预期行为兼容。

4.18 AS-NAMING-00008 Fall back to INITIAL_CONTEXT_FACTORY {0}

原因：AAS在初始化 JNDI 上下文（`InitialContext`）时输出的提示性日志，表明默认的 JNDI 初始上下文工厂（Initial Context Factory）未正常工作或未配置，服务器已回退到备选的上下文工厂实现（{0} 为备选工厂类名，如 `com.sun.enterprise.naming.impl.SerialInitContextFactory`）。

处理方案：若应用功能正常，无需处理；若存在 JNDI 相关错误，需检查初始上下文工厂的配置有效性及依赖服务状态，必要时使用AAS内置工厂类以保证兼容性。

4.19 AS-CONCURRENT-00001 Task [{0}] has been running on thread [{1}] for {2} seconds,which is more than the configured hung task threshold of {3} seconds in [{4}].

原因：AAS输出的警告日志，用于提示某个任务（{0}）在指定线程（{1}）上的运行时间（{2} 秒）超过了配置的“挂起任务阈值”（{3} 秒），可能存在任务执行缓慢或“挂起”（hung）的风险。{4} 通常指该阈值对应的线程池或执行器配置名称。

处理方案：需结合线程栈分析任务状态，区分是“正常耗时”还是“异常阻塞”。优化任务逻辑、解决外部依赖瓶颈、调整线程池配置是主要处理方向。及时处理可避免线程资源耗尽或功能无响应，保障应用的并发稳定性。

4.20 AS-CONCURRENT-00002 Unable to setup or reset runtime context for a task because an invalid context handle is being passed

原因：AAS在处理并发任务时输出的错误日志，表明服务器在为任务 设置或重置运行时上下文（如安全上下文、命名上下文、事务上下文等）时失败，核心原因是传入的 上下文句柄（context handle）无效。该错误会导致任务无法正常执行，可能引发任务中断、功能异常或并发处理故障。

处理方案：需从句柄有效性、传递逻辑、组件配置三个维度排查。优先通过规范上下文句柄的生成与清理、修复上下文服务配置、启用线程池上下文传播功能解决；若涉及分布式场景，需确保句柄可序列化且跨节点配置一致。及时处理可避免任务执行中断，保障并发任务的稳定运行。

4.21 AS-CONCURRENT-00003 Unable to bind {0} to JNDI location [{1}]

原因：尝试将托管对象绑定到 JNDI 命名空间时发生意外异常。

处理方案：查看异常消息以确定故障原因并采取相应措施。

4.22 AS-CONCURRENT-00004 Unable to deploy {0}

原因：通常与并发资源（如 `ManagedExecutorService`、`ManagedScheduledExecutorService`、`ManagedThreadFactory` 等）的部署过程失败有关。该错误一般是由于资源配置不当、依赖缺失或服务器状态异常导致的。

处理方案：通过定位具体资源、检查配置参数、排查依赖、验证JNDI名称、修复服务器状态逐一排查问题。

4.23 AS-RAR-05001 Failed to get the thread-pool [{0}] for resource adapter [{1}].

原因：根据池 ID 找不到线程池。

处理方案：请检查资源适配器配置中的线程池ID属性。

4.24 AS-RAR-05002 Failed to get the default thread-pool for resource adapter [{0}].

原因：表明资源适配器（{0} 所指的 JCA 适配器，如连接消息队列、外部服务的组件）无法获取默认线程池。这通常是由于默认线程池缺失、配置异常或资源适配器依赖问题导致的。

处理方案：请检查资源适配器配置中的线程池ID属性。

4.25 AS-RAR-05003: An error occurred during instantiation of the Work Manager class [{0}] for resource adapter [{1}]

原因：表明资源适配器（{1} 所指的 JCA 适配器）在实例化 Work Manager 类（{0} 所指的具体类）时失败。Work Manager 是 JCA 规范中负责管理工作线程、调度任务的核心组件，其实例化失败通常与类加载问题、依赖缺失、配置错误或代码兼容性有关。

处理方案：检查工作管理器类类型。

4.26 AS-RAR-05004 The Work named [{0}], progress [{1}].

原因：通常是一条状态信息（非错误），用于记录资源适配器（JCA 适配器）中某个 Work 任务的执行进度。该信息主要用于跟踪 Work 任务的运行状态，帮助排查资源适配器的任务调度情况。

处理方案：本身是正常的状态日志，主要用于跟踪 Work 任务进度，无需特殊处理；若任务状态异常（如频繁失败），则需结合具体日志和配置进一步分析。

4.27 AS-RAR-05005 Resource adapter association failed

原因：资源适配器在 ManagedConnectionFactory.setResourceAdapter () 期间抛出异常。

处理方案：该错误通常与资源适配器的部署配置或环境兼容性相关，通过逐步排查配置、依赖和服务器状态，可定位并解决问题。

4.28 AS-RAR-05006 Unable to load Work Context class {0}

原因：表明资源适配器（JCA 适配器）在加载指定的 Work Context 类（{0} 所指的具体类）时失败。Work Context 是 JCA 规范中用于在 Work 任务（异步任务）执行时传递上下文信息（如安全上下文、事务上下文等）的组件，其加载失败通常与类缺失、类加载问题或依赖冲突有关。

处理方案：从错误信息中获取 {0}（Work Context 类名），确认该类是否存在于资源适配器的部署包中；检查 `server.log` 堆栈跟踪，定位具体失败原因（如类缺失、依赖缺失、权限问题）；验证类加载配置（如隔离策略）和依赖是否完整，补充缺失的 JAR 包；确认 Work Context 类符合 JCA 规范，且与 AAS 版本兼容；清理服务器缓存或重启，排除类加载器故障。若涉及第三方资源适配器，建议参考其官方文档确认 Work Context 的配置要求。

4.29 AS-RAR-05007: Cannot specify both Execution Context [{0}] as well as Transaction Context [{1}] for Work [{2}] execution. Only one can be specified

原因：在提交 Work 任务（{2} 所指的任务）时，同时指定了 Execution Context（执行上下文，{0}）和 Transaction Context（事务上下文，{1}），而 JCA 规范要求两者只能选择其一，不能同时使用。

处理方案：从错误信息中获取 {2}（Work 任务名称），定位该任务的提交逻辑；检查代码或配置中是否同时设置了 `ExecutionContext` 和 `TransactionContext`；根据业务需求，移除其中一种上下文的配置，确保仅保留一种；若为第三方适配器，升级版本或参考其文档修正配置。通过遵循 JCA 规范的上下文使用规则，避免两种上下文同时关联，即可解决该冲突错误。

4.30 AS-RAR-05008 Duplicate Work Context for type [{0}]

原因：表示在提交 Work 任务时，检测到同一类型的 Work Context 被重复添加。Work Context 是 JCA 规范中用于传递特定上下文信息（如安全上下文、事务上下文、执行上下文等）的组件，每种类型的上下文在一个 Work 任务中只能存在一个，重复添加会违反规范导致错误。

处理方案：从错误信息中获取 {0}（重复的上下文类型），明确冲突的上下文类别；检查 Work 任务的代码或配置，定位重复添加该类型上下文的位置；移除重复的上下文实例，确保同一类型只保留一个；若为第三方适配器，升级版本或联系厂商获取支持。通过保证每种类型的 Work Context 在一个 Work 任务中唯一存在，即可符合 JCA 规范并解决该错误。

4.31 AS-RAR-05009 Application server cannot handle the following Work Context : {0}

原因：表示应用服务器无法处理资源适配器提交的 Work Context 类型 {0}。这通常是由于该 Work Context 不符合 JCA 规范、服务器不支持其类型，或适配器与服务器存在兼容性问题导致的。

处理方案：该错误的核心是 Work Context 类型不被服务器支持，需从规范兼容性、实现合法性和服务器状态三个维度排查解决。从错误信息中获取 {0}（不支持的 Work Context 类型），确认其是否为 JCA 标准类型；验证资源适配器的 JCA 版本与 AAS 版本的兼容性；检查 {0} 类型的实现是否符合规范，或是否存在服务器组件故障；通过升级、替换上下文类型或修复服务器环境解决兼容性问题。

4.32 AS-RAR-05010: Setting custom Work Context class [{0}] using most specific supported Work Context class [{1}]

原因： 通常是一条信息性日志（非错误），表示资源适配器提交的自定义 Work Context 类（`{0}`）已被服务器识别，并将通过最匹配的支持类型（`{1}`）进行处理。

处理方案： 是 AAS 处理自定义 Work Context 时的正常日志，用于说明服务器对自定义类的识别和关联逻辑，无需特殊处理；仅当功能异常时，需结合该信息排查接口实现是否正确。

4.33 AS-RAR-05011 Unable to set Security Context

原因： 表明资源适配器（JCA 适配器）在尝试为 Work 任务设置安全上下文（Security Context）时失败。安全上下文用于传递认证信息（如用户身份、权限等），供 Work 任务在执行时使用，其设置失败通常与安全配置错误、上下文信息无效或服务器安全组件异常有关。

处理方案： 该错误的核心是安全上下文无法被服务器验证或应用，需从认证信息、安全域配置、服务器安全组件三个维度逐步排查解决。检查安全上下文（`SecurityContext`）是否包含有效的 `Principal` 和凭证信息；验证资源适配器指定的安全域是否存在且配置正确，用户能否正常认证；查看日志排查服务器安全组件故障，修复权限或配置问题；确认 `SecurityContext` 实现符合 JCA 规范，适配器与服务器版本兼容；必要时调整安全策略，允许资源适配器设置安全上下文。

4.34 AS-RAR-05012 Unsupported callback {0} during credential mapping

原因： 在凭证映射（Credential Mapping）过程中，资源适配器使用了服务器不支持的回调类型（`{0}`）所指的具体回调类）。凭证映射是 JCA 规范中用于将应用服务器的安全凭证（如用户身份）转换为资源适配器所需凭证（如外部系统的登录信息）的过程，该过程依赖回调机制传递信息，若使用不支持的回调类型则会触发此错误。

处理方案： 该错误的核心是回调类型不被服务器支持，需通过规范兼容、配置修正或版本升级解决。从错误信息中获取 `{0}`（不支持的回调类型），确认其是否为 JAAS/JCA 标准回调；检查资源适配器的凭证映射配置和代码，定位使用该回调类型的位置；替换为服务器支持的标准回调类型，或修正自定义回调的实现使其符合规范；验证服务器安全组件是否正常，必要时升级适配器或服务器版本。

4.35 AS-CORE-JAVAE-0001 Web Container not installed

原因： 表明服务器中未安装或未启用 Web 容器（Web Container）。Web 容器是处理 Servlet、JSP、Web 应用的核心组件（如 AAS 的 `web` 模块），若缺失则无法部署或运行 Web 应用。

处理方案： 确认 AAS 安装了 Web 容器模块（检查 `modules` 目录下的核心 JAR）；启用 Web 容器模块（通过 `asadmin enable-module` 或修改 `domain.xml`）；查看 `server.log` 定位 Web 容器初始化失败的具体原因（如依赖、端口、冲突）；必要时重新安装服务器或重建域，恢复完整的 Web 容器配置。

4.36 AS-CORE-JAVAE-0002 Done with starting {0} container

原因：是 AAS服务器的一条信息性日志（非错误），表示指定的容器（{0} 所指的容器类型，如 Web 容器、EJB 容器、JPA 容器等）已成功启动完成。

处理方案：是AAS启动过程中确认容器启动完成的正常信息，用于监控服务器组件的初始化状态，无需额外操作。

4.37 AS-CORE-JAVAE-0003 Unable to start container (no exception provided)

原因：Web 容器无法正常启动。

处理方案：请检查web容器库是否正确。

4.38 AS-CORE-JAVAE-0004 Unable to start container {0}

原因：Web 容器无法正常启动。很可能存在类加载问题。

处理方案：从 `server.log` 中获取 {0}（容器名称）及关联的异常堆栈，明确失败的具体容器和根因；检查容器依赖的服务 / 模块是否正常启动，修复依赖问题；验证容器配置参数的合法性，修正错误配置；解决类缺失或冲突问题，确保核心类可正常加载；确认文件权限和文件系统状态，排除环境限制。

4.39 AS-DEPLOYMENT-04017 Error in deleting file {0}

原因：通常发生在部署、undeploy（卸载）或重新部署应用时，服务器尝试删除相关文件（如旧的应用包、临时文件或缓存文件）但失败的场景。

处理方案：解决文件被占用问题；修复文件权限问题；验证文件路径是否存在；检查文件系统状态。

4.40 AS-DEPLOYMENT-04018 Deployment manager load failure. Unable to find {0}

原因：{0} 通常指部署所需的关键资源路径，如部署描述符、应用文件或内部组件类，表示部署管理器在加载过程中找不到必要的资源，导致部署流程失败。

处理方案：验证部署文件的完整性和路径，确认错误信息中 {0} 对应的路径是否正确，文件是否真实存在；清理部署残留和临时文件；检查 AAS安装完整性；若以上步骤无效，通过AAS详细日志获取更具体的错误堆栈。

4.41 AS-DEPLOYMENT-04019 Deployment manager load failure. Unable to find {0}

原因： 部署管理器加载失败，但更侧重 部署过程中依赖的特定模块、类或配置文件缺失，错误信息格式为“Deployment manager load failure. Unable to find {0}”（{0} 通常指向具体的缺失资源，如模块 JAR、接口类或配置文件路径）。

处理方案： 通过日志定位具体缺失的资源（{0} 的值）；根据资源类型（模块、类、配置）针对性补充或修复；清理缓存和残留文件，确保部署环境干净。

4.42 AS-DEPLOYMENT-00001 Ignore {0} in archive {1}, as WLS counterpart runtime xml {2} is present in the same archive

原因： 应用包中同时包含 AAS 和 WebLogic 的专属部署配置文件，导致 AAS 部署管理器优先识别 WebLogic 的配置文件，并忽略了自身的对应配置。

处理方案： 是多服务器配置文件冲突的提示，而非致命错误，但可能导致 AAS 应用的配置未按预期生效。解决核心是：若仅部署到 AAS，删除 WebLogic 配置文件；若需跨服务器兼容，确保配置逻辑一致，以避免运行时异常。

4.43 AS-DEPLOYMENT-00002 Exception caught: {0}

原因： 该错误是“异常捕获”的通用提示，具体原因需结合 {0} 中的异常信息分析。

处理方案： 从日志中提取详细异常信息；根据异常类型（类缺失、XML 错误、资源问题等）针对性修复；必要时清理环境或简化测试，排除干扰因素。

4.44 AS-DEPLOYMENT-00003 {0} module [{1}] contains characteristics of other module type: {2}

原因： 表示部署的模块（如 WAR、JAR、EAR 等）中包含了其他类型模块的特征，导致 AAS 无法正确识别其类型，从而部署失败。

处理方案： 根据错误信息定位冲突的模块类型和特征；清理模块中不符合其类型规范的文件（如错误的描述符、目录结构）；修正构建配置，避免打包时混入无关资源。

4.45 AS-DEPLOYMENT-00004 Exception caught: {0} for the subarchive indicated by the path: {1}

原因： 主要发生在部署包含子归档（subarchive）的应用时（如 EAR 包中嵌套的 WAR、EJB JAR 等），表示子归档在处理过程中抛出了异常，导致整体部署失败。

处理方案： 通过日志确定具体的异常（{0}）和问题子归档（{1}）；单独分析子归档的完整性、配置和依赖，修复内部错误；确保子归档与父归档（如 EAR）的依赖不冲突，路径和权限合规。

4.46 AS-DEPLOYMENT-00005 Exception caught during annotation scanning

原因：表示在部署过程中，AAS的注解扫描器（负责解析应用中的 Java 注解，如 `@WebServlet`、`@EJB`、`@Entity` 等）遇到了异常，导致部署中断。

处理方案：通过日志定位具体的异常类型和涉及的类；修复类文件损坏、注解使用错误或依赖冲突；确保注解版本与 AAS 兼容，必要时清理缓存或修复服务器组件。

4.47 AS-DEPLOYMENT-00006 Adding {0} since {1} is annotated with {2}

****原因：**** 是一个信息性提示（非错误）；表示部署过程中，AAS 检测到某个类（`{1}`）使用了特定注解（`{2}`），因此自动添加了对应的组件或配置（`{0}`）以满足注解的运行时需求。

处理方案：是AAS部署时的信息提示，用于说明因注解触发的自动配置行为，目的是确保应用中的注解组件能正常运行。该信息本身无需处理，仅在关联组件出现错误时，需检查对应依赖是否完整。

4.48 AS-DEPLOYMENT-00007 Adding {0} since it is implementing {1}

原因：是一条信息性提示（非错误或警告）；表示部署过程中，AAS检测到某个类实现了特定的接口（`{1}`），因此自动添加了对应的组件、配置或支持类（`{0}`），以满足该接口的运行时规范要求。

处理方案：是 AAS 部署时的信息提示，表明服务器根据类所实现的标准接口自动添加了必要的支持组件。该信息本身无需处理，仅在关联组件运行异常时，需检查接口实现的完整性和兼容性。

4.49 AS-DEPLOYMENT-00008 Inconsistent type definition. {0} is neither an annotation nor an interface

原因：表示部署过程中，AAS 检测到某个类型（`{0}`）的定义不符合预期——该类型本应是注解（Annotation）或接口（Interface），但实际却是其他类型（如类 `class` 或枚举 `enum`），导致依赖该类型的组件无法正常解析，部署失败

处理方案：根据 `{0}` 定位问题类型；修正类型定义（改为注解或接口），或调整引用方式以符合规范；重新编译部署，确保类型使用符合 Java EE 规范要求。

4.50 AS-DEPLOYMENT-00009 The exception {0} occurred while examining the jar at file path: {1}

原因：表示在部署过程中，AAS 检查指定的 JAR 包（路径为 `{1}`）时发生了异常（`{0}` 为具体异常信息），导致部署中断。

处理方案：通过日志定位问题 JAR 包 ({1}) 和具体异常 ({0}) ；修复 JAR 包损坏、权限不足、依赖缺失等问题；确保 JAR 包格式正确、路径规范且与 AAS 兼容。

4.51 AS-DEPLOYMENT-00010 No classloader can be found to use

原因：表示部署过程中，AAS无法为应用或模块找到可用的类加载器 (classloader) ，导致类加载机制失效，部署失败。

处理方案：恢复类加载器的创建能力；确保 AAS 核心组件完整、启动正常；验证应用包结构完整且无依赖冲突；修复权限和资源问题，清理类加载器缓存。

4.52 AS-DEPLOYMENT-00011 Error in annotation processing: {0}

原因：表示在部署过程中，AAS的注解处理器（负责解析和处理应用中的 Java 注解，如 @Entity 、 @WebService 、 @Inject 等）在处理注解时发生了错误，导致部署中断。

处理方案：通过日志获取 {0} 中的具体错误信息，定位涉及的注解、类和处理器；修复注解使用错误、自定义处理器逻辑问题或依赖冲突；确保注解及处理器与 AAS 版本和规范兼容。

4.53 AS-DEPLOYMENT-00012 Cannot load {0} reason : {1}

****原因：** ** 表示部署过程中无法加载指定的资源 ({0}) ， {1} 则描述了具体原因，导致部署失败。

处理方案：根据 {0} 定位资源，结合 {1} 分析具体原因（缺失、权限、损坏等）；针对性修复资源缺失、权限问题或版本冲突；确保资源路径正确、格式合法且与 AAS 环境兼容。

4.54 AS-DEPLOYMENT-00013 An exception was caught during library jar processing: {0}

****原因：** ** 表示在部署过程中，AAS 处理应用的库 JAR 包（如 WEB-INF/lib 中的依赖 JAR 或 EAR 的 lib 目录中的共享 JAR）时发生了异常 ({0} 为具体异常信息) ，导致部署中断。

处理方案：通过日志定位具体异常类型和问题 JAR 包；修复 JAR 损坏、依赖冲突、权限不足等问题；确保 JAR 包格式正确、路径规范且与AAS环境兼容。

4.55 AS-DEPLOYMENT-00014 Could not load {0}

****原因：** ** 表示部署过程中无法加载指定的资源 ({0}) 通常是类、JAR 包、配置文件或其他部署所需的关键资源)，导致部署流程中断。该错误与 AS-DEPLOYMENT-00012 类似，但更侧重“加载动作本身失败”，具体原因需结合上下文日志分析。

处理方案： 确认 {0} 资源的存在性和路径正确性；验证资源的完整性、兼容性及访问权限；补充缺失的依赖，解决类加载冲突。

4.56 AS-DEPLOYMENT-00015 Unsupported deployment descriptors

element {0} value {1}

****原因：** ** 表示部署描述符（如 `web.xml`、`ejb-jar.xml`、`glassfish-web.xml` 等）中包含不被支持的元素（{0}）或元素值（{1}），导致 AAS 无法解析部署配置，部署失败。

处理方案： 确认 AAS 支持的 Java EE 规范版本；根据错误信息定位并修正不合法的元素或值；确保配置文件的版本声明和命名空间符合规范。

4.57 AS-DEPLOYMENT-00016 Unknown port-component-name {0} port, all sub elements will be ignored

原因： 主要出现在部署包含 Web Service（如 JAX-WS 服务）的应用时，表示部署描述符（通常是 `webservicex.xml` 或 `sun-jaxws.xml`）中定义的 `<port-component-name>` 为 {0} 的端口组件未被识别，因此该端口组件下的所有子元素（如绑定地址、服务实现类等配置）将被忽略，可能导致 Web Service 无法正常发布。

处理方案： 确保 `<port-component-name>` 的值与服务实现类的注解、WSDL 定义一致，或移除无效的冗余配置。

4.58 AS-DEPLOYMENT-00017 DOLUtils: converting EJB to web bundle id {0}

原因： 是一条信息性提示（非错误或警告）；表示部署过程中，AAS 的 DOL（Deployment Object Library，部署对象库）工具正在将一个 EJB 组件转换为 Web 捆绑包（web bundle），其中 {0} 是该转换过程中生成的唯一标识（ID）。

处理方案： 是 AAS 部署内嵌 EJB 的 Web 应用时的正常信息提示，表明服务器正在将 EJB 组件转换为 Web 兼容格式以支持协同运行。该信息无需处理，仅需关注应用部署后的功能是否正常即可。

4.59 AS-DEPLOYMENT-00020 Application validation failed for application: {0}, jndi-name: {1}, resource adapter name: {2} is wrong

原因： 表示应用在验证阶段失败，原因是指定的资源适配器名称（{2}）不正确，导致与 JNDI 名称（{1}）关联的资源无法正常绑定，最终部署失败。

处理方案： 确认 AAS 中已部署的资源适配器的准确名称；修正应用配置中资源适配器名称的拼写或匹配问题；确保资源适配器已正确部署且与 JNDI 名称绑定有效。

4.60 AS-DEPLOYMENT-00022 For administered-object resource: {0}, there is no application part in its resource adapter name: {1}.

原因： 表示管理对象资源（{0}）所关联的资源适配器名称（{1}）中缺少“应用部分”（application part），不符合 AAS 对资源适配器名称的格式要求，导致资源无法正确关联和部署。

处理方案： 确认资源适配器的完整名称（格式为 [应用名]/[适配器名]）；修正管理对象资源配置，确保 `resource-adapter-name` 包含应用部分；部署资源适配器时关联应用，使其名称自动包含应用上下文。

4.61 AS-DEPLOYMENT-00023 For connection-factory resource: {0}, there is no application part in its resource adapter name: {1}.

原因： 表示连接工厂资源（{0}）所关联的资源适配器名称（{1}）中缺少“应用部分”（application part），不符合 AAS 对资源适配器名称的格式要求，导致连接工厂资源无法正确绑定和使用，部署失败。

处理方案： 确认资源适配器的完整名称（格式为 [应用名]/[适配器名]）；修正连接工厂资源配置，确保 `resource-adapter-name` 包含应用部分；部署资源适配器时关联应用，使其名称自动包含应用上下文。

4.62 AS-DEPLOYMENT-00024 For jms-connection-factory resource: {0}, there is no application part in its resource adapter name: {1}

原因： 表示 JMS 连接工厂资源（{0}）所关联的资源适配器名称（{1}）中缺少“应用部分”（application part），不符合 AAS 对资源适配器名称的格式要求，导致 JMS 连接工厂无法正确绑定，部署失败。

处理方案： 确认 JMS 资源适配器的完整名称（格式为 [应用名]/[适配器名]）；修正 JMS 连接工厂配置，确保 `resource-adapter-name` 包含应用部分；部署资源适配器时关联应用，使其名称自动包含应用上下文。

4.63 AS-DEPLOYMENT-00025 For jms-destination resource: {0}, there is no application part in its resource adapter name: {1}

原因： 表示 JMS 目标资源（{0}，如 JMS 队列或主题）所关联的资源适配器名称（{1}）中缺少“应用部分”（application part），不符合 AAS 对资源适配器名称的格式要求，导致 JMS 目标资源无法正确绑定，部署失败。

处理方案： 确认 JMS 资源适配器的完整名称（格式为 [应用名]/[适配器名]）；修正 JMS 目标资源配置，确保 `resource-adapter-name` 包含应用部分；部署资源适配器时关联应用，使其名称自动包含应用上下文。

4.64 AS-DEPLOYMENT-00026 JNDI lookup failed for the resource: Name: {0}, Lookup: {1}, Type: {2}

原因：表示应用在尝试通过 JNDI (Java Naming and Directory Interface) 查找指定资源时失败。其中 {0} 是资源的名称 (通常是应用中声明的资源引用名) ; {1} 是实际执行 JNDI 查找时使用的名称 (如 `java:comp/env/jdbc/MyDataSource`) ; {2} 是预期的资源类型 (如 `javax.sql.DataSource` 、 `javax.jms.ConnectionFactory` 等) 。

处理方案：确认 AAS 中已配置目标资源, 且 JNDI 名称 ({1}) 正确; 修正应用中资源引用与全局 JNDI 的映射关系; 确保资源类型匹配且配置正确 (能正常初始化) 。

4.65 AS-DEPLOYMENT-00027 Resource Adapter not present: RA Name: {0}, Type: {1}

原因：表示应用依赖的资源适配器 (Resource Adapter, 简称 RA) 不存在或未部署。其中 {0} 是应用所需的资源适配器名称; {1} 是该资源适配器的类型 (通常是资源适配器的接口或实现类信息) 。

处理方案：确认名称为 {0} 的资源适配器已在 AAS 中部署; 确保资源适配器名称与应用配置一致, 类型符合 {1} 要求; 修复资源适配器部署失败的问题 (如文件损坏、依赖缺失) 。

4.66 AS-DEPLOYMENT-00028 Skipping resource validation

原因：是一条信息性提示 (非错误或警告) ; 表示部署过程中跳过了对应用所引用资源 (如数据源、JMS 连接工厂、资源适配器等) 的验证步骤。

处理方案：是 AAS 在特定部署场景下跳过资源验证步骤的信息提示, 属于正常的优化行为。该提示本身无需处理, 只需关注应用部署后的资源访问是否正常即可。

4.67 AS-DEPLOYMENT-02015 Skipped processing for module {0} as its module type was not recognized

原因：是一条信息性提示 (非错误, 但可能暗示部署问题) ; 表示部署过程中, AAS无法识别模块 {0} 的类型, 因此跳过了对该模块的处理。

处理方案：提示 AAS跳过了未识别类型的模块 {0} 。若该模块是应用必需的, 需通过修复模块结构、规范命名或显式声明模块类型使其被正确识别; 若为辅助文件, 则无需处理。确保必需模块符合 Java EE 规范即可消除该提示并正常部署。

4.68 AS-DEPLOYMENT-02016 Error occurred

原因：表示部署过程中发生了未被特定错误码归类的异常，导致部署中断。由于该错误本身未提供具体细节，需结合上下文日志分析具体原因。

处理方案：通过服务器日志获取具体异常信息（如堆栈跟踪）；根据异常类型定位问题（如配置错误、类冲突、资源不足等）；针对性修复应用包、配置或服务器环境。

4.69 AS-EJB-00001 [{0}]: trimLru(), resetting head and tail

原因：是一条与 EJB（Enterprise JavaBean）容器缓存管理相关的调试级信息；该信息主要用于记录 EJB 容器在执行 LRU（Least Recently Used，最近最少使用）缓存清理策略时的内部状态调整，通常不影响应用的正常运行。

处理方案：是 AAS EJB 容器在执行 LRU 缓存清理时的正常调试信息，用于记录缓存链表的状态调整，不影响应用功能。仅当频繁出现并伴随性能问题时，才需要调整有状态会话 Bean 的缓存配置。

4.70 AS-EJB-00002 [{0}]: Exception in backingStore.remove([1])

原因：表示 EJB 容器在通过后端存储（backing store）移除某个 EJB 实例（通常是有状态会话 Bean）时发生了异常。其中 {0} 是 EJB 缓存或后端存储的标识（如 `StatefulSessionBeanCache`）；{1} 是被移除的 EJB 实例的 ID（用于唯一标识该实例）。

处理方案：通过日志确定异常类型（文件 / 数据库访问、序列化问题等）；修复后端存储的访问问题（权限、磁盘空间、数据库连接）；确保有状态会话 Bean 可正常序列化，避免并发冲突。

4.71 AS-EJB-00003 [{0}]: passivateEJB(), Exception caught ->

原因：表示 EJB 容器在执行 `passivateEJB()` 方法（钝化 EJB 实例）时捕获到异常。其中 {0} 通常是 EJB 组件的标识（如有状态会话 Bean 的名称或缓存标识）。

处理方案：确保 SFSB 所有字段可序列化（或标记为 `transient`）；修复后端存储的访问问题（权限、磁盘空间、数据库连接）；处理 `@PrePassivate` 方法中的异常。

4.72 AS-EJB-00004 [{0}]: Cannot load from BACKUPSTORE FOR Key: [1]

原因：表示 EJB 容器在尝试从后端存储（BACKUPSTORE）加载指定键（{1}）对应的 EJB 实例（通常是有状态会话 Bean，SFSB）时失败。其中 {0} 是 EJB 缓存或后端存储的标识（如有状态会话 Bean 缓存名称）；{1} 是用于标识 EJB 实例的唯一键（通常是实例 ID）。

处理方案：根据日志定位存储访问问题（文件 / 数据库）并修复；清理无效的实例数据，让客户端重新创建实例；优化 SFSB 的生命周期管理，避免无效引用。

4.73 AS-EJB-00005 [{0}]: Exception while loading from backup session: [{1}]

原因：表示 EJB 容器在从备份会话存储 (backup session) 加载指定会话 ({1}) 的 EJB 实例 (通常是有状态会话 Bean, SFSB) 时发生异常。其中 {0} 是 EJB 容器组件或缓存的标识 (如有状态会话 Bean 缓存名称) ; {1} 是备份会话的唯一标识 (通常是会话 ID 或实例 ID) 。

处理方案：根据日志定位具体异常类型 (类问题、存储问题) ; 修复类路径、序列化兼容性或存储访问配置; 清理无效备份数据, 确保集群配置一致。

4.74 AS-EJB-00006 [{0}]: Error while loading from backup session: [{1}]

原因：表示 EJB 容器在从备份会话存储 (backup session store) 加载指定会话 ({1}) 的 EJB 实例 (主要是有状态会话 Bean, SFSB) 时发生错误。其中 {0} 通常是 EJB 容器组件或缓存的标识 (如有状态会话 Bean 缓存名称) ; {1} 是备份会话的唯一标识 (如会话 ID 或实例 ID) 。

处理方案：根据日志定位存储访问问题或代码异常; 修复存储配置、权限或服务可用性; 清理无效数据并确保激活过程无未处理异常。

4.75 AS-EJB-00007 [{0}]: Exception during backingStore.passivateSave([{1}])

原因：表示 EJB 容器在通过后端存储 (backingStore) 执行 `passivateSave` 操作 (即钝化并保存有状态会话 Bean 实例) 时发生异常。其中 {0} 是 EJB 缓存或后端存储的标识 (如有 `StatefulSessionBeanCache` 等) ; {1} 是被钝化的 EJB 实例的唯一 ID。

处理方案：确保 SFSB 所有字段可序列化 (或标记为 `transient`) ; 修复后端存储的访问问题 (权限、磁盘空间、数据库连接) ; 处理 `@PrePassivate` 方法中的异常。

4.76 AS-EJB-00008 [{0}]: Iterator(), resetting head.lPrev

原因：是一条与 EJB 容器内部缓存管理相关的调试级信息; 该信息主要用于记录 EJB 容器在遍历缓存中的 EJB 实例 (通常是有状态会话 Bean) 时, 对缓存链表的头节点 (`head`) 的前驱指针 (`lPrev`) 进行重置的内部操作, 属于容器正常的缓存维护行为。

处理方案：是 AAS EJB 容器在初始化缓存迭代器时, 重置双向链表头节点前驱指针的正常调试信息, 属于容器内部缓存管理的一部分。该信息可忽略, 无需干预。

4.77 AS-EJB-00009 [{0}]: Exiting TrimTimedoutBeans() because current cache state: [{1}]

原因： 是一条与 EJB 容器缓存管理相关的信息性提示；该信息表示 EJB 容器在执行 `TrimTimedoutBeans()` 方法（清理超时的 EJB 实例）时，因当前缓存状态（`{1}`）不符合清理条件而退出该方法。

处理方案： 是 AAS EJB 容器在清理超时实例时，因缓存状态无需清理而退出操作的正常信息。该信息可忽略，仅需关注缓存是否正常管理实例（如无内存泄漏）即可。

4.78 AS-EJB-00010 [{0}]: TrimTimedoutBeans(), resetting head.lPrev

原因： 是一条与 EJB 容器缓存管理相关的调试级信息；该信息记录了 EJB 容器在执行 `TrimTimedoutBeans()` 方法（清理超时 EJB 实例）时，对缓存链表的头节点（`head`）前驱指针（`lPrev`）进行重置的内部操作，属于容器正常的缓存维护行为。

处理方案： 是 AAS EJB 容器在清理超时的有状态会话 Bean 实例时，重置缓存链表头节点前驱指针的正常调试信息，属于容器内部缓存管理的一部分。该信息可忽略，无需干预。

4.79 AS-EJB-00011 [{0}]: Exiting TrimUnSortedTimedoutBeans() because current cache state: [{1}]

原因： 是一条与 EJB 容器缓存管理相关的信息性提示；该信息表示 EJB 容器在执行 `TrimUnSortedTimedoutBeans()` 方法（清理未排序缓存中的超时 EJB 实例）时，因当前缓存状态（`{1}`）不符合清理条件而退出该方法。

处理方案： 是 AAS EJB 容器在清理未排序缓存中的超时实例时，因缓存状态无需清理而退出操作的正常信息。该信息可忽略，仅需关注缓存是否正常释放资源（如无内存泄漏）即可。

4.80 AS-EJB-00012 [SFSBContainer] Exception while initializing SessionSynchronization methods

原因： 表示有状态会话 Bean（SFSB）容器在初始化实现了 `SessionSynchronization` 接口的 EJB 实例的相关方法时发生异常。

处理方案： 确保接口方法签名正确、访问权限为 `public`；处理方法内部的异常，避免初始化中断；检查 EJB 类定义是否符合容器要求（非 `final`、接口实现正确）。

4.81 AS-EJB-00013 [SFSBContainer] Exception while loading checkpoint info

原因：表示有状态会话 Bean (SFSB) 容器在加载“检查点信息” (checkpoint info) 时发生异常。检查点信息是 SFSB 容器用于持久化或恢复 Bean 实例状态的关键数据 (类似状态快照)，通常在服务器重启、故障恢复或实例激活时加载。

处理方案：根据日志定位具体异常类型 (类问题、存储问题)；修复类路径、序列化兼容性或存储访问配置；清理无效检查点数据，确保集群配置一致。

4.82 AS-EJB-00014 Exception creating ejb object : [{0}]

原因：表示容器在创建 EJB 对象 ({0} 通常是 EJB 的名称或类名) 时发生异常，导致 EJB 实例无法正常初始化。

处理方案：通过日志定位具体失败环节 (类、注入、初始化等)；修复类结构、依赖配置或初始化逻辑；确保 EJB 符合规范，容器环境正常。

4.83 AS-EJB-00015 Exception creating ejb local object [{0}]

原因：表示容器在创建 EJB 本地对象 ({0} 通常是 EJB 的名称或本地接口名) 时发生异常。本地 EJB 对象 (Local EJB) 用于同一应用内部的组件通信 (如 EJB 之间、Servlet 调用 EJB)，其创建失败会导致应用内组件交互中断。

处理方案：确保本地接口定义正确，EJB 实现类匹配接口方法签名；修复类路径、构造函数及权限问题；处理依赖注入和初始化过程中的异常。

4.84 AS-EJB-00016 Couldn't update timestamp for: [{0}]; Exception: [{1}]

原因：表示 EJB 容器在尝试更新某个 EJB 实例 ({0} 通常是实例 ID 或标识) 的时间戳时失败， {1} 为具体的异常信息。

处理方案：根据日志定位存储访问问题 (文件 / 数据库) 并修复；清理无效存储数据，确保实例记录完整；优化缓存配置，减少并发冲突。

4.85 AS-EJB-00017 Cannot register bean for checkpointing

原因：表示 EJB 容器无法为某个 EJB 实例 (通常是有状态会话 Bean, SFSB) 注册“检查点” (checkpointing) 机制。检查点机制用于定期持久化 EJB 实例的状态 (如状态快照)，以支持故障恢复或钝化 / 激活流程，注册失败会导致实例状态无法被持久化，可能引发会话丢失或恢复失败。

处理方案： 确保 SFSB 及字段可序列化（或标记为 `transient`）；修复检查点存储的配置和访问权限；保证实例正常初始化，生命周期管理正确。

4.86 AS-EJB-00018 Error during checkpoint ([{0}]. Key: [{1}]) [{2}]

原因： 表示 EJB 容器在对标识为 {0} 的 EJB 组件（通常是有状态会话 Bean，SFSB）执行检查点（checkpoint）操作时发生错误。其中 {0} 是 EJB 组件的名称或标识； {1} 是该 EJB 实例的唯一键（如实例 ID）； {2} 是具体的异常信息。

处理方案： 确保 SFSB 及字段可序列化（或标记为 `transient`）；修复后端存储的访问和写入问题（权限、磁盘空间、数据库连接）；清理损坏数据，确保实例状态有效。

4.87 AS-EJB-00019 sfsb checkpoint error. Name: [{0}]

原因： 表示有状态会话 Bean（SFSB）在执行检查点（checkpoint）操作时发生错误，其中 {0} 是该 SFSB 的名称或标识。检查点操作是 SFSB 容器将实例状态持久化到后端存储（如文件系统、数据库）的关键过程，用于支持故障恢复、钝化 / 激活或集群状态同步，操作失败可能导致会话状态丢失或恢复异常。

处理方案： 确保 SFSB 及字段可序列化（或标记为 `transient`）；修复存储配置和访问权限，确保状态可正常写入；清理损坏数据，规范实例生命周期管理。

4.88 AS-EJB-00020 sfsb checkpoint error. Key: [{0}]

原因： 表示有状态会话 Bean（SFSB）在执行检查点（checkpoint）操作时发生错误，其中 {0} 是该 SFSB 实例的唯一标识（如实例 ID 或键值）。检查点操作的核心是将 SFSB 实例的状态（字段数据、事务上下文等）持久化到后端存储（文件系统或数据库），用于故障恢复、钝化 / 激活或集群同步，操作失败可能导致实例状态丢失或无法恢复。

处理方案： 根据日志定位具体实例的错误原因（序列化 / 存储 / 状态）；修复序列化问题、存储访问权限或实例生命周期管理；清理该实例的损坏数据，确保检查点操作可正常执行。

4.89 AS-EJB-00021 Exception in afterCompletion : [{0}]

原因： 表示有状态会话 Bean（SFSB）在执行 `afterCompletion` 回调方法时发生异常，其中 {0} 通常是具体的异常信息或 EJB 标识。

处理方案： 在 `afterCompletion` 中添加全面的异常捕获和处理；检查并确保依赖资源的有效性；简化方法逻辑，避免高风险操作。

4.90 AS-EJB-00022 1. passivateEJB() returning because containerState: [{0}]

原因： 是一条与有状态会话 Bean (SFSB) 生命周期管理相关的信息性提示；该信息表示 EJB 容器在尝试执行 `passivateEJB()` 方法（钝化 EJB 实例）时，因当前容器状态（`{0}`）不符合钝化条件而退出该方法。

处理方案： 是 AAS EJB 容器在尝试钝化 SFSB 实例时，因容器状态不允许而退出操作的正常信息。该信息可忽略，仅需关注容器是否正常管理实例生命周期（如无内存泄漏）即可。

4.91 AS-EJB-00023 Extended EM not serializable. Exception: [{0}]

原因： 表示扩展持久化上下文 (Extended EntityManager, 简称“Extended EM”) 不可序列化，在需要序列化包含该上下文的有状态会话 Bean (SFSB) 时抛出异常，其中 `{0}` 是具体的序列化相关异常（如 `NotSerializableException`）。

处理方案： 用 `transient` 修饰 `EntityManager` 字段，避免其被序列化；在 `@PostActivate` 中利用容器自动重新注入的特性，确保激活后 EM 可用；钝化前清理实体依赖，避免间接序列化问题。

4.92 AS-EJB-00024 Error during passivation: [{0}]; [{1}]

原因： 表示 EJB 容器在对标识为 `{0}` 的 EJB 实例（主要是有状态会话 Bean, SFSB）执行钝化 (passivation) 操作时发生错误，`{1}` 为具体的异常信息。钝化是 SFSB 生命周期中的关键过程，指容器将长期未使用的实例状态序列化并存储到后端存储（如文件系统、数据库）以释放内存，操作失败可能导致内存占用过高或实例状态丢失。

处理方案： 确保实例及字段可序列化（或标记为 `transient`）；处理 `@PrePassivate` 中的异常，避免中断流程；修复存储配置和访问权限，清理损坏数据。

4.93 AS-EJB-00025 Error during passivation of [{0}]

原因： 表示 EJB 容器在对标识为 `{0}` 的 EJB 实例（主要是有状态会话 Bean, SFSB）执行钝化 (passivation) 操作时发生错误。钝化是 SFSB 生命周期中释放内存的关键过程，容器会将长期未使用的实例状态序列化并存储到后端存储（如文件系统、数据库），操作失败可能导致内存占用过高、实例状态丢失或后续激活失败。

处理方案： 确保实例及字段可序列化（或标记为 `transient`）；处理 `@PrePassivate` 中的异常，避免中断流程；修复存储配置和访问权限，清理损坏数据。

4.94 AS-EJB-00026 sfsb passivation error. Key: [{0}]

原因： 表示有状态会话 Bean (SFSB) 在执行钝化 (passivation) 操作时发生错误，其中 `{0}` 是该 SFSB 实例的唯一标识（如实例 ID 或键值）。钝化是 SFSB 生命周期中释放内存的关键过程——容器将长期未使用的实例状态序列化并存储到后端存储（文件系统或数据库），操作失败可能导致内存占用过高、实例状态丢失或后续激活失败。

处理方案： 根据日志定位具体实例的错误原因（序列化 / 预处理 / 存储）；修复序列化问题、处理预处理异常或存储访问权限；清理该实例的损坏数据，确保钝化操作可正常执行。

4.95 AS-EJB-00028 Error during Stateful Session Bean activation for key [{0}]

原因：表示容器在激活（activation）标识为 {0} 的有状态会话 Bean（SFSB）实例时发生错误。激活是 SFSB 生命周期中的关键过程——容器将之前钝化（passivation）到后端存储（如文件系统、数据库）的实例状态重新加载到内存并恢复使用，操作失败会导致客户端无法访问该实例，可能引发会话丢失或功能异常。

处理方案：根据日志定位具体实例的错误原因（反序列化 / 存储 / 初始化）；修复类路径、序列化兼容性或存储访问问题；清理损坏数据，处理激活回调中的异常。

4.96 AS-EJB-00029 [{0}]: Error during backingStore.shutdown()

原因：表示 EJB 容器的后端存储（backing store）在执行关闭（shutdown）操作时发生错误，其中 {0} 通常是后端存储的标识（如“file”“jdbc”或具体存储名称）。后端存储用于持久化有状态会话 Bean（SFSB）的钝化状态或检查点数据，其关闭操作失败可能导致资源未释放、数据不一致或服务器关闭异常。

处理方案：根据日志定位存储类型及具体异常（文件 / 数据库相关）；释放锁定资源、修复权限或连接池配置；手动清理残留数据，调整关闭顺序确保资源正确释放。

4.97 AS-EJB-00030 [{0}]: Error during onShutdown()

原因：表示 EJB 容器的某个组件（通常是与有状态会话 Bean 管理相关的服务，如缓存管理器、检查点服务等）在执行 `onShutdown()` 回调方法时发生异常，其中 {0} 是该组件的标识（如“sfsb-cache”“checkpoint-service”等）。`onShutdown()` 方法用于在服务器关闭或容器销毁时执行组件的清理逻辑（如释放资源、持久化状态），操作失败可能导致资源泄漏或状态数据不一致。

处理方案：根据日志定位具体组件和异常原因（资源 / 依赖 / 状态）；释放锁定资源、修复配置或调整关闭顺序；手动清理残留数据，确保组件正常关闭。

4.98 AS-EJB-00031 [{0}]: Error while undeploying ctx. Key: [{1}]

原因：表示在卸载（undeploy）应用上下文（context）时，EJB 容器处理标识为 {1} 的资源（通常是 EJB 实例、缓存项或绑定到 JNDI 的对象）时发生错误，其中 {0} 是相关的 EJB 组件或容器服务标识。卸载操作失败可能导致资源残留（如未释放的 EJB 实例、JNDI 绑定未移除），影响后续部署或服务器稳定性。

处理方案：根据日志定位具体资源（`Key: [{1}]`）和异常原因；释放资源引用、终止事务或重启依赖服务；手动卸载并清理残留文件，确保资源彻底释放。

4.99 AS-EJB-00032 Cannot add idle bean cleanup task

原因：表示 EJB 容器在尝试添加“空闲 Bean 清理任务”时失败。该任务用于定期清理长期未使用的有状态会话 Bean (SFSB) 实例（因超过“空闲超时时间”），以释放内存资源，任务添加失败可能导致空闲实例堆积，引发内存泄漏或服务器性能下降。

处理方案：确保调度线程池资源充足，服务正常运行；修复 SFSB 缓存配置（如超时时间、存储配置）；清理容器临时数据，必要时升级服务器版本；通过以上措施，可使清理任务成功添加并执行，避免空闲实例堆积导致的资源问题。

4.100 AS-EJB-00033 Got exception during removeExpiredSessions (but the reaper thread is still alive)

原因：表示 EJB 容器的“过期会话清理线程 (reaper thread)”在执行 `removeExpiredSessions` 方法（清理过期的有状态会话 Bean 实例）时发生异常，但该清理线程本身仍处于活跃状态。

处理方案：根据日志定位异常类型（存储 / 实例 / 逻辑 Bug）；修复存储访问问题、清理无效实例或升级服务器；调整缓存配置，避免过期实例堆积。

4.101 AS-EJB-00034 Error during checkpoint(, but session not destroyed)

****原因：**** 表示有状态会话 Bean (SFSB) 在执行检查点 (checkpoint) 操作时发生错误，但对应的会话 (SFSB 实例) 并未被销毁。检查点操作用于将 SFSB 实例的状态持久化到后端存储（如文件系统、数据库）以支持故障恢复，错误发生后会话仍保持活跃状态，但状态可能未被正确保存，存在后续恢复失败的风险。

处理方案：根据日志定位具体异常，修复序列化或存储配置问题；优化实例状态管理，避免并发修改或事务冲突；确保存储长期稳定，必要时手动触发检查点重试。

4.102 AS-EJB-00035 Error during checkpoint

原因：表示有状态会话 Bean (SFSB) 在执行检查点 (checkpoint) 操作时发生错误。检查点是 EJB 容器将 SFSB 实例的状态（如字段值、事务上下文）序列化并持久化到后端存储（文件系统或数据库）的关键过程，用于支持故障恢复、集群状态同步或实例状态持久化。操作失败可能导致状态无法保存，进而引发服务器重启后会话丢失、恢复异常等问题。

处理方案：确保实例及字段可序列化（或标记为 `transient`）；修复存储配置和访问权限，确保状态可正常写入；清理损坏数据，规范实例生命周期管理。

4.103 AS-EJB-00036 TopLevel AvailabilityService.getAvailabilityEnabled: [{0}]

原因：是一条与高可用性 (High Availability, HA) 配置相关的信息性日志，该日志用于记录顶层“可用性服务 (AvailabilityService)”的启用状态，其中 `{0}` 是布尔值 (`true` 或 `false`)，表示当前是否启用了 EJB 高可用

性功能。

处理方案：是 AAS 记录高可用性服务启用状态的信息日志，`{0}` 的值（`true / false`）仅用于提示当前配置，无错误含义。若需在集群环境中实现 SFSB 故障转移，确保 `{0}` 为 `true` 并正确配置状态复制方式（如数据库、文件系统）即可；单节点环境中保持默认 `false` 即可。

4.104 AS-EJB-00037 TopLevel EjbAvailabilityService.getAvailabilityEnabled: `{0}`

原因：是一条与 EJB 高可用性（High Availability, HA）配置相关的信息性日志；该日志用于记录“EJB 可用性服务（EjbAvailabilityService）”的启用状态，其中 `{0}` 为布尔值（`true` 或 `false`），表示当前是否针对 EJB 组件（尤其是有状态会话 Bean SFSB）启用了高可用性功能。

处理方案：是 AAS 记录 EJB 可用性服务启用状态的信息日志，`{0}` 的值（`true / false`）仅用于提示当前 EJB 高可用性配置。若在集群环境中需要 SFSB 支持故障转移，确保 `{0}` 为 `true` 并正确配置状态复制存储（如数据库、集群文件系统）；单节点环境或无需高可用时，保持 `false` 即可。

4.105 AS-EJB-00038 Global AvailabilityEnabled: `{0}`, application AvailabilityEnabled: `{1}`

原因：是一条与高可用性（High Availability, HA）配置相关的信息性日志，该日志用于同时记录全局高可用性状态和当前应用的高可用性状态，其中 `{0}` 是布尔值（`true / false`），表示全局“可用性服务（AvailabilityService）”是否启用；`{1}` 是布尔值（`true / false`），表示当前部署的应用是否单独启用了高可用性。

处理方案：用于展示全局与应用级高可用性的状态组合，核心作用是明确应用的高可用性功能是否实际生效（需 `{0}=true` 且 `{1}=true`）。管理员可通过该日志快速定位配置不一致问题，确保高可用性功能按预期运行（如集群环境的 SFSB 故障转移）或在单节点环境中合理禁用以减少资源消耗。

4.106 AS-EJB-00039 Exception while trying to determine availability-enabled settings for this app

原因：表示容器在尝试获取当前应用的高可用性（High Availability, HA）启用状态（即 `availability-enabled` 配置）时发生异常。该错误通常与应用部署、配置读取或高可用性服务初始化相关，可能导致应用的高可用性功能无法正常生效（如状态复制、故障转移失效）。

处理方案：根据日志定位异常类型（配置解析 / 服务 / 文件访问）；修复部署描述符、确保高可用性服务正常运行；清理缓存并重新部署应用，恢复配置读取能力。

4.107 AS-EJB-00040 StatefulContainerBuilder AvailabilityEnabled [{0}] for this application

原因： 是一条与有状态会话 Bean (SFSB) 高可用性配置相关的信息性日志，该日志用于记录“有状态容器构建器 (StatefulContainerBuilder)”为当前应用设置的高可用性启用状态，其中 {0} 是布尔值 (true 或 false)，表示当前应用的 SFSB 是否启用了高可用性功能 (如状态复制、故障转移)。

处理方案： 用于记录当前应用 SFSB 容器的高可用性启用状态 ({0}=true / false)，是配置生效的直接反馈。在集群环境中，若需 SFSB 支持故障转移，需确保 {0}=true 并正确配置状态复制存储 (如数据库、集群文件系统)；单节点环境中保持 false 即可减少资源消耗。管理员可通过该日志快速确认高可用性配置是否符合预期。

4.108 AS-EJB-00041 StatefulContainerBuilder.buildStoreManager() storeName: [{0}]

原因： 是一条与有状态会话 Bean (SFSB) 存储管理相关的信息性日志，该日志用于记录“有状态容器构建器 (StatefulContainerBuilder)”在创建 SFSB 存储管理器 (StoreManager) 时使用的存储名称 (storeName)，其中 {0} 是具体的存储标识 (如文件路径、数据库连接池名称等)。

处理方案： 用于记录 SFSB 容器存储管理器使用的存储名称 (storeName)，是存储配置生效的直接反馈。管理员可通过该日志确认 SFSB 状态数据的存储位置，辅助排查存储相关问题 (如权限不足、连接失败)，确保 SFSB 的钝化、检查点及高可用性功能正常运行。

4.109 AS-EJB-00042 Could not instantiate backing store for type [{0}]

原因： 表示 EJB 容器无法为指定类型 ({0})，如 file 或 jdbc) 的后端存储 (backing store) 创建实例。后端存储用于持久化有状态会话 Bean (SFSB) 的状态 (如钝化、检查点数据)，实例化失败会导致 SFSB 无法正常进行状态管理，可能引发钝化失败、状态丢失或应用功能异常。

处理方案： 根据 {0} 确定存储类型，针对性检查目录权限或数据库配置；确存储资源 (目录 / 连接池 / 表) 可用且配置正确；必要时重置为默认存储类型，恢复 SFSB 状态管理功能。

4.110 AS-EJB-00043 StatefulContainerbuilder instantiated store: {0}, with ha-enabled [{1}], and backing store configuration: {2}

原因： 是一条与有状态会话 Bean (SFSB) 存储管理和高可用性配置相关的信息性日志，该日志用于详细记录 SFSB 容器构建器 (StatefulContainerBuilder) 成功实例化后端存储后的关键信息，帮助管理员确认存储和高可用性配置的实际生效状态。

处理方案：是 SFSB 后端存储初始化成功的确认信息，通过 {0}（存储类型）、{1}（高可用状态）、{2}（配置详情）三个字段，清晰展示存储的技术实现、高可用性联动状态及具体配置参数。管理员可通过该日志快速验证存储和高可用性配置的正确性，确保 SFSB 状态管理（钝化、激活、故障转移）按预期运行。

4.111 AS-EJB-00044 Error while adding idle bean passivator task

原因：表示 EJB 容器在尝试添加“空闲 Bean 钝化任务”时失败。该任务用于定期将长期未使用的有状态会话 Bean（SFSB）实例钝化（passivate）到后端存储（释放内存），任务添加失败可能导致空闲实例长期占用内存，引发内存泄漏或服务器性能下降。

处理方案：确保调度线程池资源充足，服务正常运行；修复 SFSB 缓存配置（如超时时间、存储配置）；清理容器临时数据，必要时升级服务器版本。

4.112 AS-EJB-00045 Error while adding idle bean removal task

原因：表示 EJB 容器在尝试添加“空闲 Bean 移除任务”时失败。该任务用于定期清理长期未使用且超过最大存活时间的有状态会话 Bean（SFSB）实例（彻底销毁而非钝化），任务添加失败可能导致无效实例堆积，占用内存和后端存储资源，影响服务器性能。

处理方案：确保调度线程池和服务正常运行；修复 Max Lifetime 等清理策略配置；手动清理堆积实例，必要时升级服务器，通过以上措施，可使移除任务成功添加并执行，及时销毁无效 SFSB 实例，避免资源泄漏。

4.113 AS-EJB-00046 Error while removing idle beans for [{0}]

原因：表示 EJB 容器在执行“空闲 Bean 移除”操作时发生错误，其中 {0} 通常是有状态会话 Bean（SFSB）的容器标识或应用名称。该操作用于清理超过最大生命周期（Max Lifetime）或长期空闲且无法钝化的 SFSB 实例，错误发生可能导致无效实例残留，占用内存和存储资源。

处理方案：根据日志定位异常类型，终止无效事务或修复存储问题；处理并发冲突，确保移除操作时实例状态稳定；清理缓存并验证配置，避免实例堆积。

4.114 AS-EJB-00047 Error while removing expired beans for [{0}]

原因：表示 EJB 容器在清理“过期的有状态会话 Bean（SFSB）实例”时发生错误，其中 {0} 通常是 SFSB 容器标识或所属应用名称。“过期”指实例超过配置的生存周期（如 Max Lifetime）或空闲时间（Cache Idle Timeout）且无法通过钝化保留，容器需强制销毁以释放资源，错误发生可能导致过期实例残留，占用内存和存储资源。

处理方案：根据日志定位异常类型，终止无效事务或修复存储问题；处理并发冲突，确保清理时实例状态稳定；清理缓存并验证过期策略配置，避免实例堆积。

4.114.1 AS-EJB-00048 Passivation-capable value of stateful session bean [{0}] is false, it should not have any PrePassivate nor PostActivate configuration, but you have configuration at [{1}]

原因：表示有状态会话 Bean (SFSB) {0} 的“可钝化性 (passivation-capable)”被设置为 `false` (即不支持钝化)，但该 Bean 却配置了 `@PrePassivate` 或 `@PostActivate` 注解 (或对应的 XML 部署描述符配置)，而 {1} 指出了具体的配置位置 (如类或方法名)。

处理方案：根据错误信息定位冲突的 SFSB 和配置位置；要么删除回调配置 (保留 `passivation-capable=false`)，要么启用钝化 (保留回调配置)；确保继承或拦截器中无残留的冲突配置。

4.115 AS-EJB-00049 Cannot find stateful session bean [{0}] in memory, and will not read it from disk because current stateful session bean passivation-capable value is false

原因：表示容器在内存中找不到有状态会话 Bean (SFSB) {0} 的实例，且由于该 SFSB 的“可钝化性 (passivation-capable)”被设置为 `false` (不支持钝化)，容器也不会从磁盘 (后端存储) 中读取其实例状态，最终导致实例无法恢复。

处理方案：启用钝化 (`passivation-capable=true`)，允许实例持久化到磁盘；调整缓存配置，减少不必要的实例清理；优化客户端逻辑，避免依赖易丢失的内存实例。

4.116 AS-EJB-00050 Cache is shutting down, {0} stateful session beans will not be restored after restarting since passivation is disabled

原因：是一条警告信息，表示 SFSB 缓存正在关闭，且由于 {0} 个有状态会话 Bean (SFSB) 的“可钝化性 (passivation-capable)”被设置为 `false` (禁用钝化)，这些实例在服务器重启后将无法恢复。

处理方案：警告提示 缓存关闭时，禁用钝化的 SFSB 实例将无法在重启后恢复。若业务允许状态丢失，可忽略此警告；若需保留状态，需启用钝化 (`passivation-capable=true`)、确保实例可序列化，并配置有效的后端存储。通过这些措施，可使 SFSB 实例在服务器重启后正常恢复，避免状态丢失。

4.117 AS-EJB-00051 Disabling high availability for the stateful session bean {0}, as its marked non passivable

原因：是一条信息性日志 (或警告)；表示由于有状态会话 Bean (SFSB) {0} 被标记为“不可钝化 (non-passivable)”，容器已为其禁用高可用性功能。

处理方案： 日志提示不可钝化的 SFSB 被自动禁用高可用性，因高可用性依赖钝化机制实现状态管理。若业务无需高可用性，可忽略此日志；若需启用，需将 SFSB 配置为可钝化（`passivation-capable=true`）并确保其可序列化，同时验证高可用性全局配置。通过这些措施，可使 SFSB 在集群环境中支持状态复制和故障转移。

4.118 AS-EJB-00052 Internal Error

原因： 表示 EJB 容器在处理过程中发生了未预期的内部错误。这类错误通常由容器内部逻辑缺陷（Bug）、资源竞争异常或状态不一致导致，而非应用代码直接引发，可能影响 EJB 组件的正常运行（如部署失败、方法调用异常、事务处理错误等）。

解决方案： 通过日志定位错误发生的具体场景和异常类型；采取重启、简化配置、升级版本等措施规避或修复；若涉及已知 Bug，参考官方解决方案或升级容器。

4.119 AS-EJB-00053 The feature {0} requires Full Java EE Profile to be supported

原因： 表示当前使用的 EJB 功能 {0} 需要“完整 Java EE 规范（Full Java EE Profile）”支持，但服务器可能运行在“Web 概要文件（Web Profile）”或其他简化版本中，导致该功能无法启用。

处理方案： 确认服务器版本是否为 Full Profile，若不是则升级；若无法升级，修改应用以移除依赖 Full Profile 的功能（如远程 EJB、高级 MDB 特性等）。

4.120 AS-EJB-00054 Portable JNDI names for EJB {0}: {1}

原因： 是一条信息性日志，用于记录 EJB 组件 {0}（通常是 EJB 名称或类名）的可移植 JNDI 名称（Portable JNDI Names），其中 {1} 列出了该 EJB 对应的一个或多个标准 JNDI 名称。

处理方案： 用于记录 EJB 的可移植 JNDI 名称，这些名称遵循 Java EE 规范，确保 EJB 查找的跨服务器兼容性。开发人员可通过日志明确 EJB 的标准命名，方便客户端代码编写和问题排查，是 EJB 部署和使用中的重要参考信息。

4.121 AS-EJB-00055 Apusic-specific (Non-portable) JNDI names for EJB {0}: {1}

原因： 是一条信息性日志，用于记录 EJB 组件 {0}（通常是 EJB 名称或类名）的 AAS 特定 JNDI 名称（非可移植名称），其中 {1} 列出了该 EJB 在 AAS 中特有的 JNDI 绑定名称。

处理方案： 记录了 EJB 在 AAS 中的特有 JNDI 名称，这些名称为 AAS 自定义，仅在该服务器中有效，可简化本地部署的代码，但缺乏跨服务器可移植性。开发人员需根据应用的部署范围（仅 AAS 或需跨服务器）选择使用特定名称或可移植名称，避免迁移时出现问题。

4.122 AS-EJB-00056 A system exception occurred during an invocation on EJB {0}, method: {1}

原因：表示在调用 EJB {0} 的方法 {1} 时发生了系统级异常（System Exception）。这类异常通常由 EJB 容器或底层系统错误引发，而非应用代码中的业务异常，可能导致 EJB 方法调用失败、事务回滚或会话失效。

处理方案：通过日志堆栈定位具体异常类型和触发点；针对性修复资源问题、代码缺陷或配置错误；加强异常处理，避免未捕获的运行时异常扩散。

4.123 AS-EJB-00057 Error while creating enterprise bean context for {0} during jacc callback

原因：表示在为 EJB {0} 创建企业 Bean 上下文（Enterprise Bean Context）的过程中，执行 JACC（Java Authorization Contract for Containers）回调时发生错误。该错误与 EJB 的安全上下文初始化相关，可能导致 EJB 无法正常部署或访问被拒绝。

处理方案：根据日志定位具体的安全配置问题（策略文件、注解、领域）；修复配置冲突或无效项，确保 JACC 框架能正常处理权限回调；验证安全服务和领域的可用性，恢复 EJB 上下文的正常初始化。

4.124 AS-EJB-00058 Attempt to override reserved ejb interface method [{0}] in [{1}]. Override will be ignored

原因：是一条警告信息。表示 EJB 实现类 {1} 尝试重写（override）某个 EJB 规范中“预留的接口方法” {0}，但容器会忽略该重写操作。

处理方案：避免显式实现 `SessionBean` 等接口，优先使用注解定义 EJB；移除对预留方法的重写，通过 `@Resource` 等注解替代手动处理上下文。

4.125 AS-EJB-00059 Bean class for ejb [{0}] does not define a method corresponding to [{1}] interface method [{2}]

原因：表示 EJB {0} 的实现类中，没有定义与接口 {1} 中的方法 {2} 对应的实现方法，导致接口方法无法被正确映射，违反了 EJB 规范中“接口与实现类方法匹配”的要求。

处理方案：根据错误信息定位不匹配的接口方法；在实现类中补充缺失的方法，确保方法签名（名称、参数、返回值）与接口完全一致；同步接口与实现类的修改，避免因更新不同步导致的不匹配。

4.126 AS-EJB-00060 keepstate is true and will not create new auto timers during deployment

原因： 是一条信息性日志，表示在部署 EJB 时，由于配置了 `keepstate=true`（保留状态），容器不会创建新的自动定时器（Auto Timers），而是会尝试恢复之前持久化的定时器状态。

处理方案： 提示 由于 `keepstate=true`，部署时不会创建新的自动定时器，而是复用已有的持久化状态。这是符合预期的部署行为，用于确保定时器状态的连续性。若需重置定时器，可将 `keepstate` 改为 `false`；否则无需处理，日志仅为信息告知。

4.127 AS-EJB-00061 Failed to initialize the interceptor

原因： 表示 EJB 容器在初始化拦截器（Interceptor）初始化过程中发生失败。拦截器用于在 EJB 方法调用前后插入自定义逻辑（如日志、事务管理、安全校验等），初始化失败会导致依赖该拦截器的 EJB 无法正常部署或运行。

处理方案： 根据日志堆栈定位拦截器及具体异常类型；修复拦截器代码中的逻辑错误、配置错误或依赖问题；确保拦截器类符合 EJB 规范（非抽象、有无参构造函数等）。

4.128 *AS-EJB-00062 [*BaseContainer*] Could not create MonitorRegistryMediator. [{0}]*

原因： 表示 EJB 核心容器（`BaseContainer`）在创建 `MonitorRegistryMediator` 组件时失败，其中 `{0}` 通常包含具体的异常信息（如 `NullPointerException`、`ClassNotFoundException` 等）。

`MonitorRegistryMediator` 是 EJB 容器中负责监控数据注册与管理的组件，用于收集和暴露 EJB 的运行监控指标（如方法调用次数、事务统计等），其创建失败可能导致 EJB 监控功能不可用，但通常不影响 EJB 核心功能的基本运行。

处理方案： 根据日志异常类型修复监控组件依赖或配置；启用监控服务（若需要监控功能）或禁用监控（临时规避）；升级服务器版本以修复已知 Bug。

4.129 AS-EJB-00063 Exception in backingStore.size()

原因： 表示 EJB 容器在调用后端存储（backing store）的 `size()` 方法时发生异常。后端存储是用于持久化有状态会话 Bean（SFSB）状态的存储机制（如文件系统、数据库等），`size()` 方法用于获取存储中钝化的 SFSB 实例数量，异常通常意味着存储访问失败或存储本身存在问题。

处理方案： 根据日志定位存储类型和具体异常（文件 / 数据库问题）；修复存储访问权限、连接或数据损坏问题；必要时重启服务器或升级版本，解决并发冲突或组件 Bug。

4.130 AS-JMS-CORE-00001 JMS Service Connection URL is : {0}

原因： 是一条信息性日志，用于记录AAS 内置 JMS（Java Message Service）服务的连接 URL，其中 {0} 是具体的 URL 地址，客户端或应用可通过该 URL 连接到 JMS 服务。

处理方案： 用于记录AAS 内置 JMS 服务的连接 URL，是客户端连接 JMS 服务的关键参考信息。开发人员可通过该 URL 配置客户端连接参数，验证服务可用性，或排查连接相关问题。若需调整连接地址或端口，可通过管控台或命令修改 JMS 服务配置。

4.131 AS-JMS-CORE-00002 ADDRESSLIST in setJmsServiceProvider : {0}

原因： 是一条信息性日志，用于记录在设置 JMS 服务提供程序（JMS Service Provider）时使用的地址列表（ADDRESSLIST），其中 {0} 是具体的地址列表内容。该地址列表用于指定 JMS 客户端或服务端连接到 JMS 服务（通常是 OpenMQ 消息代理）的目标地址集合。

处理方案： 用于记录 JMS 服务提供程序初始化时使用的地址列表（ADDRESSLIST），反映 JMS 服务的可连接节点信息。该日志帮助开发人员确认 JMS 服务的部署拓扑、配置客户端连接参数，以及排查连接相关问题。若需调整地址列表，可通过管控台或命令管理 JMS 主机配置。

4.132 AS-JMS-CORE-00003 Addresslist : {0}

原因： 是一条信息性日志，用于记录 JMS（Java Message Service）服务当前使用的地址列表（Address List），其中 {0} 是具体的地址列表内容。该地址列表包含了 JMS 客户端或服务端可连接的 JMS 消息代理（通常是 OpenMQ）的所有目标地址，是 JMS 通信的核心配置信息之一。

处理方案： 用于记录 JMS 服务当前使用的地址列表，是确认 JMS 服务可连接节点的关键信息。开发人员可通过该日志验证 JMS 部署模式、配置客户端连接参数，或排查连接故障。如需调整地址列表，可通过管控台或 `asadmin` 命令修改 JMS 主机配置。

4.133 AS-JMS-CORE-00004 End point determines destination name, Res name: {0}, JNDI name: {1} descriptor name : {2}

原因： 是一条信息性日志，用于记录 JMS 目标（Destination，如队列或主题）的名称映射关系。该日志表明 JMS 服务通过“端点（End point）”确定了目标的名称，并展示了资源名称、JNDI 名称和描述符名称之间的对应关系，帮助确认 JMS 目标的配置与绑定是否正确。

处理方案： 用于记录 JMS 目标的名称映射关系（资源名称、JNDI 名称、描述符名称），验证 JMS 目标配置的一致性和正确性。开发人员可通过该日志确认目标名称是否正确绑定，排查查找失败问题，理解 JMS 资源的部署与访问逻辑。确保三个名称在配置中匹配，是 JMS 消息通信正常运行的基础。

4.134 AS-JMS-CORE-00005 Successfully set Master broker on JMSRA to {0}

原因： 是一条信息性日志，表示 JMS 资源适配器 (JMSRA, JMS Resource Adapter) 已成功将 主代理 (Master broker) 设置为 {0} 所指定的 JMS 消息代理 (通常是 OpenMQ broker)。这一操作是 JMS 服务高可用或集群环境中的关键配置，确保消息处理的一致性和故障转移能力。

处理方案： 用于确认 JMS 资源适配器 (JMSRA) 已成功绑定到 JMS 集群的主代理，是集群环境中 JMS 服务正常运行的重要标志。该日志帮助管理员监控主代理状态、验证集群配置，并为排查集群协调问题提供关键参考。确保主代理正确识别和切换，是保障 JMS 服务高可用性和消息一致性的基础。

4.135 AS-JMS-CORE-00006 Failed to set Master broker on JMSRA to {0}

cause {1}

原因： 表示 JMS 资源适配器 (JMSRA) 尝试将主代理 (Master broker) 设置为 {0} 时失败，{1} 包含具体的失败原因 (如异常信息)。这一错误通常发生在 JMS 集群环境中，主代理是集群的核心协调节点，设置失败会影响 JMS 服务的高可用性和集群一致性。

处理方案： 根据日志中的 {1} 定位具体故障 (网络、选举、权限等)；恢复主代理可用性、修正集群配置或调整权限；验证主代理设置成功。

4.136 AS-JMS-CORE-00007 Successfully set Cluster brokerlist to {0}

原因： 是一条信息性日志，表示 JMS 服务已成功将集群代理列表 (Cluster brokerlist) 设置为 {0} 所指定的地址集合。该列表包含 JMS 集群中所有消息代理 (broker) 的连接地址，用于实现集群内的负载均衡、消息同步和故障转移，是 JMS 集群正常运行的核心配置。

处理方案： 用于确认 JMS 集群代理列表已成功设置，是集群环境中节点发现、负载均衡和故障转移的基础。管理员可通过该日志验证集群拓扑，排查通信问题，客户端则可参考列表配置连接参数。确保集群代理列表包含所有可用节点，是 JMS 集群高可用性和可靠性的关键。

4.137 AS-JMS-CORE-00008 Failed to set Cluster brokerlist to {0} cause {1}

原因： 表示 JMS 服务尝试将集群代理列表 (Cluster brokerlist) 设置为 {0} 时失败，{1} 包含具体的失败原因 (如异常信息)。集群代理列表是 JMS 集群中所有消息代理 (broker) 的连接地址集合，其设置失败会导致集群节点无法协同工作，影响负载均衡、消息同步和故障转移功能。

处理方案： 根据日志中的 {1} 定位具体故障 (格式、网络、配置等)；修正代理列表配置、恢复节点可用性或统一集群参数；验证列表设置成功。

4.138 AS-JMS-CORE-00009 Failed to shut down Grizzly NetworkListener : {0}

原因：表示 JMS 服务在尝试关闭 Grizzly NetworkListener ({0} 为该监听器的标识, 通常包含名称或端口) 时失败。Grizzly 是AAS底层使用的 NIO 框架, NetworkListener 负责监听网络端口并处理 JMS 消息的网络通信(如客户端连接、消息收发), 关闭失败可能导致资源泄露或服务无法正常停止。

处理方案：根据日志定位阻塞资源(连接、线程、端口); 手动释放资源或重启服务器强制回收; 修复配置或升级版本以避免根源问题。

4.139 AS-JMS-CORE-00010 Error occurs when shutting down JMSRA : {0}

原因：表示在关闭 JMS 资源适配器(JMSRA, JMS Resource Adapter) 时发生错误, {0} 包含具体的异常信息(如异常类型和消息)。JMSRA 是连接 AAS 与 JMS 消息代理(如 OpenMQ) 的核心组件, 负责管理 JMS 连接、会话和消息传递, 其关闭失败可能导致资源泄漏、服务无法正常停止或重启异常。

处理方案：根据日志中的 {0} 定位具体故障(连接、事务、网络等); 手动释放残留资源、修复网络或事务问题; 升级版本或重新部署 JMSRA 以解决组件异常。

4.140 AS-JMS-CORE-00011 Invalid RMI registry port

原因：表示 JMS 服务检测到配置的 RMI 注册表端口(RMI registry port) 无效。RMI (Remote Method Invocation) 注册表用于 JMS 服务(尤其是 OpenMQ 消息代理) 的远程服务注册与发现, 端口无效会导致 JMS 集群节点间的远程通信失败, 影响集群协同、远程管理等功能。

处理方案：定位并修正 RMI 注册表端口配置, 确保其为 1~65535 之间的有效整数; 验证端口未被占用且进程有权限使用; 重启服务使配置生效, 确认错误消失。

4.141 AS-JMS-CORE-00012 No such method {0} in the class {1}

原因：表示 JMS 服务在调用类 {1} 中的方法 {0} 时, 发现该方法不存在。这一错误通常源于 JMS 组件(如 JMS 资源适配器、消息代理客户端) 与依赖类之间的版本不兼容、类定义变更或反射调用错误, 导致方法调用失败, 影响 JMS 服务的初始化或消息处理流程。

处理方案：根据日志定位缺失的方法和类; 统一依赖版本、修复类文件或修正反射逻辑; 重启服务验证错误是否解决。

4.142 AS-JMS-CORE-00013 Connector Resource could not be closed

原因：表示 JMS 服务在尝试关闭连接器资源（Connector Resource）时失败。连接器资源是遵循 JCA（Java EE Connector Architecture）规范的资源对象（如 JMS 连接池、连接工厂关联的资源），用于管理与外部系统（如 JMS 消息代理）的连接，其关闭失败可能导致资源泄漏（如未释放的网络连接、线程或内存），影响服务器稳定性和后续资源使用。

处理方案：根据日志定位资源及失败原因（通信、状态、代码等）；清理异常资源、修复网络或应用代码、升级适配器；验证资源关闭正常，避免泄漏。

4.143 AS-JMS-CORE-00014 rardeployment.mcfcreation_error {0}

原因：该错误表示在部署 JMS 资源适配器（JMSRA）过程中，创建连接工厂（Managed Connection Factory, MCF）时发生失败。连接工厂是 JCA 规范中管理连接的核心组件，负责创建和管理与 JMS 消息代理的物理连接，其创建失败会导致应用无法获取 JMS 连接，进而影响所有依赖 JMS 的功能（如消息发送、接收、MDB 处理）。

处理方案：根据 {0} 中的信息定位具体故障点（配置、网络、依赖等）；修正配置参数、确保代理可用、补充依赖或调整权限；重新部署并验证连接工厂创建成功。

4.144 AS-JMS-CORE-00015 Exception while getting configured RMI port : {0}

原因：表示 JMS 服务在获取配置的 RMI 端口（RMI port）时发生异常，{0} 包含具体的异常信息（如 `NullPointerException`、`NumberFormatException` 等）。RMI 端口是 JMS 服务（尤其是 OpenMQ 消息代理）用于远程方法调用和集群通信的关键配置，获取失败会导致 RMI 服务无法初始化，影响 JMS 集群协同、远程管理等功能。

处理方案：根据 {0} 中的异常定位配置或访问问题；修复配置文件、修正端口值或调整权限；重启服务验证 RMI 端口读取成功。

4.145 AS-JMS-CORE-00016 Failed to start Grizzly proxy for MQ broker

原因：表示启动用于 MQ 消息代理（如 OpenMQ）的 Grizzly 代理时失败。Grizzly 是 AAS 底层的高性能 NIO 网络框架，此处的 Grizzly 代理负责转发客户端与 MQ broker 之间的网络通信（如消息收发、连接管理），其启动失败会导致客户端无法与 MQ broker 建立连接，进而影响所有依赖 JMS 的功能（如消息驱动 Bean（MDB）、JMS 客户端通信）。

处理方案：根据日志异常定位具体故障（端口、网络、依赖等）；释放端口、修复网络、补充依赖或修正配置；重启服务验证代理启动成功，确保 JMS 通信正常。

4.146 AS-JMS-CORE-00017 Failed to create addresslist due to the exception : {0}

原因：表示 JMS 服务在创建 地址列表 (addresslist) 时失败, {0} 包含具体的异常信息 (如 `IllegalArgumentException`、`URISyntaxException` 等)。地址列表是 JMS 客户端或服务端用于连接到消息代理 (如 OpenMQ) 的目标地址集合 (格式为 “协议:// 主机: 端口”), 其创建失败会导致 JMS 服务无法识别可用的消息代理节点, 进而影响客户端连接、消息收发等核心功能。

处理方案：根据 {0} 中的异常定位具体问题 (配置、格式、解析等); 修正 JMS 主机配置、确保地址格式正确或修复依赖; 重启服务验证地址列表创建成功, 恢复 JMS 连接功能。

4.147 AS-JMS-CORE-00018 Error executing method {0} of the class {1}

原因：表示 JMS 服务在执行类 {1} 中的方法 {0} 时发生异常。该错误通常源于方法调用过程中的运行时异常 (如参数错误、状态不一致、依赖故障等), 会导致 JMS 服务的相关功能 (如消息处理、连接管理、集群协同) 中断。

处理方案：根据日志和异常堆栈定位方法、类及具体故障原因; 修复参数合法性、类状态、依赖可用性或代码逻辑; 验证方法执行正常, 恢复 JMS 相关功能。

4.148 AS-JMS-CORE-00019 MDB destination not specified

原因：表示消息驱动 Bean (MDB) 未指定目标 (Destination)。MDB 是依赖 JMS 消息的企业级组件, 需要绑定到特定的 JMS 目标 (如队列 Queue 或主题 Topic) 以接收消息, 若未指定目标, MDB 无法启动, 进而导致消息处理功能失效。

处理方案：通过注解或部署描述符为 MDB 明确配置目标的 JNDI 名称和类型; 确保配置的目标在 AAS 中已存在; 重新部署应用, 验证 MDB 启动成功并能处理消息。

4.149 AS-JMS-CORE-00020 Failed to validate endpoint

原因：表示 JMS 服务在验证 端点 (endpoint) 时失败。这里的 “端点” 通常指 JMS 目标 (如队列 Queue、主题 Topic) 或 MDB (消息驱动 Bean) 与目标的绑定关系, 是 JMS 消息收发的关键节点。验证失败会导致端点无法正常工作, 影响消息的发送、接收或处理。

处理方案：确认端点关联的目标存在且配置合法; 修复 MDB 与目标的绑定配置 (注解或 XML); 检查并赋予必要的访问权限; 验证端点通过验证, 恢复消息通信功能。

4.150 AS-JMS-CORE-00021 Cannot obtain master broker

原因：表示 JMS 服务（通常是 OpenMQ 集群环境）无法获取或确定 主代理（Master broker）。主代理是 JMS 集群的核心节点，负责协调集群状态、管理共享资源（如持久化消息、分布式事务）和处理故障转移，无法获取主代理会导致集群无法正常协同工作，影响消息可靠性和高可用性。

处理方案：确保集群节点存活且网络通畅；修复选举机制或手动指定主代理；统一集群配置，验证主代理状态正常。

4.151 AS-JMS-CORE-00022 Error while loading connector resources during recovery : {0}

原因：表示在 恢复过程（通常是事务恢复或服务重启后的资源恢复）中，加载 JMS 连接器资源（Connector Resources）时发生错误，{0} 包含具体的异常信息（如 ResourceException、NamingException 等）。连接器资源（如 JMS 连接池、连接工厂）是 JMS 服务与消息代理通信的基础，恢复时加载失败会导致未完成的事务无法恢复、资源无法重用，进而影响 JMS 服务的一致性和可用性。

处理方案：根据 {0} 中的异常定位具体故障（配置、网络、日志等）；修复配置、恢复代理、清理日志或调整权限；重启服务验证资源加载成功，确保 JMS 服务恢复正常。

4.152 AS-JMS-CORE-00023 Exception in reading mdb-container configuration : [{0}]

原因：通常与 消息驱动 Bean (MDB) 容器的配置异常相关。

处理方案：该错误的核心是 MDB 容器初始化时无法解析配置，需从 配置文件语法、JMS 资源匹配、部署包完整性、服务器配置四个方向排查。重点查看 AAS 的详细日志（server.log），根据 {0} 占位符中的具体信息（如缺失的 JNDI 名称、XML 错误行号）定位问题，针对性修复。

4.153 AS-JMS-CORE-00024 MQ Resource adapter upgrade started

原因：AAS 在启动或部署过程中输出的一条 信息性日志（非错误），表示 AAS 正在开始升级其内置的 JMS 资源适配器（Resource Adapter）。

处理方案：是正常的信息日志，无需担心。若升级失败导致 JMS 功能异常，需结合后续错误日志和上述方向排查，重点检查资源适配器文件完整性、配置正确性及权限问题。

4.154 AS-JMS-CORE-00025 MQ Resource adapter upgrade completed

原因：是一条 正常信息日志，表示 AAS 内置的 JMS 资源适配器（Resource Adapter）升级已成功完成。

处理方案：是一条确认 JMS 资源适配器升级成功的提示日志，标志着相关组件初始化完成，属于正常流程，无需担心。若后续 JMS 功能出现问题，需结合其他错误日志（如连接失败、资源找不到）进一步排查，而非该日志本身的问题。

题。

4.155 AS-JMS-CORE-00026 Upgrading a MQ resource adapter failed : {0}

原因：是 AAS 在升级 JMS 资源适配器 (Resource Adapter) 时发生的错误日志，表明资源适配器升级过程失败。

{0} 占位符会显示具体的失败原因 (如异常信息、配置错误等)，需结合该详情定位问题。

处理方案：表示 JMS 资源适配器升级失败，核心是根据 {0} 中的具体错误信息针对性修复。常见解决方向包括：修复资源适配器文件、修正配置、调整权限、解决 OpenMQ 服务问题或排除类冲突。若问题复杂，重新安装 AAS 并重建域通常是快速恢复的有效方式。

4.156 AS-JMS-CORE-00027 Check for a new version of MQ installation failed : {0}

原因：是 AAS 在检查 OpenMQ (其内置的消息队列) 是否有新版本时发生的错误日志。{0} 占位符会显示具体失败原因 (如网络问题、文件访问错误等)，需结合该信息定位问题。

处理方案：主要因 MQ 版本检查失败导致，核心是根据 {0} 中的信息解决网络、文件权限或安装完整性问题。若该错误不影响 JMS 核心功能 (如消息收发正常)，可暂时忽略；若伴随 JMS 服务异常，需优先修复 MQ 安装或网络配置。

4.157 AS-ORB-00001 Exception occurred when resolving {0}

原因：是 AAS 服务器在通过 ORB (Object Request Broker) 解析某个对象引用 (通常是 EJB 或 CORBA 对象) 时抛出的核心错误，{0} 占位符会显示具体的解析目标，例如一个 EJB 的 JNDI 名称 (如

`java:global/myapp/MyEJB!com.example.MyEJBRemote`) 或一个 CORBA 对象 ID。这个错误的根本原因是 ORB 在尝试定位、连接或激活目标对象时遇到了无法解决的问题。

处理方案：是一个顶层错误，指示分布式对象调用的初始阶段失败。获取完整日志，定位具体失败的对象和根本原因；从外向内排查，先确认网络和 JNDI 名称，再检查客户端配置，最后深入服务器内部；重点关注 JNDI 名称正确性、网络连通性以及客户端与服务器端的配置和类库版本一致性。

4.158 AS-ORB-00002 No Endpoints selected in com.sun.appserv.iiop.endpoints property.Using {0}:{1} instead

原因：是一条警告日志，它的出现表明 AAS 在启动时，ORB (Object Request Broker) 组件没有找到明确配置的网络端点 (endpoints)，因此它自动选择了一个默认的网络地址和端口来监听 IIOOP (Internet Inter-ORB Protocol) 请求。

处理方案：通常情况下不需要处理。

4.159 AS-ORB-00003 Problem with membership change notification.

Exception occurred : {0}

原因：表明 AAS在处理集群成员变更通知时出现了问题，可能与网络通信、集群配置一致性或实例状态有关。

处理方案：需要综合分析集群的整体状态和各个实例的日志，获取完整的异常信息*({0} 的内容)；检查集群的整体健康状况，识别出可能已经失效或配置错误的节点；重点排查网络连通性，确保所有集群成员之间可以自由通信；确保集群配置在所有节点上保持一致。

4.160 AS-ORB-00004 Could not find an endpoint to send request to

原因：ORB (Object Request Broker) 组件无法找到一个有效的网络端点 (Endpoint) 来发送一个远程请求，这个错误通常发生在尝试进行远程 EJB 调用或其他需要通过 IIOP 协议与另一个 JVM 中的对象通信的场景中。

处理方案：确认目标服务器地址和端口是否正确；确认目标服务器是否正在运行且网络可达；确认目标服务器的 ORB 服务正在监听正确的端口。

4.161 AS-ORB-00005 Unknown host: {0} Exception thrown : {1}

原因：ORB (Object Request Broker) 组件在尝试与另一个主机通信时，无法解析该主机的域名。

处理方案：验证主机名和测试 DNS 解析，检查集群实例的主机配置。

4.162 AS-ORB-00006 No Endpoints selected in com.sun.appserv.iiop.endpoints property. Using JNDI Provider URL {0} instead

原因：是一条警告日志，表明 AAS 客户端在尝试初始化 ORB 时，没有找到明确配置的 IIOP 端点，因此它自动决定使用 JNDI Provider URL 中指定的地址来作为通信的目标端点。

处理方案：这条警告本身通常不会影响客户端的正常功能，因为 AAS 客户端会自动回退到使用 JNDI Provider URL。

4.163 AS-ORB-00007 Exception : {0} thrown for bad provider URL String: {1}

原因：AAS客户端在解析或处理JNDI Provider URL时遇到了问题，因为该 URL 字符串的格式不正确或包含无效信息。

处理方案：获取错误日志中的问题 URL ({1})；核对 URL 格式是否为 `iiop://<host>:<port>`，确认 URL 是否符合 IIOP 协议的基本规范；检查 `jndi.properties` 文件或相关代码，找到配置该 URL 的源头并进行修正。

4.164 AS-WSSECURITY-00001 Request processing failed

原因：是 AAS 在处理 Web Service (WS) 安全相关请求时抛出的通用错误，表明 WS-Security 机制在验证、解析或处理请求时遇到了无法恢复的异常。该错误通常与 WS-Security 配置（如认证、加密、签名）、请求消息格式或安全凭证有效性相关。

处理方案：查看 `server.log` 中的完整堆栈，定位底层异常（如签名失败、认证失败），验证 `sun-jaxws.xml`、`web.xml` 的安全配置是否一致且正确，检查客户端请求消息的安全格式是否合规，确认密钥库、安全域、依赖库的有效性。

4.165 AS-WSSECURITY-00002 SEC2002: Container-auth: wss: Error validating request

原因：AAS 的 WS-Security 容器在尝试验证客户端请求时失败。

处理方案：查看 `server.log` 获取详细原因，针对性地检查客户端请求或服务器配置，通常能很快定位并解决问题。

4.166 AS-WSSECURITY-00003 SEC2003: Container-auth: wss: Error securing response

原因：AAS 在尝试对即将发送给客户端的响应消息进行安全处理（如签名或加密）时失败。

处理方案：查看 `server.log` 获取详细的底层异常，直接定位是密钥库问题、证书问题还是算法问题。

4.167 AS-WSSECURITY-00004 SEC2004: Container-auth: wss: Error securing request

原因：是一个与 Web Service 安全相关的错误。它表明，AAS（或客户端）在尝试对即将发送的请求消息进行安全处理（如签名或加密）时失败。

处理方案：绝大多数情况下，这个问题都与发送方的密钥库配置或证书状态有关，查看发送方日志获取详细的底层异常，直接定位是密钥库问题、证书问题还是算法问题。

4.168 AS-WSSECURITY-00005 SEC2005: Container-auth: wss: Error validating response

原因：是一个 Web Service 安全错误。AAS（作为客户端）在验证从另一个服务端收到的响应消息时失败。

处理方案：检查客户端的信任库配置和服务端的响应消息。

4.169 AS-WSSECURITY-00006 SEC2006: Container-auth: wss: Not a SOAP message context

原因：是一个 Web Service 安全处理错误。它表明，AAS 的 WS-Security 模块在尝试对一个请求或响应进行安全处理时，发现当前的消息上下文（Message Context）并不是一个 SOAP 消息上下文。

处理方案：检查请求的发送方式和应用的配置，通常，问题出在客户端发送了一个不正确的请求，或者应用的 XML 配置文件（`web.xml`，`sun-jaxws.xml`）中存在 URL 模式或过滤器映射的错误。

4.170 AS-WSSECURITY-00007 EJB Webservice security configuration Failure

原因：AAS在解析 EJB Web Service 的安全配置时发现了问题，导致部署失败。

处理方案：查看 `server.log` 的详细部署错误信息，根据线索去修正相应的部署描述符文件或 Java 注解。

4.171 AS-WSSECURITY-00008 Servlet Webservice security configuration Failure

原因：AAS在部署一个基于 Servlet 的 JAX-WS Web Service 时，发现其安全配置存在问题，导致部署失败。

处理方案：查看 `server.log` 的详细部署错误信息，检查安全配置。

4.172 AS-WSSECURITY-00009 BASIC AUTH username/password http header parsing error for {0}

原因：AAS在尝试解析 HTTP `Authorization` 请求头以进行 BASIC 认证时，遇到了格式错误。

处理方案：查看 AAS的 `server.log`。错误信息 `for {0}` 中的 `{0}` 占位符通常会被实际导致错误的 `Authorization` 头内容或请求详情所替换，确保客户端发送的 `Authorization` 头完全符合 HTTP BASIC 认证的规范。

4.173 AS-WSSECURITY-00010 Servlet Webservice security configuration Failure

原因：表示在部署基于 Servlet 的 JAX-WS Web Service 时，安全配置存在问题导致部署失败。

处理方案： 查看 `server.log` 的详细部署错误信息，根据线索去修正相应的部署描述符文件、Java 注解或服务器配置。

4.174 AS-WSSECURITY-00011 Following exception was thrown:

原因： 这是一个通用的包装器错误。它本身并没有提供具体的错误原因，只是表明在 WS-Security 处理过程中发生了一个异常，并且这个异常被捕获和记录下来，真正的问题隐藏在它后面跟着的具体异常信息中。

处理方案： 查看 `server.log` 中 `AS-WSSECURITY-00011` 错误的完整上下文，根据根本异常的类型和消息，参考相应的错误处理方案。

4.175 AS-WEB-ADMIN-00001 Unable to register StatsProvider {0} with Monitoring Infrastructure. No monitoring data will be collected for {1} and {2}

原因： 是一个来自 AAS管理模块的警告。它表明，AAS在尝试为某个组件（如 Web 应用、EJB、HTTP 服务等）注册一个统计数据提供者（StatsProvider）时失败，导致此问题的原因通常与AAS 的监控配置或模块加载有关。

处理方案： 确定服务器实例是否正常启动，这个警告通常不会影晌应用程序的核心功能（只是监控数据缺失），可根据需要进行解决。

4.176 AS-WEB-ADMIN-00002 Current server config is null

原因： 是 AAS启动或配置加载阶段的严重错误，核心原因是服务器无法找到或加载有效的配置实例（通常是 `server-config`），导致核心功能初始化失败。

处理方案： 优先级最高的解决方案是重建域（无需复杂排查，直接恢复默认配置）。若需保留原有应用，可先备份应用文件，重建域后重新部署，避免因手动修改 XML 导致二次错误。

4.177 AS-WEB-ADMIN-00003 The acceptor threads must be at least 1

原因： 是一个配置验证错误。AAS的某个网络监听器（Network Listener）配置的“acceptor threads”（接收线程）数量小于 1，这是无效的。

处理方案： 将网络监听器的 Acceptor Threads数量设置为一个大于或等于 1 的有效整数即可。

4.178 AS-WEB-ADMIN-00004 Listener {0} could not be created, actual reason: {1}

原因： 通常出现在应用服务器启动 / 重启时，因 HTTP/HTTPS 监听器（如 `http-listener-1`、`http-listener-2`）无法初始化导致，直接表现为应用服务器无法对外提供 Web 服务（如 6888/6887 端口无法访问），常见于开发测试环境端口冲突、生产环境低端口权限不足或 SSL 证书配置错误场景。

处理方案： 查看日志里 `{1}` 的具体原因，优先解决端口 / 权限问题。

4.179 AS-WEB-ADMIN-00005 A default virtual server is required. Please use --default-virtual-server to specify this value

原因： Web 容器必须绑定默认虚拟服务器（Virtual Server）才能正常处理 HTTP 请求，若删除了默认虚拟服务器（如 `server`）、配置文件中未指定默认值，或启动时未通过参数指定，就会触发该报错，直接导致应用服务器无法启动或 Web 服务不可用。

处理方案： 启动时临时指定 `--default-virtual-server server`，或在 `domain.xml` 中永久配置默认虚拟服务器。

4.180 AS-WEB-ADMIN-00006 --defaultVS and --default-virtual-server conflict. Please use only --default-virtual-server to specify this value

原因： 在启动命令中同时使用了 `--defaultVS`（简写参数）和 `--default-virtual-server`（完整参数）来指定默认虚拟服务器，两个参数作用完全相同，云服务器不允许重复配置，因此触发该冲突报错。

处理方案： 二选一保留 `--default-virtual-server`，或直接在 `domain.xml` 中永久配置默认虚拟服务器。

4.181 AS-WEB-ADMIN-00007 Attribute value (default-virtual-server = {0}) is not found in list of virtual servers defined in config

原因： 指定的默认虚拟服务器名称（`{0}` 填充的内容，如 `myvs`、`server1`），在应用服务器的 `domain.xml` 配置中不存在对应的 `<virtual-server>` 节点，比如指定了 `default-virtual-server=myvs`，但配置里只有 `server` 这个虚拟服务器，从而触发校验失败。

处理方案： 将默认值改为配置中已存在的虚拟服务器 ID（如 `server`），或新增对应名称的虚拟服务器配置。

4.182 AS-WEB-ADMIN-00008 Http Listener named {0} already exists

原因： 在创建 / 修改 HTTP 监听器时，指定的监听器名称（`{0}` 填充的内容，如 `http-listener-1`、`my-http-listener`），在应用服务器的 `domain.xml` 中已经有同名的 `<network-listener>` 节点，应用服务器不允许监听器重名，因此触发该报错。

处理方案： 删除重复的监听器配置，或修改新监听器的名称为唯一值。

4.183 AS-WEB-ADMIN-00009 Port [{0}] is already taken for address [{1}], please choose another port

原因： 配置的 HTTP/HTTPS 监听器要绑定的端口（{0}，如 8080）和地址（{1}，如 0.0.0.0/[localhost](#)），已被本机其他进程或AAS自身其他监听器占用，无法重复绑。

处理方案： 优先修改AAS监听器端口为未占用值，或停止占用进程 / 清理自身重复配置。

4.184 AS-WEB-ADMIN-00010 Network Listener named {0} already exists.

原因： 在创建 / 新增网络监听器（涵盖 HTTP/HTTPS 监听器）时，指定的监听器名称（{0} 填充的内容，如 `http-listener-1`，在 `domain.xml` 的 `<network-listener>` 节点中已存在同名配置，AAS禁止监听器重名，因此触发该报错。

处理方案： 删除重复的监听器配置，或将新监听器名称改为唯一值。

4.185 AS-WEB-ADMIN-00011 Protocol {0} has neither a protocol nor a port-unification configured

原因： 创建的网络监听器关联了名为 {0} 的协议（如 `my-protocol`），但该协议在 `domain.xml` 中既未指定具体的协议类型（如 `http/1.1`、`ssl`），也未开启端口统一（允许单个端口处理多种协议），监听器无法识别要处理的网络协议，因此初始化失败。

处理方案： 给协议补充具体类型（如 `http/1.1`），或开启端口统一，或直接复用默认协议。

4.186 AS-WEB-ADMIN-00012{0} create failed

原因： 该报错是应用服务器的通用创建失败提示，{0} 会填充具体失败的对象名称（如 `HTTP Listener`、`Virtual Server`、`Protocol`、`Network Listener` 等），本质是对应对象（监听器 / 虚拟服务器 / 协议等）因配置错误、权限不足、资源冲突（如端口占用）、依赖缺失等原因无法初始化。

处理方案： 先从日志中确定失败对象和详细原因，再按对应对象的常见问题（端口、权限、配置）解决。

4.187 AS-WEB-ADMIN-00013 The specified protocol {0} is not yet configured

原因： 在网络监听器中指定了协议名称 {0} (如 `my-http-protocol`、`custom-ssl-protocol`)，但在 `domain.xml` 中完全没有定义该名称的协议节点，监听器找不到对应的协议配置，因此初始化失败。

处理方案： 补全对应名称的协议配置，或改用AAS内置的默认协议。

4.188 AS-WEB-ADMIN-00014 Failed to create http-redirect for {0}: {1}

原因： 是 AAS配置 HTTP 重定向规则时触发的失败提示：`{0}` 填充重定向关联的对象 (如虚拟服务器 `server`、HTTP 监听器 `http-listener-1`)，`{1}` 填充具体失败原因 (如重定向 URL 格式非法、目标端口不可达、权限不足、配置冲突、虚拟服务器不存在等)，本质是重定向规则的参数或依赖配置错误，导致规则无法生效。

处理方案： 先从日志提取具体失败原因，再修正重定向 URL / 目标地址 / 权限等问题。

4.189 AS-WEB-ADMIN-00015 An http element for {0} already exists.

Cannot add duplicate http

原因： 该报错是AAS配置重复导致：`{0}` 填充具体对象 (如协议 `http/1.1`、网络监听器 `http-listener-1`)，在给该对象配置 `<http>` 节点时，发现 `domain.xml` 中已有同名 / 同归属的 `<http>` 元素 (比如给同一个协议重复配置 `http/1.1` 类型的 `<http>` 节点)，AAS禁止同一对象下重复定义 `<http>` 元素，因此触发该报错。

处理方案： 删除重复的 `<http>` 节点，或合并不同参数到单个 `<http>` 元素。

4.190 AS-WEB-ADMIN-00016 An http-redirect element for {0} already exists. Cannot add duplicate http-redirect

原因： 该报错是AAS配置 HTTP 重定向时的重复冲突问题：`{0}` 填充重定向关联的对象 (如虚拟服务器 `server`、HTTP 监听器 `http-listener-1`)，你在给该对象新增 `<http-redirect>` 重定向规则时，发现 `domain.xml` 中已有同名对象的 `<http-redirect>` 节点 (比如给 `server` 虚拟服务器重复配置两次 HTTP 重定向)，AAS禁止同一对象下重复定义重定向规则，因此触发该报错。

处理方案： 删除重复的重定向节点，或修改现有节点参数满足新需求。

4.191 AS-WEB-ADMIN-00017 {0} protocol already exists. Cannot add duplicate protocol

原因： 该报错是AAS协议配置重复导致：`{0}` 填充具体的协议名称 (如 `http/1.1`、`https`、`custom-protocol`)，你在创建 / 新增协议时，发现 `domain.xml` 中已有同名的 `<protocol>` 节点 (比如重复定义 `http/1.1` 协议)，AAS禁止协议名称重复，因此触发该报错。

处理方案： 删除重复的协议节点，或将新协议名称改为唯一值并更新关联监听器配置。

4.192 AS-WEB-ADMIN-00018 Failed to create protocol {0}

原因： 该报错是AAS自定义协议时的通用失败提示，{0} 填充具体协议名称（如 `custom-http`、`my-ssl-protocol`），本质原因包括：协议名称重复、协议类型配置非法（如未指定 `http/1.1` / `ssl`）、SSL 证书配置错误、依赖的传输层（如 `tcp`）未配置、权限不足无法写入协议配置等。

处理方案： 先从日志找具体失败原因，再针对性修正协议名称 / 配置 / 权限，或直接复用默认协议。

4.193 AS-WEB-ADMIN-00019 {0} create failed: {1}

原因： 该报错是AAS针对传输层（Transport）相关对象（{0} 通常填充 `tcp`、`udp` 或自定义传输层名称）的创建失败通用提示，{1} 是具体失败原因。核心根因包括：传输层名称重复、传输层配置非法（如未指定线程池、协议类型不匹配）、权限不足无法创建套接字、线程池不存在 / 配置错误等。

处理方案： 先从日志提取具体失败原因，再修正传输层名称 / 线程池 / 权限配置，或复用默认 `tcp` 传输层。

4.194 AS-WEB-ADMIN-00020 {0} create failed. Given class is not a ProtocolFilter: {1}

原因： 配置的协议过滤器类未实现 `ProtocolFilter` 接口，或类加载 / 配置错误，导致类型不匹配。

处理方案： 修正过滤器类使其实现接口并正确加载，或删除非法的过滤器配置。

4.195 AS-WEB-ADMIN-00021 {0} create failed. Given class is not a ProtocolFinder: {1}

原因： 该报错是AAS配置端口统一（Port Unification）时的类型不匹配问题，{0} 填充失败的协议查找器（ProtocolFinder）名称（如 `custom-finder`），{1} 填充你指定的查找器类名（如 `com.example.MyFinder`）。

处理方案： 修正查找器类使其实现接口并正确加载，或删除非法的查找器配置（关闭端口统一）。

4.196 AS-WEB-ADMIN-00022 {0} transport already exists. Cannot add duplicate transport

原因： 该报错是AAS传输层（Transport）配置重复导致：{0} 填充具体的传输层名称（如 `tcp`、`udp`、`custom-tcp-transport`），在创建 / 新增传输层时，发现 `domain.xml` 中已有同名的 `<transport>` 节点

(比如重复定义默认的 `tcp` 传输层) , AAS禁止传输层名称重复, 因此触发该报错。

处理方案: 删除重复的传输层节点并合并参数, 或将新传输层名称改为唯一值并更新关联监听器。

4.197 AS-WEB-ADMIN-00023 Failed to create transport {0}

原因: 该报错是AAS传输层 (Transport, 如 `tcp / udp / 自定义传输层`) 创建失败的通用提示, `{0}` 填充具体传输层名称。核心根因包括: 传输层名称重复、配置参数非法 (如线程池不存在、类型错误)、权限不足 (无法创建套接字 / 绑定端口)、依赖的线程池配置错误、传输层类型不支持 (如非 `tcp / udp`) 等。

处理方案: 先从日志找具体失败原因, 再修正传输层名称 / 配置 / 权限, 或直接复用默认 `tcp` 传输层。

4.198 AS-WEB-ADMIN-00024 Please use only networklisteners

原因: 该报错是AAS配置语法 / 参数规范问题: 在命令行、配置文件 (`domain.xml`) 或管理控制台中, 使用了AAS已废弃的监听器相关关键词 (如 `httplisteners` 、 `listeners`) , 或混用了不同版本的监听器配置术语, 而AAS明确要求统一使用 `networklisteners` (网络监听器) 作为核心配置标识, 因此触发该规范校验报错。

处理方案: 将命令 / 配置中的 `httplisteners / listeners` 等关键词统一改为 `networklisteners / network-listener` 。

4.199 AS-WEB-ADMIN-00025 Virtual Server named {0} already exists

原因: 该报错是AAS虚拟服务器配置重复导致: `{0}` 填充具体的虚拟服务器名称 (如 `server` 、 `my-virtual-server`) , 在创建 / 新增虚拟服务器时, 发现 `domain.xml` 中已有同名的 `<virtual-server>` 节点 (比如重复创建默认的 `server` 虚拟服务器) , AAS禁止虚拟服务器名称重复, 因此触发该报错。

处理方案: 删除重复的虚拟服务器配置, 或将新虚拟服务器名称改为唯一值。

4.200 AS-WEB-ADMIN-00026 {0} create failed

原因: 该报错是AAS针对虚拟服务器 (Virtual Server) 的通用创建失败提示 (`{0}` 几乎都是虚拟服务器名称, 如 `server` 、 `my-virtual-server`) , 是 00025 (重名)、00005 (无默认虚拟服务器) 等具体报错的兜底提示。本质原因包括: 虚拟服务器名称重复、配置参数非法 (如 `host` 地址格式错误、`docroot` 路径不存在 / 权限不足)、依赖的监听器 / 协议未配置、AAS权限不足无法写入配置等。

处理方案: 先从日志找具体失败原因, 再修正名称 / 路径 / 权限 / 默认配置。

4.201 AS-WEB-ADMIN-00028 Specified http listener, {0}, doesn't exist

原因：该报错是AAS配置引用错误导致：在配置虚拟服务器、重定向规则、协议关联等操作时，引用了名为 {0} 的 HTTP 监听器，但该监听器并未在 `domain.xml` 中创建（或名称拼写错误、已被删除），AAS找不到对应的监听器配置，因此触发该引用校验报错。

处理方案：修正引用的监听器名称为已存在的有效值，或创建对应的监听器，或删除无效引用

4.202 AS-WEB-ADMIN-00029 {0} delete failed

原因：该报错是AAS删除 Web 相关组件时的通用失败提示，{0} 填充具体要删除的对象名称（如虚拟服务器 `server`、网络监听器 `http-listener-1`、协议 `http/1.1`、传输层 `tcp` 等）。

处理方案：先从日志找具体原因，解除依赖 / 修正权限 / 核对名称后再删除，核心内置组件禁止删除。

4.203 AS-WEB-ADMIN-00030 {0} Network Listener doesn't exist

原因：该报错是AAS操作 / 配置引用错误导致：在修改、删除、关联网络监听器（{0} 填充监听器名称，如 `http-listener-2`）时，引用的这个监听器并未在 `domain.xml` 中创建（或名称拼写错误、已被删除、大小写不匹配），AAS找不到对应的网络监听器配置，因此触发该存在性校验报错。

处理方案：修正引用的监听器名称为已存在的有效值，或先创建对应的监听器再操作。

4.204 AS-WEB-ADMIN-00031 Deletion of NetworkListener {0} failed

原因：删除网络监听器失败，核心原因是监听器被依赖、是内置组件、权限不足或处于运行状态。

处理方案：先解除虚拟服务器对监听器的依赖，非内置监听器可在停止域后删除，内置监听器仅修改配置不删除。

4.205 AS-WEB-ADMIN-00032 {0} http-redirect doesn't exist

原因：操作 HTTP 重定向时引用了不存在的配置（对象无重定向、名称拼写错误），触发存在性校验失败。

处理方案：终止针对不存在重定向的操作，或先给目标对象创建重定向再执行后续操作，同时核对名称拼写。

4.206 AS-WEB-ADMIN-00033 Deletion of http {0} failed

原因：删除 http 元素失败，核心原因是该元素为协议 / 监听器的核心依赖、元素不存在、权限不足或对象运行中。

处理方案：核心依赖的 http 元素禁止删除，仅可修改参数；非核心冗余的 http 元素需停止域、赋权后再删除。

4.207 AS-WEB-ADMIN-00034 Deletion of http-redirect {0} failed

原因：删除 HTTP 重定向配置失败，核心原因是配置生效中、配置不存在、权限不足或文件锁定 / 损坏。

处理方案：先停止域（或禁用重定向）解除配置生效状态，赋权后再执行删除，不存在的配置直接终止操作。

4.208 AS-WEB-ADMIN-00035 {0} protocol doesn't exist

原因：操作或配置中引用了未创建、拼写错误或已删除的协议，触发存在性校验失败。

处理方案：修正引用的协议名称为已存在的有效值，先创建对应的协议再执行后续操作。

4.209 AS-WEB-ADMIN-00036 {0} protocol is being used in the network listener {1}

原因：要操作的协议被网络监听器关联依赖，AAS禁止删除被使用的协议。

处理方案：删除协议前需先修改监听器的协议关联（解除依赖），修改协议配置可直接操作无需解除依赖。

4.210 AS-WEB-ADMIN-00037 Deletion of Protocol {0} failed

原因：删除协议失败，核心原因是协议被监听器依赖、是核心内置组件、协议不存在或权限不足。

处理方案：删除自定义协议前先解除监听器依赖，核心内置协议仅可修改配置不删除，操作前核对协议存在性和权限。

4.211 AS-WEB-ADMIN-00038 {0} delete failed: {1}.

原因：该报错是 GlassFish 针对 Web 组件删除操作的增强型通用失败提示，{0} 删除失败，具体原因：{1}。

处理方案：先根据 {1} 定位原因，解除依赖/赋权/核对名称后再删除，核心内置组件禁止删除。

4.212 AS-WEB-ADMIN-00039 No {0} element found with the name {1}

原因：操作指定的 {1} 名称 {0} 类型元素不存在，或名称拼写错误。

处理方案：先创建目标元素再执行操作，或修正元素名称为已存在的有效值。

4.213 AS-WEB-ADMIN-00040 {0} transport is being used in the network listener {1}

原因：{0} 传输层正被网络监听器 {1} 使用，无法执行删除 / 修改核心属性等操作。

处理方案：删除自定义传输层前先修改监听器的传输层关联（解除依赖），核心内置传输层仅可修改配置不删除。

4.214 AS-WEB-ADMIN-00041 Deletion of Transport {0} failed

原因： 删除传输层失败，核心原因是传输层被监听器依赖、是核心内置组件、传输层不存在或权限不足。

处理方案： 删除自定义传输层前先解除监听器依赖，核心 `tcp` 传输层仅可修改配置不删除。

4.215 AS-WEB-ADMIN-00042 {0} transport doesn't exist

原因： 操作或配置中引用了未创建、拼写错误或已删除的传输层，触发存在性校验失败。

处理方案： 将引用改为已存在的传输层（如 `tcp`），或先创建目标传输层再执行操作。

4.216 AS-WEB-ADMIN-00043 {0} delete failed

原因： 该报错是AAS针对传输层（Transport）的通用删除失败提示（区别于 00029 全局删除失败、00041 传输层专属删除失败），`{0}` 填充具体传输层名称（如 `tcp`、`custom-tcp`）。

处理方案： 删除自定义传输层前先解除监听器依赖，内置 `tcp` 传输层仅可修改配置不删除。

4.217 AS-WEB-ADMIN-00044 Specified virtual server, {0}, doesn't exist

原因： 操作或配置中引用了未创建、拼写错误或已删除的虚拟服务器，触发存在性校验失败。

处理方案： 将引用改为已存在的虚拟服务器（如 `server`），或先创建目标虚拟服务器再执行操作。

4.218 AS-WEB-ADMIN-00045 Specified virtual server, {0}, can not be deleted because it is referenced from http listener, {1}

原因： 指定的虚拟服务器 {0} 无法删除，原因是它被 HTTP 监听器 {1} 引用。

处理方案： 先在虚拟服务器配置中移除被引用的监听器，解除依赖后再执行删除操作。

4.219 AS-WEB-ADMIN-00046 Monitoring Registry does not exist. Possible causes are 1) Monitoring is not turned on or at a lower level 2) The corresponding container (web, ejb, etc.) is not loaded yet

原因： 监控注册表不存在，根源是监控未开启 / 级别低或 Web 容器未加载。

处理方案： 先将 Web 容器监控级别设为 `HIGH` 并重启域，再确认容器正常启动。

4.220 AS-WEB-UTIL-00001 Resource "{0}" is missing

原因：指定名称为 {0} 的资源不存在，根源是路径错误、资源丢失、JNDI 配置错误或权限不足。

处理方案：先明确缺失的资源类型，再核对路径 / 重新部署资源 / 修正 JNDI 配置 / 赋权。

4.221 AS-WEB-UTIL-00002 Failed tracking modifications of "{0}" : {1}

原因：AAS监控指定资源 {0} 的修改时失败，{1} 会明确具体原因（权限 / 不存在 / 锁定等）

处理方案：先从 {1} 定位原因，优先解决权限问题（最常见），再核对资源路径 / 释放锁定 / 清理磁盘。

4.222 AS-WEB-UTIL-00003 WebappClassLoader.findClassInternal({0}) security exception: {1}

原因：加载 {0} 类时触发安全异常，根源是安全策略限制、类加载权限不足或签名验证失败。

处理方案：优先通过 `server.policy` 精准授权（避免全局放宽），其次检查类加载范围 / 文件权限，测试时可临时关闭安全管理器排障。

4.223 AS-WEB-UTIL-00004 Security Violation, attempt to use Restricted Class: {0}

原因：应用调用了AAS禁止访问的受限类 {0}，触发安全违规拦截。

处理方案：先替换代码避免调用受限类（最安全），第三方依赖触发时精准放宽安全策略，禁止全局关闭安全管理器。

4.224 AS-WEB-UTIL-00005 Class {0} has unsupported major or minor version numbers, which are greater than those found in the Java Runtime Environment version {1}

原因：类 {0} 的编译 Java 版本高于运行AAS的 JRE 版本 {1}，版本不兼容。

处理方案：优先重新编译自研代码为服务器 JRE 兼容版本，第三方 jar 包替换为低版本兼容版，避免盲目升级服务器 JRE。

4.225 AS-WEB-UTIL-00006 Unable to load class with name [{0}], reason: {1}

原因： 加载 {0} 类失败, {1} 是定位根源的关键（类找不到 / 依赖缺失 / 版本不兼容等）。

处理方案： 先通过 {1} 确定原因, 优先排查类是否存在、依赖是否缺失, 再处理权限 / 版本 / 加载器问题。

4.226 AS-WEB-UTIL-00007 The web application [{0}] registered the JDBC driver [{1}] but failed to unregister it when the web application was stopped. To prevent a memory leak, the JDBC Driver has been forcibly unregistered

原因： Web 应用 {0} 启动时注册了 JDBC 驱动 {1}, 但停止 / 卸载应用时未主动注销该驱动, AAS为防止内存泄漏, 强制注销了这个驱动。

处理方案： 优先使用AAS JNDI 数据源, 无法修改代码时通过监听器主动注销驱动。

4.227 AS-WEB-UTIL-00008 JDBC driver de-registration failed for web application [{0}]

原因： 应用主动注销 JDBC 驱动失败, 根源是权限不足、驱动实例获取失败、驱动本身有缺陷或多应用冲突。

处理方案： 优先改用 GlassFish JNDI 数据源, 无法修改代码时优化注销逻辑 + 升级驱动 + 放宽权限。

4.228 AS-WEB-UTIL-00009 Exception closing input stream during JDBC driver de-registration for web application [{0}]

原因： 注销 JDBC 驱动时关闭输入流抛出异常, 根源是流已关闭 / 被锁定、驱动 IO 缺陷、权限不足或注销逻辑不完整。

处理方案： 优先改用 GlassFish JNDI 数据源, 无法修改代码时优化流关闭逻辑（空值判断 + 异常捕获）+ 升级驱动 + 释放资源锁定。

4.229 AS-WEB-UTIL-00010 This web container has not yet been started

原因： 操作时 Web 容器未启动 / 未就绪, 根源是操作时机过早或容器启动失败。

处理方案： 优先检查操作时机（延迟执行）, 再排查容器启动问题（端口占用、配置损坏、组件未加载）。

4.230 AS-WEB-UTIL-00011 Failed to check for ThreadLocal references for

web application [{0}]

原因： AAS检测应用 {0} 的 ThreadLocal 引用时失败，根源是 ThreadLocal 使用不当、权限不足、容器组件缺陷或第三方库冲突。

处理方案： 优先规范 ThreadLocal 使用（使用后必 remove），再放宽安全策略 / 升级AAS / 排查库冲突，禁用检测仅作为应急方案。

4.231 AS-WEB-UTIL-00012 Unable to determine string representation of key of type [{0}]

原因： AAS无法将类型为 {0} 的键转换为字符串，根源是键类型未重写 toString()、是特殊类型，或容器 / 库不兼容。

处理方案： 优先给自定义类重写 toString()，特殊类型提前转换为基础字符串，复杂场景封装工具类统一处理。

4.232 AS-WEB-UTIL-00013 Unknown

原因： 这是AAS Web 工具模块的「兜底型未知错误」，意味着容器在处理 Web 应用请求 / 操作时触发了未被具体分类的异常，没有更详细的错误信息来定位具体原因。简单来说就是系统知道出错了，但无法识别错误的具体类型。

处理方案： 先开启详细日志找到底层真实异常，再按具体异常类型（空指针 / IO / 内存等）解决，最后恢复日志级别。

4.233 AS-WEB-UTIL-00014 Unable to determine string representation of value of type [{0}]

原因： AAS无法将类型为 {0} 的值转换为字符串，根源是值类型未重写 toString()、是特殊不可序列化类型，或体积过大。

处理方案： 优先给自定义类重写安全的 toString()，特殊类型（流 / 集合 / 二进制）转换为基础字符串（JSON/Base64），大对象改用数据库存储。

4.234 AS-WEB-UTIL-00015 The web application [{0}] created a ThreadLocal with key of type [{1}] (value [{2}]). The ThreadLocal has been correctly set to null and the key will be removed by GC

原因： 是正常提示而非错误，表示应用的 ThreadLocal 已被正确清理，无内存泄漏。

处理方案：无需修复代码，仅在日志冗余时调整日志级别。

4.235 AS-WEB-UTIL-00016 The web application [{0}] created a ThreadLocal with key of type [{1}] (value [{2}]) and a value of type [{3}] (value [{4}]) but failed to remove it when the web application was stopped. Threads are going to be renewed over time to try and avoid a probable memory leak

原因：应用 {0} 的 ThreadLocal 未清理，触发内存泄漏预警，AAS会重启线程规避风险。

处理方案：所有 ThreadLocal 必在 finally 块中调用 remove ()，配合 Filter/Listener 全局清理，优化线程池参数加速回收。

4.236 AS-WEB-UTIL-00017 Failed to find class sun.rmi.transport.Target to clear context class loader for web application [{0}]. This is expected on non-Sun JVMs

原因：是兼容性提示而非错误，因使用非 Sun JVM 导致容器找不到专属类，不影响应用功能。

处理方案：无需修复代码，仅在日志冗余时调整日志级别。

4.237 AS-WEB-UTIL-00018 Failed to clear context class loader referenced from sun.rmi.transport.Target for web application [{0}]

原因：AAS清理 sun.rmi.transport.Target 关联的类加载器失败，根源是权限限制、RMI 状态异常或 JVM 版本兼容。

处理方案：优先忽略（不影响功能），日志冗余时调整级别，有内存泄漏顾虑时放宽安全策略 / 优化 RMI 配置。

4.238 AS-WEB-UTIL-00019 Removed [{0}] ResourceBundle references from the cache for web application [{1}]

原因：是正常操作提示，表示应用 {1} 的 {0} 个资源包缓存被容器清理，无异常、无风险。

处理方案：无需修复代码，仅在日志冗余时调整日志级别。

4.239 AS-WEB-UTIL-00020 Failed to clear ResourceBundle references for web application [{0}]

原因： AAS清理应用 {0} 的 ResourceBundle 引用失败，根源是静态持有资源包、并发访问、权限限制或容器缺陷。

处理方案： 优先修复代码，避免静态持有 ResourceBundle，其次放宽权限 / 调整缓存配置，屏蔽日志仅作为应急方案。

4.240 AS-WEB-UTIL-00021 Illegal JAR entry detected with name {0}

原因： AAS检测到名称为 {0} 的非法 JAR 条目，根源是条目名称含非法字符、长度超限、JAR 包损坏或违反规范。

处理方案： 先定位非法条目和所属 JAR 包，再按类型修复（清理非法字符 / 缩短名称 / 重新打包），核心是保证 JAR 条目符合 Java 规范。

4.241 AS-WEB-UTIL-00022 Unable to validate JAR entry with name {0}

原因： AAS无法验证名称为 {0} 的 JAR 条目，根源是 JAR 包损坏、条目为空、权限限制或容器 / JDK 不兼容。

处理方案： 先定位问题条目和 JAR 包，优先重新获取 / 打包 JAR 包（解决损坏问题），其次调整权限 / 兼容版本。

4.242 AS-WEB-UTIL-00023 Unable to create {0}

原因： AAS无法创建类型为 {0} 的资源，根源是资源依赖缺失、权限 / 内存不足、配置错误或容器故障。

处理方案： 先定位 {0} 的具体资源类型和真实异常，再按类型修复（补全依赖、调整权限 / 内存、修复配置）。

4.243 AS-WEB-UTIL-00024 Unable to delete {0}

原因： AAS无法删除 {0} 指向的文件 / 目录，根源是文件被锁定、权限不足、路径不存在或文件系统异常。

处理方案： 先定位目标路径和失败原因，优先释放占用进程 / 调整权限（解决核心问题），其次优化代码释放句柄 / 定时清理。

4.244 AS-WEB-UTIL-00025 Unable to read data for class with name [{0}]

原因： AAS无法读取名称为 {0} 的类数据，根源是类文件损坏 / 缺失、路径错误、类加载器异常或依赖冲突。

处理方案： 先定位问题类和真实失败原因，优先重新编译 / 打包类文件（解决损坏问题），其次修正路径 / 补全依赖 / 调整类加载器。

4.245 AS-WEB-UTIL-00026 Unable to purge bean classes from BeanELResolver

原因：AAS无法清理 `BeanELResolver` 中的 Bean 类缓存，根源是 Bean 类静态引用、类加载器泄漏、EL 解析器配置异常或权限限制。

处理方案： 优先重启域快速恢复，其次避免静态持有 Bean 类、清理 `ThreadLocal` 解决类加载器泄漏，手动清理 EL 缓存作为兜底。

4.246 AS-WEB-UTIL-00027 extra-class-path component [{0}] is not a valid pathname

原因：AAS配置的 `extra-class-path` 组件 `{0}` 是无效路径名，根源是路径不存在、格式错误、权限异常或配置格式错误。

处理方案： 先定位配置位置和错误路径值，优先修正路径存在性 / 拼写 / 格式（解决核心问题），其次调整权限 / 分隔符。

4.247 AS-WEB-UTIL-00028 The clearReferencesStatic is not consistent in context.xml for virtual servers

原因：AAS多虚拟服务器的 `context.xml` 中 `clearReferencesStatic` 取值不一致，导致静态引用清理策略混乱。

处理方案： 统一所有层级（全局 / 虚拟服务器 / 应用）的 `clearReferencesStatic` 取值（优先设为 `true`），删除冲突的局部配置。

4.248 AS-WEB-UTIL-00029 class-loader attribute dynamic-reload-interval in sun-web.xml not supported

原因：`sun-web.xml` 中配置的 `class-loader` 属性 `dynamic-reload-interval` 不被当前版本支持，根源是属性废弃、配置格式错误或版本不兼容。

处理方案： 先移除无效属性消除报错，再通过 `asadmin` 命令或 `domain.xml` 配置替代的动态重载规则。

4.249 AS-WEB-UTIL-00030 Property element in sun-web.xml has null 'name' or 'value'

原因： sun-web.xml 中 <property> 元素的 name / value 为空，根源是配置显式空值、语法解析异常或动态变量未定义。

处理方案： 定位并修复所有空值的 <propert>节点，确保 name/value 非空、XML 语法合法，动态值添加默认值。

4.250 AS-WEB-UTIL-00031 Ignoring invalid property [{0}] = [{1}]

原因： 检测到属性 {0}={1} 无效并忽略，根源是属性名不支持、值类型 / 范围非法、归属错误或格式错误。

处理方案： 先定位无效属性的名称 / 值和配置位置，再修正属性名 / 值（或移至正确配置文件），无意义的属性直接删除。

4.251 AS-WEB-UTIL-00032 The xml element should be [{0}] rather than [{1}]

原因： 解析 XML 时发现标签名错误（期望 {0}，实际是 {1}），根源是拼写 / 大小写错误、版本不兼容、层级错误或命名空间问题。

处理方案： 先定位错误标签名和配置文件，将 {1} 修正为 {0}，同时确保标签嵌套层级和命名空间符合规范。

4.252 AS-WEB-UTIL-00033 This is an unexpected end of document

原因： 解析 XML 时遇到「文档意外结束」，根源是标签未闭合、文件截断、特殊字符未转义或文件为空。

处理方案： 先定位损坏的 XML 文件，补全闭合标签 / 自闭合标签、恢复完整内容、转义特殊字符，再校验 XML 语法。

4.253 AS-WEB-UTIL-00034 Unexpected type of ClassLoader. Expected: java.net.URLClassLoader, got: {0}

原因： 期望 URLClassLoader 但获取到 {0} 类型的类加载器，根源是自定义 ClassLoader、依赖冲突、版本不兼容或部署方式错误。

处理方案： 先定位异常类加载器类型，优先移除自定义 ClassLoader / 冲突依赖，确保继承 URLClassLoader，匹配AAS与 JDK 版本。

4.254 AS-WEB-UTIL-00035 Unable to load class {0}, reason: {1}

原因： 加载类 {0} 失败，{1} 是具体原因（类缺失、依赖不足、版本不兼容等）。

处理方案： 先提取 {0} 和 {1} 的核心信息，按原因补全类 / 依赖、修复版本 / 权限 / 初始化问题。

4.255 AS-WEB-UTIL-00036 Invalid URLClassLoader path component: [{0}] is neither a JAR file nor a directory

原因：URLClassLoader 的路径组件 {0} 既非 JAR 文件也非目录，根源是路径类型非法、不存在、格式错误或权限异常。

处理方案：先定位无效路径和配置位置，修正为合法的「目录 / JAR 文件」路径，或直接移除无效路径。

4.256 AS-WEB-UTIL-00037 Error trying to scan the classes at {0} for annotations in which a ServletContainerInitializer has expressed interest

原因：扫描路径 {0} 下的类以查找 SCI 关注的注解时失败，根源是路径异常、类 / JAR 损坏、权限不足或 SCI 配置异常。

处理方案：先定位扫描路径和失败根因，优先修正路径合法性、修复损坏的类 / JAR、调整权限 / 内存，再解决 SCI 依赖 / 配置问题。

4.257 AS-WEB-UTIL-00038 Ignoring [{0}] during Tag Library Descriptor (TLD) processing

原因：AAS处理 TLD 时忽略了 {0}，根源是 TLD 文件无效、路径 / 引用错误、依赖缺失或权限不足。

处理方案：先定位被忽略的对象和原因，优先修复 TLD 语法 / 版本、修正 JSP 引用路径、补全依赖。

4.258 AS-WEB-UTIL-00039 Unable to determine TLD resources for [{0}] tag library, because class loader [{1}] for [{2}] is not an instance of java.net.URLClassLoader

原因：加载 TLD 的类加载器 {1} 不是 URLClassLoader，导致AAS无法定位 {0} 标签库的资源。

处理方案：优先恢复 URLClassLoader 类型（移除自定义 ClassLoader / 冲突依赖、Spring Boot 改 WAR 打包），再确保 TLD 资源可被扫描。

4.259 AS-WEB-CORE-00001 Configuration error: Must be attached to a Context

原因： Web 组件未正确依附于 Context（Web 应用上下文），根源是配置层级错误、Context 未创建、资源引用错误或组件初始化时机不当。

处理方案： 将组件配置移至应用级 Context 配置文件（web.xml/glassfish-web.xml），修复配置语法 / 资源映射，确保组件在 Context 初始化后执行。

4.260 AS-WEB-CORE-00002 Authenticator[{0}]: {1}

原因： 身份验证器 {0} 执行时出现 {1} 错误，根源是配置错误、资源异常、权限问题或运行时异常。

处理方案： 先提取 {0} 和 {1} 的核心信息，按错误类型修正 Realm 配置、数据源、页面路径或角色映射。

4.261 AS-WEB-CORE-00003 Exception getting debug value

原因： 读取 debug 配置值时抛出异常，根源是配置值类型非法、配置文件异常、核心模块损坏或参数未定义。

处理方案： 修正 debug 配置值为合法类型，修复配置文件语法 / 权限，恢复容器核心模块或补充未定义的参数。

4.262 AS-WEB-CORE-00004 Unexpected error forwarding or redirecting to login page

原因： AAS转发 / 重定向到登录页时失败，根源是登录页路径错误、页面本身异常、Context 初始化失败或权限问题。

处理方案： 确保登录页路径合法（不在 WEB-INF 下、存在且可访问），修复页面语法错误，保证请求链不被中断。

4.263 AS-WEB-CORE-00005 Started

原因： 是正常启动日志，而非错误，无需修复。

处理方案： 无需处理。

4.264 AS-WEB-CORE-00006 Stopped

原因： 是正常停止日志，而非错误，无需修复。

处理方案： 无需处理。

4.265 AS-WEB-CORE-00007 Process session destroyed on {0}

原因： 是正常日志，而非错误，表明会话被正常销毁。

处理方案： 无需处理。

4.266 AS-WEB-CORE-00008 Process request for "{0}"

原因：是正常日志，而非错误，标记容器开始处理指定路径 {0} 的请求。

处理方案：无需处理。

4.267 AS-WEB-CORE-00009 Principal {0} has already been authenticated

原因：用户 {0} 已完成认证，容器检测到重复认证请求并抛出该提示，本质是「重复触发认证逻辑」。

处理方案：前端添加防重复提交，后端判断用户认证状态后跳过重复认证逻辑。

4.268 AS-WEB-CORE-00010 Checking for SSO cookie

原因：是正常日志，而非错误，标记容器正在检查 SSO Cookie。

处理方案：无需处理。

4.269 AS-WEB-CORE-00011 SSO cookie is not present

原因：是正常日志，而非错误，仅表明客户端无 SSO Cookie。

处理方案：无需处理。

4.270 AS-WEB-CORE-00012 Checking for cached principal for {0}

原因：是正常日志，而非错误，标记容器正在检查已缓存的认证主体（性能优化行为）。

处理方案：无需处理。

4.271 AS-WEB-CORE-00013 Found cached principal {0} with auth type {1}

原因：是正常的信息级日志，无任何错误含义，表明容器命中了已缓存的用户认证信息。

处理方案：无需处理。

4.272 AS-WEB-CORE-00014 No cached principal found, erasing SSO cookie

原因：是正常的状态维护日志，而非错误。

处理方案：无需处理。

4.273 AS-WEB-CORE-00015 Associate sso id {0} with session {1}

原因： 是正常的信息级日志，无任何错误含义，表明容器完成了 SSO ID 与用户会话的关联。

处理方案： 无需处理。

4.274 AS-WEB-CORE-00016 Registering sso id {0} for user {1} with auth type {2}

原因： 是正常的信息级日志，无任何错误含义，表明容器为已认证用户成功注册了 SSO ID。

处理方案： 无需处理。

4.275 AS-WEB-CORE-00017 Looking up certificates

原因： 是正常日志，而非错误，标记容器正在加载 / 校验 SSL / 客户端证书。

处理方案： 无需处理。

4.276 AS-WEB-CORE-00018 No certificates included with this request

原因： 是正常日志，而非错误，仅表明请求未携带客户端证书。

处理方案： 无需处理。

4.277 AS-WEB-CORE-00019 No client certificate chain in this request

原因： 是正常日志，仅表明请求缺少完整客户端证书链。

处理方案： 无需处理。

4.278 AS-WEB-CORE-00020 Cannot authenticate with the provided credentials

原因： 用户提交的认证凭据无效，或 Realm / 代码配置错误导致认证失败。

处理方案： 优先核对用户名 / 密码 / 证书有效性；检查 Realm 配置（数据源、加密方式、表字段）；修复自定义代码对认证流程的干扰。

4.279 AS-WEB-CORE-00021 Unable to determine target of zero-arg dispatcher

原因：分发操作 (forward/include) 缺少有效目标路径，容器无法确定请求转发 / 包含的资源。

处理方案：修复 Servlet/JSP 中无参数 / 空路径的分发代码；检查 URL 映射和框架转发规则配置；确保所有分发操作都指定具体的目标资源路径。

4.280 AS-WEB-CORE-00022 Unable to acquire RequestDispatcher for {0}

原因：容器无法为 {0} 路径创建 `RequestDispatcher`，因目标资源不存在 / 路径错误 / 权限不足。

处理方案：验证并修正转发 / 包含的目标路径（拼写、大小写、层级）；动态路径添加非空 / 合法性校验；确保 Servlet 映射配置与转发路径匹配。

4.281 AS-WEB-CORE-00023 Unable to acquire RequestDispatcher for {0} in servlet context {1}

原因：在 Servlet 上下文 {1} 中，容器无法为路径 {0} 创建 `RequestDispatcher`，因上下文 / 路径无效、跨上下文权限未开启。

处理方案：解析日志中的 {0}（目标路径）和 {1}（上下文），验证资源是否存在；修正路径拼写 / 大小写 / 层级，动态路径添加严格校验。

4.282 AS-WEB-CORE-00024 Error invoking AsyncListener

原因：`AsyncListener` 的回调方法执行时抛出未捕获异常，或异步上下文操作不当。

处理方案：给监听器所有回调方法添加 `try-catch`，定位并修复具体异常（空指针、重复操作等）；正确注册监听器实例，添加非空 / 状态校验；确保异步上下文最终调用 `complete()` / `dispatch()` 释放资源；按需调整AAS异步超时 / 线程池配置。

4.283 AS-WEB-CORE-00025 Asynchronous dispatch already in progress, must call ServletRequest.startAsync first

原因：异步分发已在进行，或未启动异步就执行异步操作，违反了 `AsyncContext` 的生命周期规则。

处理方案：严格遵循「启动异步→处理业务→分发 / 完成」的流程，先调用 `startAsync()` 再执行异步操作；操作前校验异步状态 (`isAsyncStarted()`)，避免重复 / 过期操作；多线程场景下串行化异步操作，防止状态冲突。

4.284 AS-WEB-CORE-00026 Must not call AsyncContext.addListener after the container-initiated dispatch during which ServletRequest.startAsync was called has returned to the container

原因： Web 容器检测到在「容器触发的分发流程（调用 `ServletRequest.startAsync()` 的那次分发）已返回容器后」，才尝试调用 `AsyncContext.addListener()` 注册异步监听器（`AsyncListener`）。这是 Servlet 异步处理的生命周期错误，违反了「异步监听器必须在启动异步的分发流程未结束前注册」的规则，导致监听器无法被容器正确绑定，进而引发异步事件监听失效（如超时、错误无法被捕获）。

处理方案： 严格遵循「`startAsync()` 后立即在主线程注册监听器」的规则，注册完成后再执行其他操作；禁止在异步线程中注册监听器，特殊场景可通过 `onStartAsync()` 补充注册；封装安全注册方法，提前校验注册时机，避免静默失败。

4.285 AS-WEB-CORE-00027 Must not call AsyncContext.setTimeout after the container-initiated dispatch during which ServletRequest.startAsync was called has returned to the container

原因： Web 容器检测到在「调用 `ServletRequest.startAsync()` 的容器触发分发流程已返回容器后」，才尝试调用 `AsyncContext.setTimeout()` 设置异步超时时间。这是 Servlet 异步处理的生命周期错误，违反了「异步上下文的基础配置（如超时时间）必须在启动异步的分发流程未结束前完成」的规则，导致超时时间设置失效，可能引发异步请求无限阻塞、资源泄漏。

处理方案： 严格遵循「`startAsync()` 后立即在主线程设置超时」的规则，设置完成后再执行其他操作；所有参数解析、耗时逻辑前置到 `startAsync()` 之前，避免主线程耗时导致分发流程提前返回；封装安全设置方法，校验时机和参数合法性，避免静默失败；特殊场景需调整超时，仅在 `onStartAsync()` 回调中补充设置。

4.286 AS-WEB-CORE-00028 The connector has already been initialized

原因： Web 容器检测到尝试对一个已经完成初始化的连接器（Connector，如 HTTP/HTTPS 连接器、线程池连接器等）重复执行初始化操作。这是容器启动 / 配置层面的错误，通常由连接器配置重复加载、自定义代码手动触发连接器初始化、容器重启时状态未清理，或部署脚本重复执行初始化命令导致，会引发连接器启动失败、端口占用，甚至容器无法正常启动。

处理方案： 定位重复初始化的连接器，检查配置 / 脚本 / 代码中是否有重复执行的初始化逻辑；创建连接器前先检查状态，避免重复创建；自定义代码中添加状态校验，仅初始化一次；清理容器缓存、删除重复配置，确保连接器配置唯一；正常停止容器，避免状态残留。

4.287 AS-WEB-CORE-00029 Error registering connector

原因： Web 容器在注册连接器（Connector，如 HTTP/HTTPS 监听器、线程池连接器等）时发生异常。这是容器启动 / 配置层面的严重错误，表明连接器的注册流程（将连接器绑定到容器、端口、线程池等）失败，会直接导致连接器无法启动、对应端口无法访问，甚至整个AAS容器启动失败。

处理方案： 从日志堆栈定位具体失败原因（端口占用 / 配置错误 / 权限等）；针对性修复：释放端口、修正配置、赋予权限、重建证书 / 线程池；无法定位时，重置连接器为默认配置并清理缓存。

4.288 AS-WEB-CORE-00030 Failed to instantiate HttpHandler

原因： Web 容器无法实例化 HTTP 处理器（HttpHandler）。这是容器请求处理层面的严重错误，`HttpHandler` 是 AAS处理 HTTP 请求的核心组件（负责解析请求、路由、响应处理），实例化失败会导致容器无法处理任何 HTTP 请求，表现为端口可访问但返回 500 错误、请求超时，甚至容器启动后无法提供服务。

处理方案： 从日志堆栈定位具体失败原因（类缺失 / 构造异常 / 版本冲突）；修复自定义 `HttpHandler` 构造函数（添加无参构造、捕获异常）；解决类加载冲突（排除冲突依赖、重置容器核心 JAR）；匹配AAS与 JDK 版本，确保兼容性。

4.289 AS-WEB-CORE-00031 mod_jk invalid Adapter implementation: {0}

原因： Web 容器检测到mod_jk 连接器配置的适配器（Adapter）实现类 `{0}` 无效。

处理方案： 修正 `workers.properties` 中的适配器类名为AAS专属的 `JkAdapter`；在AAS中开启 AJP13 连接器，确保端口未被占用；匹配 mod_jk 与AAS 版本，确保适配器类存在且兼容；验证 Apache 配置，确保 mod_jk 模块正确加载。

4.290 AS-WEB-CORE-00032 Protocol handler instantiation failed: {0}

原因： Web 容器无法实例化指定的协议处理器（Protocol handler）`{0}`（如 `http/1.1`、`ajp/1.3`、`https/1.1` 等）。协议处理器是容器处理特定网络协议请求的核心组件，实例化失败会导致对应协议的连接器（如 HTTP/HTTPS/AJP 监听器）无法启动，表现为容器启动失败、对应端口无法访问，或请求无法被正确解析。

处理方案： 从日志中提取失败的协议名称，修正配置拼写 / 格式；修复协议处理器依赖（重置 Grizzly JAR、排除冲突依赖）；修正非法参数（端口、SSL 配置），确保权限 / 端口可用；匹配AAS与 JDK 版本，解决兼容性问题。

4.291 AS-WEB-CORE-00033 The connector has already been started

原因： Web 容器检测到尝试对一个已经处于启动（running）状态的连接器（Connector，如 HTTP/HTTPS/AJP 监听器）重复执行启动操作。这是连接器生命周期管理的典型错误，违反了「连接器只能在停止（stopped）状态下执行启

动」的规则，会导致连接器启动命令失败、容器日志持续报错，严重时可能引发连接器状态异常（如端口占用但无法处理请求）。

处理方案：启动连接器前必须校验状态；脚本 / 代码添加状态判断，避免重复触发启动命令；修改配置遵循「停止→修改→启动」顺序，清理容器状态残留；优先使用容器自动启动，减少手动操作。

4.292 AS-WEB-CORE-00034 Protocol handler start failed: {0}

原因：Web 容器在启动指定的协议处理器（Protocol handler）{0}（如 http/1.1、https/1.1、ajp/1.3）时失败。协议处理器是连接器处理特定网络协议的核心组件，启动失败会导致对应连接器（如 HTTP 监听器）无法提供服务，表现为容器启动后对应端口无法访问、请求超时，或容器直接启动失败。

处理方案：从日志堆栈定位具体失败原因（端口 / SSL / 权限 / 依赖）；针对性修复：释放端口、修正 SSL 配置、赋予权限、匹配版本、调整资源限制；无法定位时，重置协议处理器为默认配置并清理缓存。

4.293 AS-WEB-CORE-00035 Coyote connector has not been started

原因：对尚未启动的 Coyote 连接器执行操作（如停止、分发请求），违反连接器生命周期规则。

处理方案：操作连接器前必须校验状态，仅对 `running` 状态的连接器执行停止 / 修改等操作；修复连接器启动失败的根因（端口占用、配置错误），确保连接器能正常启动；脚本 / 代码添加容错逻辑，避免重复操作未启动的连接器；配置连接器自动启动，减少手动操作遗漏。

4.294 AS-WEB-CORE-00036 Protocol handler destroy failed: {0}

原因：协议处理器销毁失败，根源是未遵循生命周期顺序、资源未释放、权限不足或依赖冲突。

处理方案：销毁前必须先停止协议处理器，遵循「stop → destroy」顺序；强制释放被占用的端口 / 线程资源，修复自定义销毁逻辑的缺陷；排除依赖冲突、赋予足够权限，避免容器异常终止导致资源锁定。

4.295 AS-WEB-CORE-00037 An exception or error occurred in the container during the request processing

原因：请求处理过程中出现未捕获的异常，根源是应用代码缺陷、容器配置错误、资源不足或环境异常。

处理方案：从日志堆栈定位具体异常类型和代码位置；修复代码缺陷（空指针、SQL 错误等），添加全局异常处理；调整容器配置（请求体大小、线程池、内存），修复资源 / 权限问题；最小化测试验证容器本身是否正常，缩小问题范围。

4.296 AS-WEB-CORE-00038 HTTP listener on port {0} has been disabled

原因：指定端口的 HTTP 监听器被禁用（手动配置、启动失败自动禁用、脚本 / 集群配置冲突），导致端口无法监听请求。

处理方案：通过 `asadmin` 命令或直接修改 `domain.xml` 启用监听器；修复监听器启动失败的根因（端口占用、配置错误），避免被自动禁用；集群环境中确保配置一致，脚本中删除错误的禁用逻辑；添加启动校验，确保关键监听器始终启用。

4.297 AS-WEB-CORE-00039 Error parsing client cert chain into array of `java.security.cert.X509Certificate` instances

原因：AAS解析客户端 X509 证书链时失败，根源是证书格式非法、链不完整、信任库配置错误、算法不兼容或证书过期。

处理方案：验证并修复客户端证书格式 / 完整性，确保为标准 X509 且链顺序正确；将根证书导入AAS信任库，启用兼容的加密算法；修正 HTTPS 连接器的客户端认证配置，代理场景下确保证书转发；替换过期证书，临时禁用 CRL 校验（应急）。

4.298 AS-WEB-CORE-00040 No Host matches server name {0}

原因：请求的服务器名称（域名 / IP）与AAS配置的虚拟主机列表不匹配，导致请求无法路由。

处理方案：添加 / 修改虚拟主机配置，绑定请求的服务器名称；修复反向代理的 Host 头转发，确保请求头合法；重新部署应用并绑定正确的虚拟主机；配置默认虚拟主机兜底，处理未匹配的请求。

4.299 AS-WEB-CORE-00041 Cannot use this object outside a servlet's service method or outside a filter's doFilter method

原因：在 Servlet/Filter 的请求处理方法外使用了绑定请求生命周期的对象，违反 Servlet 规范。

处理方案：异步处理使用 `AsyncContext` 而非自定义线程，避免跨线程操作请求对象；仅存储请求中的数据而非整个请求对象，杜绝全局变量缓存请求对象；确保响应流操作在关闭前完成，Filter 中断链前完成请求读取；框架中正确使用请求级对象的注入方式。

4.300 AS-WEB-CORE-00042 Cannot set a null ReadListener object

原因：尝试为非阻塞 IO 设置 null 的 ReadListener，违反 Servlet 3.1+ 规范参数校验规则。

处理方案：绝对禁止调用 `setReadListener(null)`，必须传入有效的 ReadListener 实现类实例；对可能返回 null 的监听器创建逻辑增加 null 校验，异常时返回兜底的空实现；确保所有代码分支都能创建非 null 的 ReadListener，条件判断不遗漏；设置监听器前必须启用非阻塞模式（`setNonBlocking(true)`）。

4.301 AS-WEB-CORE-00043 Cannot set a null WriteListener object

原因：尝试为非阻塞写入操作设置 null 的 WriteListener，违反 Servlet 3.1+ 规范参数校验规则。

处理方案：绝对禁止调用 `setWriteListener(null)`，必须传入有效的 WriteListener 实现类实例；对可能返回 null 的监听器创建逻辑增加 null 校验，异常时返回兜底的空实现；确保所有代码分支都能创建非 null 的 WriteListener，条件判断不遗漏；设置监听器前必须启用非阻塞模式（`setNonBlocking(true)`），并正确管理异步上下文。

4.302 AS-WEB-CORE-00044 Failed to skip {0} characters in the underlying buffer of CoyoteReader on readLine()

原因：CoyoteReader 的 `readLine()` 方法跳过字符时越界，根源是请求体读取方式冲突、缓冲区状态异常、编码 / 大小不兼容。

处理方案：禁止混用 `getParameter()` 和 `getReader()` 读取请求体，统一读取方式；提前设置字符编码，调整缓冲区大小适配超大请求体；非阻塞 IO 场景下使用 `ReadListener` 替代阻塞的 `readLine()`；避免在输入流关闭 / 读取完毕后调用 `readLine()`。

4.303 AS-WEB-CORE-00045 Stream closed

原因：对已关闭的输入 / 输出流执行读写操作，违反流的生命周期约束。

处理方案：严格遵循「读写→刷新→关闭」的流操作顺序，关闭后不再操作；调用 `sendRedirect()` / `sendError()` 后立即返回，不执行后续响应操作；禁止缓存 / 复用请求 / 响应流对象，仅缓存流中的数据；异步处理中通过 `AsyncContext` 管理流，统一控制生命周期。

4.304 AS-WEB-CORE-00046 Already set read listener

原因：对同一个请求对象重复设置 ReadListener，违反 Servlet 3.1+ 非阻塞 IO 规范。

处理方案：通过请求属性标记「是否已设置 ReadListener」，所有设置逻辑前先校验；统一在 Filter 或 Servlet 中设置 ReadListener，避免链路重复；读取失败时复用已有 ReadListener 实现重试，而非重复调用 `setReadListener()`；捕获重复设置的异常，避免程序崩溃。

4.305 AS-WEB-CORE-00047 Cannot set ReaderListener for non-async or non-upgrade request

原因：在非异步 / 非升级的普通请求中设置 ReaderListener，违反 Servlet 非阻塞 IO 规范。

处理方案： 异步请求：严格遵循「启动 AsyncContext → 启用非阻塞 → 设置 ReaderListener」的顺序；同步请求：放弃设置 ReaderListener，使用传统阻塞方式读取请求体；增加请求类型校验（`request.isAsyncStarted()`），避免无效设置。

4.306 AS-WEB-CORE-00048 Error in invoking ReadListener.onDataAvailable

原因： `ReadListener.onDataAvailable()` 方法内抛出未捕获的异常，根源是 IO 操作违规、空指针 / 类型转换错误、异步资源管理不当。

处理方案： 全覆盖捕获 `onDataAvailable()` 内的所有异常；遵循非阻塞读取规则，仅使用 `isReady()` + `read(byte[])` 读取请求体；确保异步上下文资源正常释放；对变量 / 参数做严格的空指针 / 边界校验。

4.307 AS-WEB-CORE-00049 The WriteListener has already been set

原因： 对同一个响应对象重复设置 `WriteListener`，违反 Servlet 3.1+ 非阻塞 IO 规范。

处理方案： 通过请求属性标记「是否已设置 WriteListener」，所有设置逻辑前先校验；不同业务逻辑通过调整 Listener 内部状态实现，而非重复设置；重试逻辑通过 Listener 自定义重试方法实现，捕获重复设置异常；Filter 与 Servlet 协同，统一设置 Listener 并复用。

4.308 AS-WEB-CORE-00050 Cannot set WriteListener for non-async or non-upgrade request

原因： 在非异步 / 非升级的普通请求中设置 `WriteListener`，违反 Servlet 3.1+ 非阻塞 IO 规范。

处理方案： 异步请求：严格遵循「启动 AsyncContext → 启用非阻塞 → 设置 WriteListener」的顺序；同步请求：放弃设置 `WriteListener`，使用传统阻塞方式写入响应；增加请求类型校验（`request.isAsyncStarted()`），避免无效设置。

4.309 AS-WEB-CORE-00051 Error in invoking WriteListener.onWritePossible

原因： `WriteListener.onWritePossible()` 方法内抛出未捕获的异常，根源是 IO 操作违规、空指针 / 类型转换错误、异步上下文管理不当。

处理方案： 全覆盖捕获 `onWritePossible()` 内的所有异常，避免抛到容器层；遵循「判空→判态→写入」的非阻塞写入原则，确保流操作安全；严格管理异步上下文状态，确保资源正常释放；对变量 / 参数做严格的空指针 / 边界校验。

4.310 AS-WEB-CORE-00052 `getReader()` has already been called for this request

原因：对同一个请求重复 / 混合调用 `getReader()` / `getInputStream()`，违反 Servlet 请求体读取的单一性原则。

处理方案：一次性读取请求体并缓存，后续复用缓存数据而非重复调用读取方法；用

`HttpServletRequestWrapper` 包装请求，支持 Filter/Servlet 重复读取；禁止 `getParameter()` 与 `getReader()` / `getInputStream()` 混用；字符流 / 字节流二选一，统一读取方式。

4.311 AS-WEB-CORE-00053 `getInputStream()` has already been called for this request

原因：对同一个请求重复 / 混合调用 `getInputStream()` / `getReader()`，违反 Servlet 请求体读取的单一性原则。

处理方案：一次性读取请求体并缓存为字节数组，后续复用缓存数据而非重复调用 `getInputStream()`；用

`HttpServletRequestWrapper` 包装请求，支持 Filter/Servlet 重复读取字节流；禁止 `getParameter()` 与 `getInputStream()` 混用，文件上传场景使用专用框架统一解析；二进制数据必须用 `getInputStream()` 读取，文本数据可任选但需统一。

4.312 AS-WEB-CORE-00054 Unable to determine client remote address from proxy (returns null)

原因：代理未传递客户端 IP 头，或 AAS 未配置解析代理 IP，导致无法识别客户端远程地址。

处理方案：代理层：配置 `X-Forwarded-For` / `X-Real-IP` 头传递真实客户端 IP；AAS 层：启用代理 IP 解析，配置信任的代理 IP；代码层：兼容解析代理头，兜底使用 `remoteAddr`。

4.313 AS-WEB-CORE-00055 Unable to resolve IP address {0} into host name

原因：AAS 尝试对客户端 / 代理 IP 进行 DNS 反向解析 (IP → 主机名) 时失败，占位符 `{0}` 是具体解析失败的 IP 地址。

处理方案：禁用解析。

4.314 AS-WEB-CORE-00056 Exception thrown by attributes event listener

原因： 属性事件监听器的回调方法抛出未捕获的异常，根源是空指针 / 类型转换错误、业务逻辑异常、监听器配置错误。

处理方案： 监听器方法内全覆盖捕获异常，记录详细日志但不抛到容器层；对属性值、事件对象做严格的判空 / 类型校验；耗时业务异步处理，监听器保持轻量化；确保监听器配置正确（无重复注册、有默认构造方法）。

4.315 AS-WEB-CORE-00057 Cannot call setAttribute with a null name

原因： 调用 `setAttribute()` 时传入 `null` 作为属性名，违反 Servlet 规范。

处理方案： 调用 `setAttribute()` 前强制校验属性名（非 `null`、非空字符串）；封装安全设置属性的工具方法，统一拦截非法属性名；动态属性名设置默认值，避免返回 `null`；增加异常捕获，友好处理非法属性名场景。

4.316 AS-WEB-CORE-00058 Unable to determine canonical name of file [{0}] specified for use with sendfile

原因： 使用 `sendfile` 传输文件时，无法解析文件的规范路径名，根源是路径无效、权限不足、编码错误或 `sendfile` 配置问题。

处理方案： 所有文件操作使用绝对路径，调用 `getCanonicalFile()` 解析规范路径；传输前校验文件存在性、可读性、路径合法性；处理中文 / 特殊字符路径，防止编码错误；异常时禁用 `sendfile` 降级为普通 IO 传输。

4.317 AS-WEB-CORE-00059 Unable to set request character encoding to {0} from context {1}, because request parameters have already been read, or ServletRequest.getReader() has already been called

原因： 请求字符编码设置时机过晚（参数 / 请求体已读取），违反 Servlet 规范导致设置无效。

处理方案： 编码设置必须是请求处理的第一个操作，优先通过全局 Filter 实现；编码 Filter 设为第一个执行，且仅负责设置编码不读取参数；GET 请求需额外配置 URI 编码，POST 请求确保 `setCharacterEncoding()` 在 `getParameter()` / `getReader()` 之前。

4.318 AS-WEB-CORE-00060 Attempt to re-login while the user identity already exists

原因： 在同一请求 / 会话中重复调用 `request.login()`，或登录逻辑重复执行，违反 Servlet 身份验证规范。

处理方案： 登录前通过 `getUserPrincipal()` 校验认证状态，避免重复调用；登录逻辑单一入口、精准路径匹配，避免全局执行；前端防重复提交，后端做幂等处理；避免声明式与程序式验证混用。

4.319 AS-WEB-CORE-00061 changeSessionId has been called without a session

原因：调用 `changeSessionId()` 时无有效 `HttpSession`，违反会话操作规范。

处理方案：严格遵循「先创建会话，后更换 ID」的顺序，调用前执行非空校验；对 `changeSessionId()` 调用添加异常捕获，兜底处理无会话场景；仅在登录 / 权限提升等关键节点更换会话 ID，避免无意义调用。

4.320 AS-WEB-CORE-00062 Cannot create a session after the response has been committed

原因：创建会话时响应已提交，无法修改响应头传递 `JSESSIONID`，违反 `Servlet` 规范。

处理方案：严格遵循「先创建会话，后输出响应」的顺序，避免响应提交后操作会话；禁止不必要的 `flush` 操作，增大缓冲区延迟响应提交；创建会话前通过 `response.isCommitted()` 校验状态，异常时重定向兜底。

4.321 AS-WEB-CORE-00063 Invalid URI encoding; using HTTP default

原因：URI 编码配置值非法 / 位置错误，导致 AAS 降级使用默认编码。

处理方案：在 `domain.xml` 的 `http-listener` 节点配置正确的 URI 编码（`UTF-8`）；区分 GET（URI 编码）和 POST（请求体编码）的处理方式；系统层面配置 `UTF-8` 语言环境，避免编码兼容问题。

4.322 AS-WEB-CORE-00064 Invalid URI character encoding; trying ascii

原因：URI 编码配置无效 / URI 包含 ASCII 无法处理的字符，AAS 降级为 ASCII 解析导致乱码 / 失败。

处理方案：全局配置 URI 编码为 `UTF-8`，并配置 JVM 编码环境；前端强制对 GET 参数做 `encodeURIComponent` 编码，特殊字符转为 Unicode 转义；后端添加解码兜底逻辑，捕获编码异常并友好处理。

4.323 AS-WEB-CORE-00065 Request is within the scope of a filter or servlet that does not support asynchronous operations

原因：异步配置不完整或操作违反 `Servlet` 异步规范。

处理方案：优先通过配置 `asyncSupported=true` 解决全链路支持问题，其次规范代码中异步上下文的使用，最后优化容器和框架配置。

4.324 AS-WEB-CORE-00066 ServletRequest.startAsync called again without any asynchronous dispatch, or called outside the scope of any such dispatch, or called again within the scope of the same dispatch

原因：违反了 Servlet 异步规范中一个分发周期内只能启动一次异步上下文的规则。

处理方案：调用 `startAsync()` 前检查 `isAsyncStarted()`，避免重复调用；规范 `AsyncContext` 的生命周期，完成后显式调用 `complete()`；框架场景中避免手动异步与框架自动异步冲突，容器层面升级 / 配置以修复状态管理 Bug。

4.325 AS-WEB-CORE-00067 Response already closed

原因：对已关闭 / 提交的响应执行写操作。

处理方案：所有响应操作前通过 `isCommitted()` 检查状态，避免无效写入；规范多线程 / 异步场景的响应操作，加锁并控制生命周期；框架 / 容器层面避免在请求后期（如 `afterCompletion()`）操作响应，调整缓冲配置减少提前提交。

4.326 AS-WEB-CORE-00068 ServletRequest.startAsync called outside the scope of an async dispatch

原因：违反了 `startAsync()` 仅允许在异步分发作用域内调用的 Servlet 异步规范。

处理方案：禁止在异步线程中直接调用 `startAsync()`，需通过 `dispatch()` 触发异步分发后，在分发的 Servlet / 过滤器中调用；调用前通过 `isAsyncDispatching()` / `isAsyncStarted()` 检查状态，优先复用现有异步上下文；框架场景中使用 `DeferredResult` / `Callable` 替代手动操作，升级容器修复状态标记 Bug。

4.327 AS-WEB-CORE-00069 The request has not been put into asynchronous mode, must call ServletRequest.startAsync first

原因：尝试操作异步上下文（`AsyncContext`）相关功能，但未先调用 `ServletRequest.startAsync()` 将请求置入异步模式。

处理方案：所有异步操作前，先调用 `startAsync()` 启动异步模式，且通过 `isAsyncStarted()` 校验状态；避免在 `complete()` / `dispatch()` 后操作 `AsyncContext`，确保仅在异步模式有效期内操作；框架场景中遵循封装逻辑，过滤器中先执行链再操作 `AsyncContext`，全链路开启 `asyncSupported=true`。

4.328 AS-WEB-CORE-00070 Request already released from asynchronous mode

原因： 请求的「异步模式」已被「释放 (released)」 (即异步上下文终结)，却仍尝试执行依赖异步模式的操作。

处理方案： 所有异步操作前强制校验 `isAsyncStarted()`，返回 `false` 则终止操作；仅在异步模式生效时调用 `complete() / dispatch()`，避免重复释放；框架场景中依赖封装的异步组件 (如 `DeferredResult`)，不手动干预异步释放流程。

4.329 AS-WEB-CORE-00071 Unable to perform error dispatch

原因： 异步异常触发的错误分发因请求 / 响应状态异常或配置问题失败。

处理方案： 异步线程中全面捕获异常，未提交响应时手动处理错误，避免容器触发分发；规范错误页面配置，确保路径合法且开启异步支持；框架 / 容器层面优化，通过全局异常处理器接管错误，升级容器修复分发 Bug。

4.330 AS-WEB-CORE-00072 Request.{0} is called without multipart configuration. Either add a @MultipartConfig to the servlet, or a multipart-config element to web.xml

原因： 处理文件上传的 Servlet 缺失 `multipart` 配置。

处理方案： 为上传 Servlet 添加 `@MultipartConfig` 注解或 `web.xml` 中的 `<multipart-config>` 配置；框架场景中配置 `MultipartResolver` (Spring MVC) 或直接使用 Spring Boot 自动配置；校验请求类型和容器权限，确保配置参数合理。

4.331 AS-WEB-CORE-00073 This should not happen-breaking background lock: sess = {0}

原因： 应用服务器 特有的会话 (Session) 锁机制异常。

处理方案： 减少 Session 并发操作，缩短锁持有时间；规范 Session 生命周期管理，避免销毁后操作；调整 AAS 锁 / 线程池配置，升级版本修复底层 Bug；非集群场景禁用分布式 Session，降低锁复杂度。

4.332 AS-WEB-CORE-00074 Must not use request object outside the scope of a servlet's service or a filter's doFilter method

原因： 请求对象被传递到其合法作用域之外操作。

处理方案： 仅在 Servlet `service()` / Filter `doFilter()` 执行期间操作请求对象，异步场景通过 `AsyncContext` 获取；需跨线程 / 全局使用请求数据时，仅拷贝数据而非持有请求对象引用；框架场景中规范异步方法、拦截器的请求操作边界，禁止在请求生命周期结束后操作。

4.333 AS-WEB-CORE-00075 Error during finishResponse

原因： 容器执行响应收尾 `finishResponse()` 时触发的资源 / 状态异常。

处理方案： 规范响应流操作，禁止手动关闭流，捕获客户端断开连接异常；避免响应状态冲突，确保响应操作一次性完成，异步场景规范 `complete()` 调用；优化框架 / 容器配置，升级修复底层 Bug，排查资源泄漏。

4.334 AS-WEB-CORE-00076 getWriter() has already been called for this response

原因： 混用响应的字符流与字节流，或重复调用流方法。

处理方案： 单个响应仅调用一次 `getWriter()` 或 `getOutputStream()`，全程复用流引用；跨 Filter/Servlet 协调流的使用，避免重复获取；框架场景中交由框架统一处理流，或使用响应包装器 / `StreamingResponseBody` 规避冲突。

4.335 AS-WEB-CORE-00077 getOutputStream() has already been called for this response

原因： 对同一个 `HttpServletResponse` 对象，重复调用 `getOutputStream()`（获取字节输出流），或在调用 `getOutputStream()` 后又调用 `getWriter()`（字符输出流）。

处理方案： 单一响应：全程仅使用一种流类型（字节流 / 字符流）；单一流实例：获取一次流后复用引用，不重复调用 `getOutputStream()`；单一操作主体：跨 Filter/Servlet 协调流的使用，通过包装器统一管理。

4.336 AS-WEB-CORE-00078 Cannot reset buffer after response has been committed

原因： 响应已提交却尝试重置缓冲区。

处理方案： 所有重置操作前强制检查 `isCommitted()`，未提交才执行；增大缓冲区、禁止手动 `flush()`，避免响应提前提交；规范条件分支和框架拦截器的响应逻辑，确保提交状态一致。

4.337 AS-WEB-CORE-00079 Cannot change buffer size after data has been written

原因：在已向 `HttpServletResponse` 的输出流 / 字符流写入数据后，尝试调用 `setBufferSize()` 修改响应缓冲区大小。

处理方案：将 `setBufferSize()` 放在响应操作的最前置位置（获取流、写入数据、设置编码 / 头前），或通过 AAS全局配置缓冲区大小，彻底避免代码中动态调整的时机问题。

4.338 AS-WEB-CORE-00080 Cannot call sendError() after the response has been committed

原因：响应已提交却调用 `sendError()`。

处理方案：调用 `sendError()` 前强制检查 `resp.isCommitted()`，未提交才执行；调整逻辑顺序，先校验错误条件，再写入响应内容；增大缓冲区、禁止手动 `flush()`，延迟响应提交；框架场景使用全局异常处理器 / 容器错误页面，替代手动 `sendError()`。

4.339 AS-WEB-CORE-00081 Cannot call sendRedirect() after the response has been committed

原因：响应已提交却调用 `sendRedirect()`。

处理方案：重定向前强制检查 `resp.isCommitted()`，未提交才执行；调整逻辑顺序，先校验重定向条件，再写入响应内容；增大缓冲区、禁止手动 `flush()`，延迟响应提交；框架场景使用封装的重定向方式，避免手动操作。

4.340 AS-WEB-CORE-00082 Null response object

原因：`HttpServletResponse` 对象为 `null`，无法执行响应操作。

处理方案：所有操作响应的代码前强制添加非空校验，避免空引用；修复自定义响应包装 / 传递逻辑，确保包装器传入非空响应，异步场景使用 `AsyncContext`；升级AAS版本、调整容器资源配置，解决底层初始化异常。

4.341 AS-WEB-CORE-00083 Not allowed to call this javax.servlet.ServletContext method from a ServletContextListener that

was neither declared in the application's deployment descriptor nor annotated with WebListener

原因：项目中存在一个自定义的 `ServletContextListener` 监听器实现类，这个监听器既没有在 web 应用的核心部署描述文件 `web.xml` 中做显式的声明配置，也没有在监听器类上添加 JavaEE 标准的 `@WebListener` 注解，导致容器无法合法识别该监听器的身份，容器判定这个「未合法注册」的监听器调用 `ServletContext` 的相关核心方法属于非法操作。

处理方案：在自定义 `ServletContextListener` 实现类的类名上方，直接添加 JavaEE 标准注解

`javax.servlet.annotation.WebListener`，无需修改任何配置文件，容器会自动扫描并合法注册该监听器。或在项目的 `WEB-INF/web.xml` 部署描述文件中，对该自定义监听器做显式的注册声明，在 `<web-app>` 根标签内添加 `<listener>` 标签，配置监听器的完整类路径。

全国统一服务热线
4008-555-800



金蝶天燕云计算股份有限公司(简称“金蝶天燕云”)成立于2000年,前身为“金蝶中间件公司”,是金蝶集团旗下新一代软件基础云平台服务商,云计算国家标准制定企业,国家信创产业核心软件企业。金蝶天燕是国家863重点研发计划与核高基重大专项承接企业,也是“两网一站四库十二金”国家重点工程的基础平台提供商,产品广泛应用于政府、军工、金融、能源等关键行业,累计服务客户总数超过10万家。

Apusic
金蝶天燕

云计算国家标准制定企业
金蝶集团旗下基础软件企业
信息技术应用创新核心企业
官网: www.apusic.com

