

CA Workload Automation AE

API Reference Guide

r11.3



This documentation, which includes embedded help systems and electronically distributed materials, (hereinafter referred to as the "Documentation") is for your informational purposes only and is subject to change or withdrawal by CA at any time.

This Documentation may not be copied, transferred, reproduced, disclosed, modified or duplicated, in whole or in part, without the prior written consent of CA. This Documentation is confidential and proprietary information of CA and may not be disclosed by you or used for any purpose other than as may be permitted in (i) a separate agreement between you and CA governing your use of the CA software to which the Documentation relates; or (ii) a separate confidentiality agreement between you and CA.

Notwithstanding the foregoing, if you are a licensed user of the software product(s) addressed in the Documentation, you may print or otherwise make available a reasonable number of copies of the Documentation for internal use by you and your employees in connection with that software, provided that all CA copyright notices and legends are affixed to each reproduced copy.

The right to print or otherwise make available copies of the Documentation is limited to the period during which the applicable license for such software remains in full force and effect. Should the license terminate for any reason, it is your responsibility to certify in writing to CA that all copies and partial copies of the Documentation have been returned to CA or destroyed.

TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS DOCUMENTATION "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT WILL CA BE LIABLE TO YOU OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE, DIRECT OR INDIRECT, FROM THE USE OF THIS DOCUMENTATION, INCLUDING WITHOUT LIMITATION, LOST PROFITS, LOST INVESTMENT, BUSINESS INTERRUPTION, GOODWILL, OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED IN ADVANCE OF THE POSSIBILITY OF SUCH LOSS OR DAMAGE.

The use of any software product referenced in the Documentation is governed by the applicable license agreement and such license agreement is not modified in any way by the terms of this notice.

The manufacturer of this Documentation is CA.

Provided with "Restricted Rights." Use, duplication or disclosure by the United States Government is subject to the restrictions set forth in FAR Sections 12.212, 52.227-14, and 52.227-19(c)(1) - (2) and DFARS Section 252.227-7014(b)(3), as applicable, or their successors.

Copyright © 2010 CA. All rights reserved. All trademarks, trade names, service marks, and logos referenced herein belong to their respective companies.

CA Technologies Product References

This document references the following CA Technologies products:

- CA Access Control
- CA AutoSys Workload Automation Connect Option (CA AutoSys WA Connect Option)
- CA Embedded Entitlements Manager (CA EEM)
- CA Job Management Option
- CA Jobtrac™ Job Management (CA Jobtrac JM)
- CA Network and Systems Management (CA NSM)
- CA NSM Event Management
- CA NSM Management Command Center (CA NSM MCC)
- CA Scheduler® Job Management (CA Scheduler JM)
- CA Service Desk
- CA Spectrum Automation Manager (formerly named CA DCA Manager)
- CA Universal Job Management Agent (CA UJMA)
- CA Workload Automation AE (formerly named CA AutoSys Workload Automation)
- CA Workload Automation Agent for UNIX (CA WA Agent for UNIX)
- CA Workload Automation Agent for Linux (CA WA Agent for Linux)
- CA Workload Automation Agent for Windows (CA WA Agent for Windows)
- CA Workload Automation Agent for i5/OS (CA WA Agent for i5/OS)
- CA Workload Automation Agent for Application Services (CA WA Agent for Application Services)
- CA Workload Automation Agent for Web Services (CA WA Agent for Web Services)
- CA Workload Automation Agent for Databases (CA WA Agent for Databases)
- CA Workload Automation Agent for SAP (CA WA Agent for SAP)
- CA Workload Automation Agent for PeopleSoft (CA WA Agent for PeopleSoft)
- CA Workload Automation Agent for Oracle E-Business Suite (CA WA Agent for Oracle E-Business Suite)
- CA Workload Automation Agent for z/OS (CA WA Agent for z/OS)
- CA Workload Automation EE (formerly named CA ESP Workload Automation)
- CA Workload Automation SE (formerly named CA 7 Workload Automation)

- CA Workload Control Center (CA WCC)
- CA Desktop and Server Management (CA DSM)

Contact CA Technologies

Contact CA Support

For your convenience, CA Technologies provides one site where you can access the information you need for your Home Office, Small Business, and Enterprise CA Technologies products. At <http://ca.com/support>, you can access the following:

- Online and telephone contact information for technical assistance and customer services
- Information about user communities and forums
- Product and documentation downloads
- CA Support policies and guidelines
- Other helpful resources appropriate for your product

Provide Feedback

If you have comments or questions about CA Technologies product documentation, you can send a message to techpubs@ca.com.

If you would like to provide feedback about CA Technologies product documentation, complete our short customer survey, which is available on the CA Support website at <http://ca.com/docs>.

Contents

Chapter 1: Introduction	19
3-tier Architecture	19
Audience	20
Deployment	20
Application Server	21
Request/Response Protocol	21
Localization	23
Authorization	23
Using the Java SDK	24
AsApi	25
ApiRequest/ApiResponse/ApiResponseSet	26
AsException/AsRunTimeException	26
Executing a Cat1 API	27
Executing a Cat2 API	29
Impersonating a User	30
Building an Application	30
Logging	33
Filtering	34
Using the C++ SDK	37
AsApi	38
ApiRequest/ApiResponse/ApiResponseSet	39
ApiException	39
Executing a Cat1 API	40
Executing a Cat2 API	41
Impersonating a User	42
Building an Application	43
Chapter 2: Java APIs	45
AbstractResponse	45
AlarmFilterAnd	46
AlarmFilterDate	47
AlarmFilterInt	49
AlarmFilterOr	52
AlarmFilterString	53
ApiRequest	55
ApiResponse	56

ApiResponseSet	57
AsApi	58
AuthenticatedUser	63
AsBadAttributesException	63
AsBadFilterFieldsException	65
AsBadJobTypeException	67
AsConstants	68
AsErrorException	75
AsException	76
AsExternalValidationException	78
AsFieldValidationException	79
AsGeneralErrorException	81
AsInitializationException	82
AsInvalidFilterResponseAttributeException	84
AsJobConstants	85
AsJobProperties	86
AsMachineDeletionException	88
AsMessage	90
AsNoAttributesException	90
AsObjectDoesNotExistException	92
AsRuntimeException	94
AsSecurityException	95
AsTimeoutException	96
AsUnknownErrorException	98
AuthenticatedUser	99
Cat1Request	100
Cat2Request	101
ComponentMachine	102
CreateParameterItem	103
DeleteBox	107
DeleteBoxReq	109
DeleteBoxRsp	110
DeleteBoxRspSet	111
DeleteCalendarDate	112
DeleteCalendarDateReq	113
DeleteCalendarDateRsp	114
DeleteCycleCalendarDays	114
DeleteCycleCalendarDaysReq	115
DeleteCycleCalendarDaysRsp	116
DeleteEntireCalendarDate	117
DeleteEntireCalendarDateReq	117
DeleteEntireCalendarDateRsp	118

DeleteExtendedCalendarDays	119
DeleteExtendedCalendarDaysReq	119
DeleteExtendedCalendarDaysRsp	120
DeleteExternalInstance	121
DeleteExternalInstanceReq	121
DeleteExternalInstanceRsp	122
DeleteGblob	122
DeleteGblocReq	123
DeleteGblocRsp	123
DeleteJblob	124
DeleteJblobReq	125
DeleteJblobRsp	126
DeleteJob	127
DeleteJobReq	128
DeleteJobRsp	129
DeleteMachine	130
DeleteMachineReq	130
DeleteMachineRsp	133
DeleteResource	134
DeleteResourceReq	135
DeleteResourceRsp	136
EnvironmentVariableItem	137
EventRequest	140
FetchGbloc	148
FetchGblocReq	149
FetchGblocRsp	149
FetchJblob	151
FetchJblobReq	152
FetchJblobRsp	153
FilterChar	154
FilterSort	155
FinderParameterItem	156
FtpCommandItem	161
GetAdapterJobStats	165
GetAdapterJobStatsReq	165
GetAdapterJobStatsRsp	166
GetAdapterJobStatsRspSet	167
GetCalendarDays	168
GetCalendarDaysReq	169
GetCalendarDaysRsp	169
GetCalendarNames	170
GetCalendarNamesReq	170

GetCalendarNamesRsp	171
GetCalendarNamesRspSet	171
GetCycleCalendarDays	172
GetCycleCalendarDaysReq	172
GetCycleCalendarDaysRsp	173
GetExtendedCalendarDays	174
GetExtendedCalendarDaysReq	174
GetExtendedCalendarDaysRsp	175
GetExternalInstances	177
GetExternalInstancesReq	178
GetExternalInstancesRsp	179
GetGblobList	180
GetGblobListReq	181
GetGblobListRsp	182
GetGeneratedExtendedCalendarDays	183
GetGeneratedExtendedCalendarDaysReq	184
GetGeneratedExtendedCalendarDaysRsp	185
GetGeneratedExtendedCalendarDaysRspSet	186
GetGlobals	187
GetGlobalsReq	188
GetGlobalsRsp	188
GetIntCodes	189
GetIntCodesReq	190
GetIntCodesRsp	190
GetJobLog	191
GetJobLogReq	192
GetJobLogRsp	193
GetJobRunsWithFilter	194
GetJobRunsWithFilterReq	196
GetJobRunsWithFilterRsp	198
GetJobStatus	201
GetJobStatusReq	201
GetJobStatusRsp	202
GetJobsWithFilter	203
GetJobsWithFilterReq	205
GetJobsWithFilterRsp	260
JobItem	269
GetJobTypesDetail	272
GetJobTypesDetailReq	273
GetJobTypesDetailRsp	273
GetMachDefs	274
GetMachDefsReq	276

GetMachDefsRsp	276
GetMachJobs	277
GetMachJobsReq	278
GetMachJobsRsp	278
GetMachRuns	280
GetMachRunsReq	281
GetMachRunsRsp	281
GetObjectNames	283
GetObjectNamesReq	284
GetObjectNamesRsp	284
GetResourcesWithFilter	285
GetResourcesWithFilterReq	285
GetResourcesWithFilterRsp	286
GetResourcesWithFilterRspSet	288
GetResourceUsage	289
GetResourceUsageReq	290
GetResourceUsageRsp	291
GetResourceUsageRspSet	293
GetSchedulerLog	294
GetSchedulerLogReq	295
GetLogRsp	295
GetStatistics	296
GetStatisticsReq	297
GetStatisticsRsp	299
GetUniqueNames	302
GetUniqueNamesReq	303
GetUniqueNamesRsp	304
I5OthersItem	305
I5ParamsItem	310
IInsertJobReq	315
IInsertJobRsp	318
IInsertUpdateJobReq	318
InsertCalendarDate	324
InsertCalendarDateReq	325
InsertCalendarDateRsp	326
InsertCycleCalendarDays	326
InsertCycleCalendarDaysReq	327
InsertCycleCalendarDaysRsp	328
InsertExtendedCalendarDays	329
InsertExtendedCalendarDaysReq	330
InsertExtendedCalendarDaysRsp	332
InsertGblob	333

InsertGblobReq	334
InsertGblobRsp	335
InsertJblob	336
InsertJblobReq	337
InsertJblobRsp	338
InsertJob	339
InsertJobReq	340
InsertJobRsp	343
InsertRealMachine	344
InsertRealMachineReq	345
InsertRealMachineRsp	350
InsertResource	351
InsertResourceReq	352
InsertResourceRsp	353
InsertUpdateDeleteJobType	354
InsertUpdateDeleteJobTypeReq	355
InsertUpdateDeleteJobTypeRsp	356
InsertUpdateExternalInstance	357
InsertUpdateExternalInstanceReq	357
InsertUpdateExternalInstanceRsp	359
InsertUpdateJob	360
InsertUpdateJobReq	361
InsertUpdateJobRsp	366
InsertVirtualMachine	368
InsertVirtualMachineReq	369
InsertVirtualMachineRsp	369
ComponentMachine	370
IntCode	371
IOverrideJobReq	372
J2eeParameterItem	377
JmxParameterItem	381
JobFilterSort	386
JobProperty	387
JobReq	388
JobRunsFilterAnd	389
JobRunsFilterChar	390
JobRunsFilterDate	392
JobRunsFilterInt	394
JobRunsFilterOr	397
JobRunsFilterString	398
ModifyCalendarDate	401
ModifyCalendarDateReq	402

ModifyCalendarDateRsp	403
ModifyCycleCalendarDays	404
ModifyCycleCalendarDaysReq	405
ModifyCycleCalendarDaysRsp	406
ModifyExtendedCalendarDays	407
ModifyExtendedCalendarDaysReq	408
ModifyExtendedCalendarDaysRsp	410
ModifyParameterItem	411
OracleProgramDataItem	415
OverrideJob	419
OverrideJobReq	420
OverrideJobRsp	425
PingApi	425
PingApiReq	426
PingApiRsp	426
PsRunCntrlArgsItem	427
RegenerateExtendedCalendar	430
RegenerateExtendedCalendarReq	431
RegenerateExtendedCalendarRsp	431
RemoveOverride	432
RemoveOverrideReq	432
RemoveOverrideRsp	433
ResourceFilterAnd	434
ResourceFilterInt	435
ResourceFilterOr	438
ResourceFilterString	439
SapArcParmItem	441
SapExtTableItem	446
SapMailListItem	450
SapPrintParmItem	455
SapRecipientItem	460
SapStepParmItem	465
SendEvt	470
SendEvtReq	471
SendEvtRsp	471
SnmpValueItem	472
.StoredProcedureArgumentItem	476
TimeOfDay	480
Tools	481
UnsendEvt	484
UnsendEvtReq	485
UnsendEvtRsp	486

UnsendEvtRspSet	487
UpdateJob	488
UpdateJobReq	489
UpdateJobRsp	492
UpdateRealMachine	493
UpdateRealMachineReq	494
UpdateRealMachineRsp	498
UpdateResource	499
UpdateResourceReq	500
UpdateResourceRsp	501
UpdateVirtualMachine	502
UpdateVirtualMachineReq	503
UpdateVirtualMachineRsp	504
UserUpdateAlarm	505
UserUpdateAlarmReq	507
UserUpdateAlarmRsp	507
WebParameterItem	508
ZOsConditionCodeItem	511

Chapter 3: C++ APIs	517
ApiExceptionWithInfo	517
ApiRequest	517
ApiResponse	518
ApiResponseSet	519
ApiTimeout	520
AsApi	521
AsExternalValidationException	523
AsFieldValidationException	523
AsGeneralErrorException	524
AsJobProperties	524
getConstantFromKeyword	525
getKeywordFromConstant	525
getKeywordsForJobType	525
getJobTypeString	526
getJobTypeInt	526
getKeywordDataType	526
AsObjectDoesNotExistException	526
AsSecurityException	527
AsUnknownErrorException	527
Cat2StartFailure	527
CreateParameterItem	528

CredsNotAuthorized	529
DeleteBox	530
DeleteBoxReq	532
DeleteBoxRsp	535
DeleteCalendarDate	536
DeleteCalendarDateReq	537
DeleteCalendarDateRsp	538
DeleteCycleCalendarDays	539
DeleteCycleCalendarDaysReq	540
DeleteEntireCalendarDate	541
DeleteEntireCalendarDateReq	542
DeleteExtendedCalendarDays	543
DeleteExtendedCalendarDaysReq	544
DeleteExternalInstance	545
DeleteExternalInstanceReq	546
DeleteGblob	547
DeleteGblobReq	548
DeleteJblob	549
DeleteJblobReq	550
DeleteJblobRsp	551
DeleteJob	552
DeleteJobReq	553
DeleteJobRsp	555
DeleteMachine	556
DeleteMachineReq	557
DeleteMachineRsp	558
DeleteResource	559
DeleteResourceReq	560
DeleteResourceRsp	561
EnvironmentVariableItem	561
FailedToSetCreds	563
FetchGblob	564
FetchGblobReq	565
FetchGblobRsp	565
FetchJblob	567
FetchJblobReq	568
FetchJblobRsp	569
FinderParameterItem	570
GetAdapterJobStats	572
GetAdapterJobStatsReq	574
GetAdapterJobStatsRsp	575
GetAlarmsWithFilter	576

GetAlarmsWithFilterReq	579
GetAlarmsWithFilterRsp	580
GetCalendarDays	583
GetCalendarDaysReq	586
GetCalendarDaysRsp	586
GetCalendarNames	587
GetCalendarNamesReq	589
GetCalendarNamesRsp	589
GetCycleCalendarDays	590
GetCycleCalendarDaysReq	592
GetCycleCalendarDaysRsp	593
GetExtendedCalendarDays	594
GetExtendedCalendarDaysReq	595
GetExtendedCalendarDaysRsp	596
GetExternalInstances	598
GetExternalInstancesReq	600
GetExternalInstancesRsp	601
GetGblobList	603
GetGblobListReq	605
GetGblobListRsp	606
GetGeneratedExtendedCalendarDays	607
GetGeneratedExtendedCalendarDaysReq	609
GetGeneratedExtendedCalendarDaysRsp	613
GetGlobalConditionsForJob	613
GetGlobalConditionsForJobReq	615
GetGlobalConditionsForJobRsp	616
GetGlobals	618
GetGlobalsReq	620
GetGlobalsRsp	621
GetIntCodes	622
GetIntCodesReq	624
GetIntCodesRsp	625
GetJobLog	626
GetJobLogReq	627
GetJobLogRsp	628
GetJobRunsWithFilter	629
GetJobRunsWithFilterReq	632
GetJobRunsWithFilterRsp	634
GetJobStatus	640
GetJobStatusReq	641
GetJobStatusRsp	641
GetJobsWithFilter	642

GetJobsWithFilterReq	644
GetJobsWithFilterRsp	648
GetJobTypesDetail	658
GetJobTypesDetailReq	661
GetJobTypesDetailRsp	661
GetLastEvtNum	662
GetLastEvtNumReq	663
GetLastEvtNumRsp	663
GetMachDefs	665
GetMachDefsReq	667
GetMachDefsRsp	668
GetMachJobs	672
GetMachJobsReq	674
GetMachJobsRsp	675
GetMachRuns	677
GetMachRunsReq	679
GetMachRunsRsp	680
GetMonitorEvent	682
GetMonitorEventReq	683
GetMonitorEventRsp	683
GetObjectNames	687
GetObjectNamesReq	690
GetObjectNamesRsp	691
GetResourcesWithFilter	692
GetResourcesWithFilterReq	694
GetResourcesWithFilterRsp	695
GetResourceUsage	698
GetResourceUsageReq	701
GetResourceUsageRsp	703
GetSchedulerLog	707
GetSchedulerLogReq	708
GetSchedulerLogRsp	709
GetStatistics	710
GetStatisticsReq	713
GetStatisticsRsp	716
GetUniqueNames	720
GetUniqueNamesReq	722
GetUniqueNamesRsp	723
I50OthersItem	724
InsertCalendarDate	726
InsertCalendarDateReq	728
InsertCycleCalendarDays	730

InsertCycleCalendarDaysReq	731
InsertExtendedCalendarDays	732
InsertExtendedCalendarDaysReq	733
InsertGblob	736
InsertGblobReq	737
InsertJblob	738
InsertJblobReq	739
InsertJblobRsp	740
InsertJob	741
InsertJobReq	742
InsertJobRsp	745
InsertRealMachine	746
InsertRealMachineReq	747
InsertResource	756
InsertResourceReq	757
InsertResourceRsp	758
InsertUpdateDeleteJobType	759
InsertUpdateDeleteJobTypeReq	760
InsertUpdateExternalInstance	762
InsertUpdateExternalInstanceReq	763
InsertUpdateJob	766
InsertUpdateJobReq	767
InsertUpdateJobRsp	783
InsertVirtualMachine	784
InsertVirtualMachineReq	785
InsertVirtualMachineRsp	786
InternalResponseError	786
InternalUnexpected	786
JobItem	787
add	788
getAttributes	788
getAttributeName	788
getBool	788
getInt	789
getJobItem	789
getItemType	789
getString	789
getVectorOfKeywordConstants	790
set	790
set	790
set	790
setItemType	791

MissingFinalResponse	791
ModifyCycleCalendarDays	792
ModifyCycleCalendarDaysReq	793
ModifyExtendedCalendarDays	794
ModifyExtendedCalendarDaysReq	795
NoResponseAvailable	797
OverrideJob	798
OverrideJobReq	799
OverrideJobRsp	805
PingApi	806
PingApiReq	807
PingApiRsp	807
RegenerateExtendedCalendar	808
RegenerateExtendedCalendarReq	809
RemoveOverride	810
RemoveOverrideReq	811
ResponseDequeueFailure	812
SapArcParmItem	812
SapExtTableItem	815
SapPrintParmItem	817
SapRecipientItem	819
SapStepParmItem	821
SendEvt	823
SendEvtReq	826
SendEvtRsp	831
StoredProcedureArgumentItem	832
UpdateJob	834
UpdateJobReq	835
UpdateJobRsp	838
UnsendEvt	839
UnsendEvtReq	841
UnsendEvtRsp	846
UpdateRealMachine	847
UpdateRealMachineReq	848
UpdateResource	853
UpdateResourceReq	854
UpdateVirtualMachine	856
UpdateVirtualMachineReq	857
UpdateVirtualMachineRsp	858
UserUpdateAlarm	859
UserUpdateAlarmReq	860
WebParameterItem	862

ZOsConditionCodeItem..... 864

Index **867**

Chapter 1: Introduction

This section contains the following topics:

- [3-tier Architecture](#) (see page 19)
- [Audience](#) (see page 20)
- [Deployment](#) (see page 20)
- [Application Server](#) (see page 21)
- [Request/Response Protocol](#) (see page 21)
- [Localization](#) (see page 23)
- [Authorization](#) (see page 23)
- [Using the Java SDK](#) (see page 24)
- [Filtering](#) (see page 34)
- [Using the C++ SDK](#) (see page 37)

3-tier Architecture

CA Workload Automation AE r11 introduces the following 3-tier architecture:

Application Server

Acts as the communication interface between the Event Server Database and the Agent and Client utilities.

Scheduler

Interprets events and, based on job definitions, initiates actions through the Agent. This is the engine of CA Workload Automation AE.

Client and Agent

Performs tasks and sends the resulting job status back to the Scheduler.

Having the business logic isolated in the Application Server allows a thin client to connect from anywhere in the enterprise. Updates need only be applied where an Application Server is installed, and third-party vendor libraries need only be installed with the Application Server.

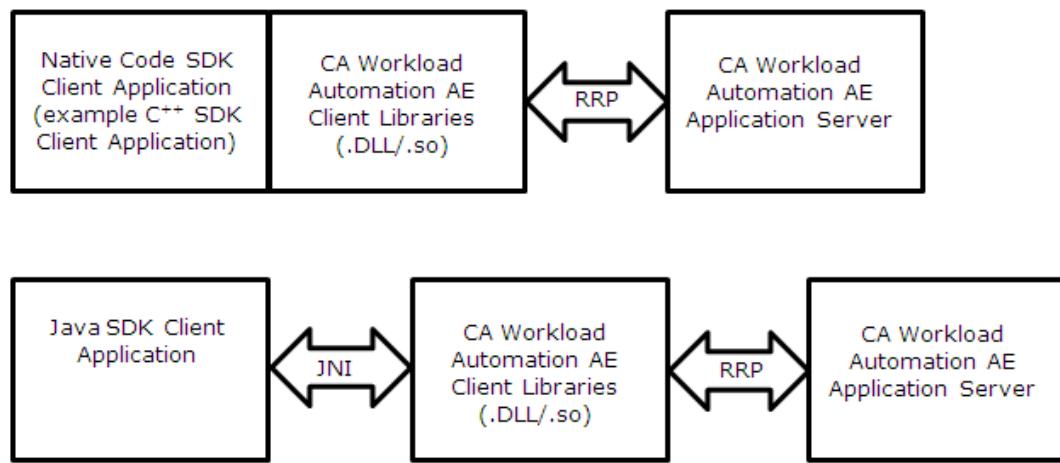
Re-architecting CA Workload Automation AE in 3 tiers necessitated refactoring of product code into easily callable units so that thin clients could access the business logic cleanly, resulting in the SDK described in this document—a ready-to-use programmatic interface to CA Workload Automation AE. Using the SDK, you can leverage the same code used by CA Workload Automation AE development to write your own applications.

Audience

This reference is intended for developers who use one or more of the SDK APIs to develop applications.

Deployment

As the following illustration shows, a Client application written in Java, C++, or native code can be run from any CA Workload Automation AE Client installation:



Native code Clients, including C++ SDK Client Applications, use normal linking procedures to interface with the CA Workload Automation AE Client libraries in a CA Workload Automation AE Client installation. Those libraries in turn use the proprietary Request/Response Protocol (RRP) to communicate with the Application Server (typically on another node).

Java Clients use the .jar file distributed with the CA Workload Automation AE installation. The .jar file uses JNI to interface with the same native code Client libraries as in the native code Client example described previously. From that point, the interaction with the Application Server is identical to the native code Client.

After a custom application is built using this SDK, the application relies on the underlying CA Workload Automation AE software to run. Therefore, the custom application must be run in a CA Workload Automation AE environment. A Client installation is sufficient for this purpose. Running in a CA Workload Automation AE environment guarantees that the necessary libraries and environment variables are available to the application.

Application Server

The CA Workload Automation AE Application Server is a highly scalable process that maintains a pool of database connections. As requests are received, the server spawns a new execution thread with a free database connection to execute the requested API. If no connections are free when a request is received, the server holds the request until a connection becomes available.

Because the server contains the CA Workload Automation AE business logic, you are isolated from all code that must be present to execute APIs, such as security libraries and database vendor libraries. Often the Application Server is installed on the same computer as the database to facilitate fast interaction with the database. The Application Server runs on any supported platform and communicates with other platforms.

Request/Response Protocol

As a 3-tier product, CA Workload Automation AE requires middleware to communicate among the tiers. Middleware must facilitate the communication between binaries across a computer network without Client applications having to worry about transport, type portability, or protocol details. CA Workload Automation AE accomplishes this with a proprietary middleware known as Request/Response Protocol (RRP).

Fundamentally, it should be possible to request anything from an Application Server with a generic, flexible, and extensible request object. That server should then be able to respond to that request with a generic, flexible, and extensible response object. Given the nature of most applications, often the returned data is manifold. Therefore, it makes sense to allow for a single request object to return multiple response objects.

We categorize these requests as follows:

Category 1 (Cat 1)

Comprises requests that expect a single response object.

Category 2 (Cat 2)

Comprises requests that expect multiple response objects. Cat 2 requests require attention to the memory requirements of large response sets. An Application Server must be able to return millions of responses without exhausting either the Client or the server computer's resources.

In a flexible request and response object a request carries with it the fundamental notion of what is being asked for. This only has meaning in the context of a particular server. Generically speaking this is a *request index*; in a specific implementation, it may correspond to the index of a function in a function table.

This is one area where RRP differs significantly from middleware like CORBA. CORBA lets you request a particular object. That object may exist anywhere, and it is the ORB's job to figure out where and connect the requestor with the implementation. In RRP, however, a request only has meaning in a particular context: a server application. The person sending the request must already know where it can be fulfilled. RRP is not as flexible, but it reduces complexity.

In addition to the request index, a request object must also contain knowledge of how to return to the requestor. This is essentially a placeholder that the server preserves by copying it to any response objects returned to the Client. The CA Workload Automation AE Client libraries set this value in a request with the knowledge that it will use that value to match a response (that will contain the value) to the initial request. Similarly, response objects must have some generic information applicable to all responses. This includes (among other things), output strings, status information, and the key used to match the response with a specific request in the Client.

Request and response objects also allow for lists of fundamental data types (for example, strings, integers, floating point values, and so on). This allows for flexibility in the existing RRP hierarchy. A large amount of response and request information can be handled by this simple mechanism. From the SDK user's perspective, this is all hidden behind wrapper classes that understand how data is arranged in these primitive Request and Response objects (RRP primitives).

To handle cat 2 requests effectively, we use a response queue that encapsulates the ability to return multiple responses without exhausting the heap. For the response queue to be effective, the Client retrieving items from the queue and the server inserting items to the queue must be asynchronous to each other. The response queue watches the size of the queue, and blocks the party inserting to the queue if it gets too full. Given that Client and server are acting asynchronously, this gives the Client time to extract items from the queue and free up room for insertions. The queue gathers the objects in the queue and sends them to the appropriate Client. The Client code retrieves the objects and puts them in the appropriate response queue (using the preserved field described previously).

RRP should not be confused with transport as RRP is middleware. RRP uses CA Secure Socket Adapter (SSA) as the transport. SSA performs port multiplexing, and by default, communication goes out over port 7163. The transmission is not encrypted. The computer where the Client process is running and the computer running the Application Server both need to have the same SSA configuration values in order to communicate, which is set up by default.

RRP is a flexible, generic solution for the movement of request and response objects between applications. The size and shape the RRP object assumes in run time varies with the run-time demand. RRP objects are designed to work between applications of very different origins with minor modification to existing applications, restricting those modifications to the point of transport.

Localization

The SDK is fully localized. Each request sent to the Application Server contains the locale of the originating Client. If the server needs to generate any string information that is suitable for presentation, it generates that string based on the specified locale. If the necessary locale-specific resource bundle is not installed on the server, it generates the strings in English.

In the native code SDK, the informational strings are accessed through `ApiResponse::info()`, `ApiResponseSet::info()` and `ApiExceptionWithInfo::what()` methods. In the Java SDK, they are in `AsMessage.getMessage()`.

Authorization

CA Workload Automation AE is a secured system, with more than one way to secure it. All APIs available in the SDK are automatically run through security checks. For example, if you attempt to retrieve a list of jobs from the API, only the set of jobs to which you have access are returned.

By default, the credentials of the user running the Client process are embedded in all requests sent to the Application Server. This should meet the security needs of the typical CA Workload Automation AE SDK Client. However, if you would like your Client application to be able to impersonate other users, the SDK can accommodate that.

To send a request on behalf of another user, you can override the default credentials of the request. However, if you do this and nothing else, the request fails before being sent to the Application Server. For an impersonating request to be released, the Client code must first authenticate that user.

A user is authenticated by calling the appropriate function. You must provide the user's ID, password, and the computer on which to verify the credentials. If the user is a Windows domain user, include the domain as part of the ID, for example, `MY_DOMAIN\my_userid`. The computer to authenticate on must be running a CA Workload Automation AE Agent. The Agent verifies that the credentials are valid. Upon successful authentication, the SDK Client libraries have those credentials marked as authenticated for the life of the Client process. That implies that subsequent requests can be sent as that user; the Client code has only to override the default credentials on a given request.

Because the Client and Application Server use a proprietary means to communicate, all requests received in the server are trusted. The decision about whether a request can be sent on behalf of another is made in the Client libraries. Configure the communication layer to employ SSL for encryption if you are concerned that this exposes the enterprise to risk.

Using the Java SDK

The base class concepts for the Java SDK are:

- AsApi
- ApiRequest
- ApiResponse
- ApiResponseSet

The Java SDK also exposes Exception classes that extend AsException and AsRunTimeException.

The Javadoc is provided on the SDK. For detailed information about the APIs, refer to the asapidoc.jar file available in the Documentation folder of your installation directory. For example, the default location of the Javadoc on Windows is as follows:

```
c:\Program Files\CA\Workload Automation AE\Documentation\asapidoc.jar
```

To expand the Javadoc, use the following command:

```
jar -xf asapidoc.jar
```

Note: The jar utility is distributed with the Java JDK.

AsApi

The AsApi class encapsulates what is in effect the connection to an Application Server, although there is no actual persistent connection maintained over the life of the object. AsApi knows how to get to the server and retrieve information from it.

Instantiating the object loads the necessary JNI libraries. For example:

```
AsApi api = new AsApi("my_server", 9000);
```

Note: A try/catch block was left out for clarity.

If the object fails, it generates an exception. In the example, the command indicates to Client libraries that all requests that use this AsApi object should be executed on the Application Server running on my_server listening on port 9000. The server must be running for a request executed against this object to be successful. A time-out results if the server has not been started.

The AsApi object is completely thread-safe and does not synchronize threads that use it. It is acceptable to create one AsApi object and have multiple threads simultaneously use it to execute APIs. It is also acceptable to create multiple AsApi objects for the same server so that each thread has its own API object.

An enterprise might have multiple servers; some for the same instance and some for other instances. An SDK Client need not be dedicated to a single server or instance. You can write an application that monitors all instances in your enterprise. Create an API object for each Application Server with which to communicate.

ApiRequest/ApiResponse/ApiResponseSet

ApiRequest, ApiResponse, and ApiResponseSet are the three classes required, in addition to the AsApi class, to execute an API. These are abstract base classes that cannot be directly instantiated. They describe and standardize the interface of all subclasses. These are concrete manifestations of the request, response, and response set described in the Request/Response Protocol section.

Each API exposed in the SDK is comprised of a set of subclasses to these base classes. For example, to retrieve global variables, there are three classes:

- GetGlobalsReq
- GetGlobalsRsp
- GetGlobalsRspSet

These classes are used solely to execute the global variable retrieval API in the server. This particular API is a cat2 API, so it includes a response set subclass.

The response set class is responsible for encapsulating the asynchronous nature of returning numerous responses to the Client. For a cat2 API, the request is executed, and a response set is produced. This response set is then iterated over by calling the hasNext() and next() methods to retrieve the set of responses.

AsException/AsRunTimeException

AsException is the base exception of all exceptions thrown by the CA Workload Automation AE Java SDK in the course of normal operation. AsRunTimeException is the base exception for those exceptions that represent unusual operation.

AsExceptions are used to signify bad return codes for requests. Therefore it is unnecessary to explicitly check for return codes. If no exception is thrown then the request completed successfully and the given response can be interrogated. Each of the AsExceptions have a method called getMessages(). This call returns an array of AsMessage objects representing all of the messages produced in response to the request. These AsMessage objects contain a parsed message key and the complete message. For example, the following could be a message sent from the Application Server:

```
CAUAJM_E_10288 *** CANNOT INSERT Job: job1, because it ALREADY EXISTS ***
```

The message key for this message is “CAUAJM_E_10288”; this code is unique for this message. With this code you can programmatically act on this message and not need to interpret the contents of the message because the message is localized appropriately by the API. The key can be retrieved from the AsMessage object with the getKey() method, and the entire message, including the key, can be retrieved with the getMessage() method. This key represents the finest grain error message that can be produced. You can deal with classes of errors by catching specific exception types. For example, all security exceptions cause an AsSecurityException to be thrown. All AsExceptions must be explicitly dealt with in the code either by a throws or a try/catch block.

AsRunTimeExceptions do not need to be explicitly dealt with in the code, but if they are not caught at some point the application will crash. These run-time exceptions are thrown in the event of unusual or unrecoverable behavior in the API. For example, if a time-out occurs an AsTimeoutException is thrown. This could mean that the server is down or maybe part of your network has become unreachable.

Executing a Cat1 API

Executing a cat1 API consists of the following steps:

- Have a pre-existing AsApi object that the request is executed against.
- Create the request object for the particular API you want to execute.
- Set the necessary fields in the request object. In the case of some APIs, there are few if any fields to set. Some requests such as InsertUpdateJobReq have a multitude of fields which can be set.
- Call the execute method on the request object, and receive the response object.
- Inspect the response object for information that was requested.

Example: Sample code for executing a Cat1 API

```
// Set the job name.  
String jobName = "sampleJob";  
  
// Set up the request.  
DeleteJobReq request = new DeleteJobReq();  
request.setRequest(jobName, false, false, false);  
  
// Execute it.  
// Note that "api" is an instance of AsApi  
// that was previously created.  
try {  
    DeleteJobRsp response = (DeleteJobRsp) request.execute(api);  
  
    if (response.hasDependentJobs())  
        System.out.println("Jobs which are dependent on the");  
  
    System.out.println("DELETED Job: " + jobName);  
  
    if (response.hasDependentJobs()) {  
        String[] dependentJobs = response.getDependentJobs();  
  
        for (int i = 0; i < dependentJobs.length; i++) {  
            System.out.println("\t" + dependentJobs[i]);  
        }  
    }  
}  
} catch (AsGeneralErrorException genEx) {  
    System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
} catch (AsSecurityException secEx) {  
    System.out.println("AsSecurityException: " + secEx.getMessage());  
} catch (AsException asEx) {  
    System.out.println("AsException: " + asEx.getMessage());  
}
```

In this case, the return information is simple. The Client code just calls `DeleteJobRsp.getDependentJobs()` to retrieve a list of jobs that are dependent on the deleted job.

The response can also contain a textual message localized and suitable for presentation to an end user. If a response contains such a message, the `ApiResponse.hasWarnings()` method returns true. The text can then be retrieved with `ApiResponse.getWarnings()` method. This string will be appropriate to the Client locale. It is returned as an array of `AsMessage` objects for convenience.

Executing a Cat2 API

A cat2 API is executed similar to cat1, but instead of getting a single response, the Client code uses a response set to retrieve multiple responses.

Example: Sample code for executing a Cat2 API

```
// setup request
GetObjectNamesReq req = new GetObjectNamesReq();
Req.setRequest("%", GetObjectNamesReq.OT_CALENDAR );

// Execute it. Note that api is an AsApi object that was
// created earlier.
IApiResponseSet rset = req.execute(api);
// Keep retrieving responses while they are available
try
{
    while (rset.hasNext()) {

        // Get the next response from the set
        GetObjectNamesRsp rsp = (GetObjectNamesRsp)rset.next();

        if ( rsp.hasWarnings() ) {
            AsMessage[] warnings = rsp.getWarnings();
            for( int i = 0; i < warnings.length; i++ ) {
                System.out.println(warnings[i].getMessage());
            }
        }
        System.out.println("name: " + rsp.getName());
    }

    if( rset.hasWarnings() ) {
        AsMessage[] warnings = rset.getWarnings();
        for( int i = 0; i < warnings.length; i++ ) {
            System.out.println(warnings[i].getMessage());
        }
    }
}
catch( AsException ex )
{
    System.out.println("GetObjectNames failed.");

    AsMessage[] errors = rsp.getMessages();
    for( int i = 0; i < errors.length; i++ ) {
        System.out.println(errors[i].getMessage());
    }
}
```

Do the same as you would for a cat1 response, but do it from within a loop. The major difference is that there will be a status and potentially an informational string contained in the response set once all responses have been retrieved. If an error status is detected, the appropriate exception is thrown. When this exception is caught, it can be interrogated for the appropriate error messages.

Note: The GetObjectNamesRspSet is not explicitly used. There is no special feature that the GetObjectNamesRspSet class contains that is not exposed by the IApiResponseSet interface. Therefore, downcasting to the GetObjectNamesRspSet is unnecessary.

Impersonating a User

You can send a request on behalf of another user ID. By default, the Client libraries deduce the user ID from the process environment, and the request is sent as that user. To override that, you must first authorize the credentials on whose behalf you are sending a request.

Example: Executing the request from the cat1 sample as user ID user_x

```
// Create AsApi object and open it.  
AsApi api = new AsApi("my_server", 9000);  
  
// Authenticate a user with the API. If the authentication is  
// successful, then that user can be used by any subsequent  
// requests simply by calling Request.setUser() (as below).  
api.authenticateWithPassword("user_x", "auth_node", "x_password");  
  
// Create the request and set the previously authenticated user  
DeleteJobReq req = new DeleteJobReq();  
req.setUser("user_x", "auth_node");  
  
// Execute it.  
DeleteJobRsp response = ( DeleteJobRsp ) request.execute( api );
```

This request is sent to the Application Server on my_server, port 9000 as if it were being sent by the authorized user. If the user does not have permission to see the requested global variable, the request fails accordingly.

Building an Application

To build an application with the Java API, include the asapi.jar and log4j-1.2.6.jar in your class path. Log4j is used internally in the Java API to provide debugging messages.

To build the sample program com.ca.autosys.sample.PrintGlobalVariable, use the following command:

```
//-----
// Copyright (c) 2006 CA. All rights reserved.
//
// This software and all information contained therein is confidential
// and proprietary and shall not be duplicated, used, disclosed or
// disseminated in any way without the express written permission of
// CA. All authorized reproductions must be marked with this language.
//
// TO THE EXTENT PERMITTED BY APPLICABLE LAW, CA PROVIDES THIS
// SOFTWARE "AS IS" WITHOUT WARRANTY OF ANY KIND, INCLUDING WITHOUT
// LIMITATION, ANY IMPLIED WARRANTIES OF MERCHANTABILITY, FITNESS FOR
// A PARTICULAR PURPOSE OR NONINFRINGEMENT. IN NO EVENT WILL CA BE
// LIABLE TO THE END USER OR ANY THIRD PARTY FOR ANY LOSS OR DAMAGE,
// DIRECT OR INDIRECT, FROM THE USE OF THIS MATERIAL, INCLUDING
// WITHOUT LIMITATION, LOST PROFITS, BUSINESS INTERRUPTION, GOODWILL,
// OR LOST DATA, EVEN IF CA IS EXPRESSLY ADVISED OF SUCH LOSS OR
// DAMAGE.
//-----
import org.apache.log4j.PropertyConfigurator;

import com.ca.autosys.services.AsApi;
import com.ca.autosys.services.AsException;
import com.ca.autosys.services.request.cat1.PingApiReq;
import com.ca.autosys.services.response.PingApiRsp;

public class Sample {

    /**
     * Constructor.
     * @param api API connection to the AutoSys application server.
     */
    public Sample(AsApi api) {
        processRequest(api);
    }

    /**
     * Process the request.
     * @param api API connection to the AutoSys application server.
     */
    private void processRequest(AsApi api) {
        PingApiReq request = new PingApiReq();

        try {
            // Execute the request, and store the response.
            PingApiRsp response = (PingApiRsp) request.execute(api);

            // Output the server version from the response.
        }
    }
}
```

```
        System.out.println("Server version is: " +
response.getServerVersion());
    } catch (AsException e) {
        System.out.println("Exception caught: " + e.getMessage());
    }
}

/**
 * Main method.
 * @param args Command line arguments.
 */
public static void main(String[] args) {
    // Configure the logger.
    PropertyConfigurator.configure("./log.properties");

    // Check for the appropriate number of command line args.
    if (args.length != 2) {
        String strUsage = "Usage: " + "java com.ca.autosys.sample.Sample
"
                     + "<server> <server_port>";
        System.out.println(strUsage);
        System.exit(0);
    } else {
        // Create connection to the AutoSys application server.
        AsApi api = new AsApi(args[0], Integer.parseInt(args[1]));

        // Create the object.
        new Sample(api);
    }
}

}
```

To run the application, use the following command:

```
C:\Temp\Sample>dir
      Volume in drive C has no label.
      Volume Serial Number is CC15-8329

      Directory of C:\Temp\Sample

07/11/2006  03:57 PM    <DIR>        .
07/11/2006  03:57 PM    <DIR>        ..
06/20/2006  10:56 AM           643 log.properties
07/11/2006  03:44 PM          3,007 Sample.java
                           2 File(s)       3,650 bytes
                           2 Dir(s)   7,664,480,256 bytes free

C:\Temp\Sample>javac -classpath
%AUTOSYS%\bin\asapi.jar;%AUTOSYS%\bin\log4j-1.2.6.jar Sample.java

C:\Temp\Sample>java -cp
.;%AUTOSYS%\bin\asapi.jar;%AUTOSYS%\bin\log4j-1.2.6.jar Sample
autosys-server.ca.com 9000
      Server version is: 1

C:\Temp\Sample>
```

Logging

The API does not set up a logger automatically. This step is left to the user of the API. Note that in the examples this logger is set up with the following command.

```
// Configure log4j
PropertyConfigurator.configure("./log.properties");
```

This command creates a root logger based on the configuration found in the file log.properties. See the log4j website for information on how to configure log4j, <http://logging.apache.org/log4j/docs/>.

If debugging is turned on for the API, output detailing the requests and responses processed by the API are detailed. The following is debug output generated from the sample program.

```
2006-07-13 11:46:53,091 com.ca.autosys.services.request.cat1.PingApiReq. [set the
init variable for your book] - Request Details
===== Request Contents =====
Request ID: [316]
--General Parameters--
Request Set: [true]
User: [null]
Machine: [null]
=====
2006-07-13 11:46:53,091
com.ca.autosys.services.request.cat1.Cat1Request.execute - Executing request: class
com.ca.autosys.services.request.cat1.PingApiReq
2006-07-13 11:46:53,153
com.ca.autosys.services.response.PingApiRsp. [set the init variable for your book] -
Response Details
$$$$ Response Contents $$$$
Server Version: [1]
--General Contents--
Return Code: [0]
Status Code: [2]
Has Info: [false]
$$$$$$$$$$$$$$$$$$$$$$$$$$$$$$
2006-07-13 11:46:53,153
com.ca.autosys.services.response.AbstractResponse.interpretBasicReturnCode -
API_SUCCESS found.
```

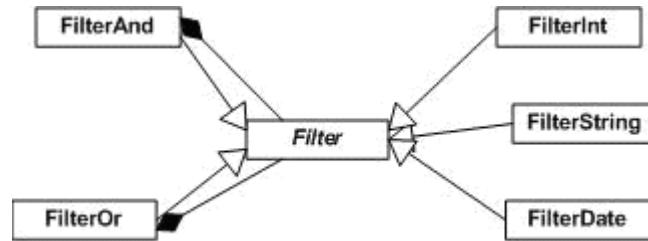
In these logs, the initial request is made to retrieve the server version. This request is executed, and a response is received that contains the server version of 1.

Filtering

The Filtering APIs are a special class of APIs that let you retrieve CA Workload Automation AE objects in a flexible way. While they are similar to all other APIs in the CA Workload Automation AE SDK in that they are comprised of requests, responses, and response sets, what goes together to make the request object is slightly more complicated and deserves additional discussion.

Given a particular CA Workload Automation AE object—currently just alarms, jobs, and jobruns are supported—it is possible to specify a list of attributes to retrieve for each object and to specify matching criteria against those objects in order to retrieve only a subset of all objects in the Event Server. The retrieval criteria are embedded in the request by way of a filter object.

To allow for the ability to logically connect these objects and nested groups of these objects, the filter classes follow the Composite Pattern. The following illustration is a simplified UML diagram to give a sense of how the various filtering classes fit together:



Note: Many classes have not been included in the diagram, and the names have been changed in favor of generic ones for the sake of discussion.

In this diagram, **FilterAnd** and **FilterOr** are composite classes. **FilterInt**, **FilterString**, and **FilterDate** are leaf classes. The composite classes are used to logically connect leaf or other composite classes. The leaf classes each know how to compare or filter against a particular object attribute type (integers, strings, and dates).

A Client program builds up a filter object made out of these classes and places that in the request. The Application Server returns the objects that match it.

Because the filtering APIs can be used to retrieve a large number of objects, they also have the capacity to specify what fields are returned. This lets you retrieve only the fields needed thereby reducing the amount of data being transferred and time to transfer.

The following example demonstrates how to execute a filter API. This example executes the `GetJobsWithFilter` API. It asks for only five job fields and only wants jobs that are named `job%` (where `%` is a wildcard) and have a status of `SUCCESS`.

Example: Executing a filter API

```
// Retrieve the following properties
int[] attributes = new int[5];
attributes[0] = GetJobsWithFilterReq.ATTR_BOX_NAME;
attributes[1] = GetJobsWithFilterReq.ATTR_COMMAND;
attributes[2] = GetJobsWithFilterReq.ATTR_JOB_NAME;
attributes[3] = GetJobsWithFilterReq.ATTR_JOB_TYPE;
attributes[4] = GetJobsWithFilterReq.ATTR_MACHINE;

// Create a filter that will match all jobs named job% that are in
// SUCCESS status.
JobFilterAnd filter = new JobFilterAnd();
filter.add(new JobFilterString(JobFilterStringFLT_JOB_NAME, "job%"));
filter.add(new JobFilterInt(JobFilterIntFLT_STATUS, 4, JobFilterInt.EQUAL));

//Create the request object.
GetJobsWithFilterReq request = new GetJobsWithFilterReq();

//Set the filter and attributes that describe our specific request.
request.setRequest(filter, attributes);

//Execute this request on the AutoSys R11 server and examine the
//response set.
//
// myApi: created previously
//
IApiResponseSet jobs = request.execute(myApi);
while(jobs.hasNext())
{
    IGetJobsWithFilterRsp current = (IGetJobsWithFilterRsp)jobs.next();
    int[] attributes = current.getAttributes();

    for( int i = 0; i < attributes.length; i++ )
    {
        int currentAttribute = attributes[i];
        String attributeName = current.getAttributeName(currentAttribute);
        String attributeValue = null;

        switch( current.getAttributeType(currentAttribute) )
        {
            case GetJobsWithFilterRsp.TYPE_BOOLEAN:
                boolean value = current.getBoolean(currentAttribute);
                // ... do something with value ...
                break;

            case GetJobsWithFilterRsp.TYPE_INT:
                attributeValue = current.getInt(currentAttribute) + "";
                // ... do something with attributeValue ...
                break;
        }
    }
}
```

```
        case GetJobsWithFilterRsp.TYPE_STRING:  
            attributeValue = current.getString(currentAttribute);  
            // ... do something with attributeValue ...  
            break;  
        }  
    }  
}
```

Note: When the response is returned, it contains the metadata that allows the response to be handled by code that did not originally generate the request. That is, the Client code does not necessarily need to know ahead of time what is included in the response. It can interrogate the response to determine that.

Using the C++ SDK

The base class concepts for the C++ SDK are:

- AsApi
- ApiRequest
- ApiResponse
- ApiResponseSet
- ApiException

The HTML documentation is provided on the SDK. For detailed information about the APIs, refer to the asapicppdoc.jar file available in the Documentation folder of your installation directory. For example, the default location of the HTML documentation on Windows is as follows:

```
c:\Program Files\CA\Workload Automation AE\Documentation\asapicppdoc.jar
```

To expand the HTML documentation, use the following command:

```
jar -xf asapicppdoc.jar
```

Note: The jar utility is distributed with the Java JDK.

AsApi

The AsApi class encapsulates what is in effect the connection to an Application Server, although there is no actual persistent connection maintained over the life of the object. AsApi knows how to get to the server and retrieve information from it.

Examples: Opening a connection to an Application Server

- Open a connection to a local Application Server:

```
AsApi api;  
api.openLocal("app_name");
```

- Open a connection to a remote Application Server:

```
AsApi api;  
api.openRemote(9000, "my_server");
```

If the object fails, it generates an exception. In the examples, the command indicates to Client libraries that all requests that use this AsApi object should be executed on the Application Server running on my_server listening on port 9000. The server must be running for a request executed against this object to be successful. A time-out results if the server has not been started.

The AsApi object is completely thread-safe and does not synchronize threads that use it. It is acceptable to create one AsApi object and have multiple threads simultaneously use it to execute APIs. It is also acceptable to create multiple AsApi objects for the same server so that each thread has its own API object.

An enterprise might have multiple servers; some for the same instance and some for other instances. An SDK Client need not be dedicated to a single server or instance. You can write an application that monitors all instances in your enterprise. Create an API object for each Application Server with which to communicate.

ApiRequest/ApiResponse/ApiResponseSet

ApiRequest, ApiResponse, and ApiResponseSet are the three classes required, in addition to the AsApi class, to execute an API. These are abstract base classes that cannot be directly instantiated. They describe and standardize the interface of all subclasses. These are concrete manifestations of the request, response, and response set described in the Request/Response Protocol section.

Each API exposed in the SDK is comprised of a set of subclasses to these base classes. For example, to retrieve global variables, there are three classes:

- GetGlobalsReq
- GetGlobalsRsp
- GetGlobalsRspSet

These classes are used solely to execute the global variable retrieval API in the server. This particular API is a cat2 API, so it includes a response set subclass.

The response set class is responsible for encapsulating the asynchronous nature of returning numerous responses to the Client. For a cat2 API, the request is executed, and a response set is produced. This response set is then iterated over by calling the hasNext() and next() methods to retrieve the set of responses.

ApiException

ApiException is the base exception of all exceptions thrown by the CA Workload Automation AE C++ SDK in the course of normal operation.

ApiException is used to signify bad return codes for requests. Therefore it is unnecessary to explicitly check for return codes. If no exception is thrown then the request completed successfully and the given response can be interrogated. Each of the ApiException has a method called what(). This call returns a message produced in response to the request.

The ApiExceptionWithInfo object contains a message key and the complete message. For example, the following could be a message sent from the Application Server:

CAUAJM_E_10288 *** CANNOT INSERT Job: job1, because it ALREADY EXISTS ***

The message key for this message is CAUAJM_E_10288; this code is unique for this message. With this code you can programmatically act on this message and not need to interpret the contents of the message because the message is localized appropriately by the API.

Executing a Cat1 API

Executing a cat1 API consists of the following steps:

- Have a pre-existing AsApi object that the request is executed against.
- Create the request object for the particular API you want to execute. For example, to delete a command job, create DeleteJobReq and DeleteJobRsp objects.
- Set the necessary fields in the request object. In the case of some APIs, there are few if any fields to set. Some requests such as InsertUpdateJobReq have a multitude of fields which can be set.
- Call the execute method on the request object, and receive the response object.
- Inspect the response object for information that was requested.

Example: Sample code for executing a cat1 API

```
try {  
    DeleteJobReq req;  
    req.setName("sampleJob");  
    DeleteJobRsp rsp;  
    req.execute(api, rsp);  
}  
catch (ApiExceptionWithInfo& e) {  
    std::cerr << "ApiExceptionWithInfo: " << e.what() << std::endl;  
}
```

Executing a Cat2 API

A cat2 API is executed similar to cat1, but instead of getting a single response, the Client code uses a response set to retrieve multiple responses.

Example: Retrieving multiple responses

```
AsApi api;

try {
    api.openRemote(9000, "myserver");
    // setup request
    GetObjectNamesReq req;
    req.setNamePattern("%");

    // Execute it.
    GetObjectNamesRspSet rset;
    req.execute(api), rset);

    // Keep retrieving responses while they are available
    while (rset.hasMore()) {

        // Get the next response from the set
        GetObjectNamesRsp rsp;
        rset.next(rsp);
        if(rsp.hasInfo())
            std::cout << rsp.info() << std::endl;
        std::cout << "Object name: " << rsp.getObjectName() << std::endl;
    }
}
catch (ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo: " << e.what() << std::endl;
}
```

Do the same as you would for a cat1 response, but do it from within a loop. The major difference is that there will be a status and potentially an informational string contained in the response set once all responses have been retrieved.

Impersonating a User

You can send a request on behalf of another user ID. By default, the Client libraries deduce the user ID from the process environment, and the request is sent as that user. To override that, you must first authorize the credentials on whose behalf you are sending a request.

Example: Executing the request from the cat1 sample as user ID user_x

```
// Create AsApi object and open it.  
AsApi api;  
api.openRemote(9000, "my_server");  
  
// Authenticate a user with the API. If the authentication is  
// successful, then that user can be used by any subsequent  
// requests simply by calling Request.setUser() (as below).  
  
api.authenticateWithPassword("user_x", "auth_node", "x_password");  
  
// Create the request and set the previously authenticated user  
GetGlobalsReq req;  
req.setUser("user_x", "auth_node");  
req.setNamePattern("TODAY");  
  
GetGlobalsRspSet rset;  
req.execute(api, rset);  
while (rset.hasMore()) {  
    GetGlobalsRsp rsp;  
    rset.next(rsp);  
    std::cout << "Name: " << rsp.name() << std::endl;  
}
```

This request is sent to the Application Server on my_server, port 9000 as if it were being sent by the authorized user. If the user does not have permission to see the requested global variable, the request fails accordingly.

Building an Application

To build a Windows application with the C++ API, include the header file, AsPublicSDK.h, and the libas_sdk.lib library to your project.

The AsPublicSDK.h file, libs and a sample project (AutoSysApiSample.vcproj) are included at the following location within the CA Workload Automation AE installation:

- Windows
%AUTOROOT%\autosys\code
- UNIX
\$AUTOSYS/code

Chapter 2: Java APIs

This chapter provides an alphabetical reference to each of the Java Application Programming Interfaces (APIs) that make up the CA Workload Automation AE Software Development Kit (SDK).

AbstractResponse

extends

`java.lang.Object`

AbstractResponse provides common methods for responses.

interpretReturnCode

The interpretReturnCode method in the AbstractResponse class converts status codes and return codes to exceptions. This method has the following syntax:

```
void interpretReturnCode() throws AsException  
AsSecurityException  
AsGeneralErrorException  
AsException
```

getWarnings

The getWarnings method in the AbstractResponse class has the following syntax:

```
AsMessage[] getWarnings()
```

hasWarnings

The hasWarnings method in the AbstractResponse class has the following syntax:

```
boolean hasWarnings()
```

AlarmFilterAnd

extends

com.ca.autosys.services.request.filter.Filter

implements

IAlarmFilterComposite, IAndComposite

AlarmFilterAnd is an AlarmFilter container that connects the added filters with a logical AND.

AlarmFilterAnd

The AlarmFilterAnd construct has the following syntax:

`public AlarmFilterAnd()`

add

The add method in the AlarmFilterAnd class logically connects a given filter to this filter with AND. This method has the following syntax:

`public void add(IAlarmFilter filter)`

filter

Specifies the filter to AND.

getComponents

The getComponents method in the AlarmFilterAnd class returns a list of cached leaf filter components. This method has the following syntax:

`public java.util.ArrayList getComponents()`

toString

The toString method in the AlarmFilterAnd class overrides toString in the java.lang.Object class. This method has the following syntax:

`public java.lang.String toString()`

AlarmFilterDate

extends

com.ca.autosys.services.request.filter.Filter

implements

IAlarmFilterDate

AlarmFilterDate is an AlarmFilter leaf that filters by dates.

AlarmFilterDate

The AlarmFilterDate constructor is a convenience constructor that builds a filter based on the contents of a given filter. This constructor has the following syntax:

public AlarmFilterDate(IAlarmFilterDate *filter*)

filter

Defines the object that implements the IJobFilterDate interface.

AlarmFilterDate

The AlarmFilterDate constructor creates an AlarmDateFilter that matches alarms that have a time between the specified begin and end dates. This constructor has the following syntax:

public AlarmFilterDate(int *id*, java.util.Date *date*, int *operator*)

id

Specifies the date property ID as one of the following:

- FLT_ALARM_TIME
- FLT_START_TIME

date

Specifies the date to filter by.

operator

Specifies one of the following operators to apply to the date filter:

- EQUAL
- GREATER
- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

getDate

The getDate method in the AlarmFilterDate class returns the date. This method has the following syntax:

```
public java.util.Date getDate()
```

getFilterId

The getFilterId method in the AlarmFilterDate class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getOperator

The getOperator method in the AlarmFilterDate class returns the operator. This method has the following syntax:

```
public int getOperator()
```

toString

The toString method in the AlarmFilterDate class overrides toString in the java.lang.Object class. This method has the following syntax:

```
public java.lang.String toString()
```

AlarmFilterInt

extends

com.ca.autosys.services.request.filter.Filter

implements

IAlarmFilterInt

AlarmFilterInt is an AlarmFilter leaf that filters on integers.

AlarmFilterInt

This AlarmFilterInt constructor creates an AlarmFilterInt that will filter integer properties according to a given operator. This constructor has the following syntax:

`public AlarmFilterInt(int id, int value, int operator)`

id

Defines the ID of the int property as one of the following:

- FLT_ALARM_TYPE
- FLT_STATE
 - AsConstants.ALARM_OPEN
 - AsConstants.ALARM_CLOSED
 - AsConstants.ALARM_ACKNOWLEDGED

value

Specifies the integer to filter against.

operator

Specifies one of the following operators to apply to the integer value:

- EQUAL
- GREATER
- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

The AlarmFilterInt constructor throws the java.lang.IllegalArgumentException if an invalid operator is used.

AlarmFilterInt

This AlarmFilterInt constructor creates an AlarmFilterInt that will filter integer properties in an array according to a given operator. This constructor has the following syntax:

```
public AlarmFilterInt(int id, int[] values, int operator)
```

id

Defines the ID of the int property as one of the following:

- FLT_ALARM_TYPE
- FLT_STATE
 - AsConstants.ALARM_OPEN
 - AsConstants.ALARM_CLOSED
 - AsConstants.ALARM_ACKNOWLEDGED

values

Specifies the integer array of values to filter against.

operator

Specifies one of the following operators to apply to the integer value:

- EQUAL
- NOT_EQUAL

The AlarmFilterInt constructor throws the java.lang.IllegalArgumentException if an operator other than EQUAL or NOT_EQUAL is used.

AlarmFilterInt

This AlarmFilterInt constructor is a convenience constructor that builds a filter based on the contents of the given filter. This constructor has the following syntax:

```
public AlarmFilterInt(IAlmFilterInt filter)
```

filter

Defines an object that implements the IAlmFilterInt interface.

toString

The `toString` method in the `AlarmFilter Int` class overrides `toString` in the `java.lang.Object` class. This method has the following syntax:

```
public java.lang.String toString()
```

getFilterId

The `getFilterId` method in the `AlarmFilterInt` class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getOperator

The `getOperator` method in the `AlarmFilterInt` class returns the operator. This method has the following syntax:

```
public int getOperator()
```

getValue

The `getValue` method in the `AlarmFilterInt` class returns the value. This method has the following syntax:

```
public int getValue()
```

getValues

The `getValues` method in the `AlarmFilterInt` class returns the set of values. This method has the following syntax:

```
public int[] getValues()
```

AlarmFilterOr

extends

com.ca.autosys.services.request.filter.Filter

implements

IAlarmFilterComposite, IOrCompositeAn

The AlarmFilter class is a container that connects the added filters with a logical OR.

AlarmFilterOr

The AlarmFilterOr constructor has the following syntax:

```
public AlarmFilterOr()
```

add

The add method in the AlarmFilterOr class logically connects a given filter to this filter with OR. This method has the following syntax:

```
public void add(IAlarmFilter filter)
```

filter

Defines the filter to OR.

toString

The toString method in the AlarmFilterOr class overrides toString in the java.lang.Object class. This method has the following syntax:

```
public java.lang.String toString()
```

getComponents

The getComponents method in the AlarmFilterOr class returns a list of cached leaf filter components. This method has the following syntax:

```
public java.util.ArrayList getComponents()
```

AlarmFilterString

extends

com.ca.autosys.services.request.filter.Filter

implements

IAlarmFilterString

The AlarmFilterString class is an AlarmFilter leaf that filters on a string.

AlarmFilterString

The AlarmFilterString constructor creates an AlarmFilterString that will match alarms that have the given string value. This constructor has the following syntax:

`public AlarmFilterString(int id, java.lang.String value, int operator)`

id

Defines one of the following IDs for the String property:

- FLT_EOID
- FLT_JOB_NAME
- FLT_MOD_USER

value

Specifies the string value to match on. This parameter can contain wild cards.

operator

Defines one of the following operators to apply to this String:

- LIKE
- NOT_LIKE

AlarmFilterString

The AlarmFilterString convenience constructor creates an alarm filter string that match alarms with a given String value. This constructor has the following syntax:

```
public AlarmFilterString(int id, java.lang.String value)
```

id

Defines one of the following IDs for the String property:

- FLT_EOID
- FLT_JOB_NAME
- FLT_MOD_USER

value

Specifies the string value to match on. This parameter can contain wild cards.

AlarmFilterString

The AlarmFilterString convenience constructor builds a filter based on the contents of the given filter. This constructor has the following syntax:

```
public AlarmFilterString(IAlarmFilterString filter)
```

filter

Defines an object that implements the IAlarmFilterString interface.

toString

The `toString` method in the `AlarmFilterString` class overrides `toString` in class `java.lang.Object`. This method has the following syntax:

```
public java.lang.String toString()
```

getFilterId

The `getFilterId` method in the `AlarmFilterString` class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getOperator

The `getOperator` method in the `AlarmFilterString` class returns the operator. This method has the following syntax:

```
public int getOperator()
```

getValue

The `getValue` method in the `AlarmFilterString` class returns the value. This method has the following syntax:

```
public java.lang.String getValue()
```

ApiRequest

extends

```
java.lang.Object
```

`ApiRequest` is the base class for functionality common to all API requests. It is not used directly as it is not associated with a particular API. Only its subclasses that are associated with APIs are used.

setUser

The `setUser` method in the `ApiRequest` class defines the user in this request. This method has the following syntax:

```
abstract void setUser(java.lang.String userName, java.lang.String machine)  
userName
```

Sets the user on whose behalf this request is executed. The `setUser` method requires a prior call to `AsApi.authenticateWithPassword`.

machine

Sets the computer on which this user is defined.

toString

The `toString` method in the `ApiRequest` class returns a multiple line string that details the values set to remember for the current object. This method has the following syntax:

```
java.lang.String toString()
```

setLocale

The `setLocale` method in the `ApiRequest` class sets the locale of the request. Strings sent back in the response will be created for the correct locale. This method has the following syntax:

```
public void setLocale(java.util.Locale locale)
```

locale

Defines the locale the server must use for all returned information.

ApiResponse

extends
AbstractResponse
implements
IApiResponse

ApiResponse is the base class for functionality common to all responses.

ApiResponse

The ApiResponse constructor creates an ApiResponse object. This constructor has the following syntax:

```
public ApiResponse()
```

getDateFromString

The getDateFromString method in the ApiResponse class extracts the date from the specified string. This method returns a date object or (if the time_str value is not in MM/dd/yyyy HH:mm format) null. This method has the following syntax:

```
static java.util.Date getDateFromString(java.lang.String time_str)
```

time_str

Defines the string that contains the date and time, in MM/dd/yyyy HH:mm format.

getStringFromDate

The getStringFromDate method in the ApiResponse class converts a date object to a string in the format MM/dd/yyyy HH:mm. This method has the following syntax:

```
static java.lang.String getStringFromDate(java.util.Date date)
```

date

Defines the date object to convert to a string in MM/dd/yyyy HH:mm format.

toString

The toString method in the ApiResponse class returns a multiple line string that details the values set to remember for the current object. The toString method in the ApiResponse class overrides toString in class java.lang.Object. This method has the following syntax:

```
java.lang.String toString()
```

getServerVersion

The `getServerVersion` method in the `ApiResponse` class returns the CA Workload Automation AE server version. This method has the following syntax:

```
public int getServerVersion()
```

ApiResponseSet

extends

AbstractResponse

implements

IApiResponseSet

`ApiResponseSet` is responsible for making multiple responses available from APIs that return more than one response.

ApiResponseSet

The `ApiResponseSet` constructor creates a `ApiResponseSet` object. This constructor has the following syntax:

```
public ApiResponseSet()
```

hasNext

The `hasNext` method in the `ApiResponseSet` class returns whether the iteration has more elements. To determine if more elements exist, an element must actually be taken off of the queue. Because of this required dequeue operation, exceptions can occur. It returns true if the iterator has more elements. See the specific response set class documentation for possible recoverable exceptions that can be caught. This method has the following syntax:

```
boolean hasNext()
```

`AsException`, the super class of all API specific return code exceptions, can be thrown. See the specific response types for more information on possible exceptions.

next

The `next` method in the `ApiResponseSet` class returns the next element in the iteration. This method has the following syntax:

```
IApiResponse next()
```

`AsException`, the super class of all API specific return code exceptions, can be thrown. See the specific response types for more information on possible exceptions.

AsApi

```
extends  
    java.lang.Object  
implements  
    IAsApi
```

AsApi is the connection to the CA Workload Automation AE Application Server. There is no restriction on the ability to create multiple API objects. AsApi provides the ability to control the overhead imposed by the API. If you only have limited interaction with the API, create the object only when it is needed and then discard it. For an application that interacts constantly with the API, create the object once at startup.

Requests can be issued on behalf of other users. By default, a request is executed on behalf of the user that is currently running the process (as identified by the OS). However, if the code calls `ApiRequest.setUser()`, it causes the request to be done on behalf of the specified user. This only works if the user has previously called `AsApi.authenticateWithPassword()` for the given user.

AsApi

This AsApi constructor creates a thin client API object that targets the given host and port. This constructor has been deprecated and has the following signature:

```
AsApi(java.lang.String host, int port, boolean haMode)
```

host

Defines the CA Workload Automation AE server host.

port

Defines the CA Workload Automation AE server port.

haMode

Specifies to construct a high-availability AsApi object. It will retrieve the list of possible servers from the passed in host/port and will automatically rollover to another application server if one becomes unavailable.

AsApi

This AsApi constructor creates a thin client API object that targets the given host and port, and is highly-available. This constructor has been deprecated and has the following signature:

```
AsApi(java.lang.String host, int port)
```

host

Defines the CA Workload Automation AE server host.

port

Defines the CA Workload Automation AE server port.

AsApi

This AsApi constructor creates a thin client API object that targets the given host and port using a specified encryption method and key. This API object is highly-available. This constructor has the following syntax:

```
public AsApi(java.lang.String host, int port, int crypt_type, java.lang.String crypt_key)
```

host

Specifies the CA Workload Automation AE server host.

port

Specifies the CA Workload Automation AE server port.

crypt_type

Specifies one of the following encryption methods:

- AsConstants.ENCRYPTION_TYPE_AES
- AsConstants.ENCRYPTION_TYPE_CONFIG
- AsConstants.ENCRYPTION_TYPE_DEFAULT
- AsConstants.ENCRYPTION_TYPE_NONE

crypt_key

Specifies the 32-hexadecimal digit or maximum 16-character pass-phrase used to generate the encryption key. This parameter is Ignored for non-AES encryption methods.

AsApi

This AsApi constructor creates a thin client API object that targets the given host and port using a specified encryption method. This API object is highly-available. This constructor has the following syntax:

```
public AsApi(java.lang.String host, int port, int crypt_type)
```

host

Specifies the CA Workload Automation AE server host

port

Specifies the CA Workload Automation AE server port.

crypt_type

Specifies one of the following encryption methods:

- AsConstants.ENCRYPTION_TYPE_AES
- AsConstants.ENCRYPTION_TYPE_CONFIG
- AsConstants.ENCRYPTION_TYPE_DEFAULT
- AsConstants.ENCRYPTION_TYPE_NONE

authenticateWithArtifact

The authenticateWithArtifact method in the AsApi class authenticates an eIAM artifact and returns the AuthenticatedUser object that contains the user and node. This method has the following syntax:

```
public AuthenticatedUser authenticateWithArtifact(java.lang.String artifact)
```

artifact

Specifies the artifact string to authenticate.

A AsSecurityException can be thrown.

authenticateWithPassword

The authenticateWithPassword method in the AsApi class authenticates a user against CA Workload Automation AE security. This method has the following syntax:

```
public boolean authenticateWithPassword(java.lang.String userName,  
java.lang.String node, java.lang.String password)
```

userName

Specifies the fully qualified user name.

node

Specifies the CA Workload Automation AE node on which to authenticate the user.

password

Specifies the user password.

authenticateAsInternal

The authenticateAsInternal method in the AsApi class authenticates a user against internal CA Workload Automation AE security, and returns an AuthenticatedUser object that contains the user and node. This method of authentication is only to be used internally between CA Technologies products. This method has the following syntax:

```
public AuthenticatedUser authenticateAsInternal(java.lang.String userName,  
java.lang.String node, java.lang.String userPassword)
```

userName

Defines the user name to authenticate.

node

Specifies the CA Workload Automation AE node on which to authenticate the user.

userPassword

Defines the password to authenticate with.

The following exceptions can be thrown if authentication fails:

- AsSecurityException—when validation fails.
- AsErrorException—when there is an internal error or system failure.
- AsTimeoutException—when the connection times out before response is received.

doRequest

The doRequest method in the AsApi class executes the given Cat1 request. This method has the following syntax:

```
void doRequest(com.ca.autosys.services.JRequest req,  
com.ca.autosys.services.JResponse rsp)
```

req

Defines the request.

rsp

Defines the response.

An AsException can be thrown.

doRequest

The doRequest method in the AsApi class executes the given Cat2 request. This method has the following syntax:

```
void doRequest(com.ca.autosys.services.JRequest req,  
com.ca.autosys.services.JResponseQ rq)
```

req

Defines the request.

rq

Defines the response queue to populate.

finalize

The finalize method in the AsApi class overrides finalize in class java.lang.Object. This method has the following syntax:

```
public void finalize()
```

AuthenticatedUser

AuthenticatedUser is the object returned from the AsApi.authenticateWithArtifact method. The object contains the user and node that were authenticated with the artifact that was passed in eTrust IAM.

extends

java.lang.Object

implements

IAuthenticatedUser

getUser

The getUser method in the AuthenticatedUser class returns the user that was authenticated with the artifact. This method has the following syntax:

```
public java.lang.String getUser()
```

getNode

The getNode method in the AuthenticateUser class returns the node that was authenticated with the artifact. This method has the following syntax:

```
public java.lang.String getNode()
```

AsBadAttributesException

extends

AsException

AsBadAttributesException is thrown to relay bad attribute messages from the CA Workload Automation AE API to the Client applications.

AsBadAttributesException

The default AsBadAttributesException constructor has the following syntax:

```
public AsBadAttributesException()
```

The cause and message are not set for this constructor.

AsBadAttributesException

This AsBadAttributesException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsBadAttributesException(java.lang.String message)
```

message

Defines the message.

AsBadAttributesException

This AsBadAttributesException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsBadAttributesException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsBadAttributesException

The AsBadAttributesException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsBadAttributesException(java.lang.String message, java.lang.Throwable  
                                cause)
```

message

Defines the detailed message. The message is retrievable by calling the `Throwable.getMessage()` method.

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsBadAttributesException

The AsBadAttributesException constructor creates a new exception with the specified detailed message and list of bad attributes. This constructor has the following syntax:

```
public AsBadAttributesException(java.lang.String message, int[] badAttributes)
```

message

Defines the detailed message.

badAttributes

Defines the list of bad attributes.

getBadAttributes

The `getBadAttributes` method in the `AsBadAttributesException` class returns the bad attributes specified in the initial request. This method has the following syntax:

```
int[] getBadAttributes()
```

AsBadFilterFieldsException

extends

`AsException`

`AsBadFilterFieldsException` is thrown to relay bad filter field messages from the CA Workload Automation AE API to the client applications.

AsBadFilterFieldsException

This `AsBadFilterFieldsException` constructor is a default constructor. The cause and message are not set. This constructor has the following syntax:

```
public AsBadFilterFieldsException()
```

AsBadFilterFieldsException

This `AsBadFilterFieldsException` constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsBadFilterFieldsException(java.lang.String message)
```

message

Defines the message.

AsBadFilterFieldsException

This `AsBadFilterFieldsException` constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsBadFilterFieldsException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsBadFilterFieldsException

This AsBadFilterFieldsException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsBadFilterFieldsException(java.lang.String message, java.lang.Throwable  
cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsBadFilterFieldsException

This AsBadFilterFieldsException constructor creates a new exception with the specified detailed message and list of bad attributes. This constructor has the following syntax:

```
public AsBadFilterFieldsException(java.lang.String message, int[] badFilterFields)
```

message

Defines the message.

badAttributes

Defines the list of bad attributes.

getBadFilterFields

The getBadFilterFields method in the AsBadFilterFieldsException class returns the bad filter fields specified in the initial request. This method has the following syntax:

```
public int[] getBadFilterFields()
```

AsBadJobTypeException

extends

AsException

AsBadJobTypeException is thrown to relay bad job type messages from the CA Workload Automation AE API to the Client applications.

AsBadJobTypeException

This AsBadJobTypeException constructor is a default constructor. The cause and message are not set. This constructor has the following syntax:

```
public AsBadJobTypeException()
```

AsBadJobTypeException

This AsBadJobTypeException constructor constructs a new exception with the specified detail message. This constructor has the following syntax:

```
public AsBadJobTypeException(java.lang.String message)
```

message

Specifies the message.

AsBadJobTypeException

This AsBadJobTypeException constructor constructs a new exception with the specified detail message and cause. This constructor has the following syntax:

```
public AsBadJobTypeException(java.lang.Throwable cause)
```

cause

Specifies the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsBadJobTypeException

This AsBadJobTypeException constructor constructs a new exception with the specified detail message and cause. This constructor has the following syntax:

```
public AsBadJobTypeException(java.lang.String message, java.lang.Throwable cause)  
message
```

Specifies the detail message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Specifies the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsConstants

extends

```
java.lang.Object
```

The list of CA Workload Automation AE constants is as follows:

ALARM_OPEN

This constant has the following syntax:

```
public static final int ALARM_OPEN
```

ALARM_ACKNOWLEDGED

This constant has the following syntax:

```
public static final int ALARM_ACKNOWLEDGED
```

ALARM_CLOSED

This constant has the following syntax:

```
public static final int ALARM_CLOSED
```

JOB_TYPE_BOX

This constant has the following syntax:

```
public static final char JOB_TYPE_BOX
```

JOB_TYPE_COMMAND

This constant has the following syntax:

```
public static final char JOB_TYPE_COMMAND
```

JOB_TYPE_FILE_WATCHER

This constant has the following syntax:

```
public static final char JOB_TYPE_FILE_WATCHER
```

JOB_TYPE_USER_0

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_0
```

JOB_TYPE_USER_1

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_1
```

JOB_TYPE_USER_2

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_2
```

JOB_TYPE_USER_3

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_3
```

JOB_TYPE_USER_4

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_4
```

JOB_TYPE_USER_5

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_5
```

JOB_TYPE_USER_6

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_6
```

JOB_TYPE_USER_7

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_7
```

JOB_TYPE_USER_8

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_8
```

JOB_TYPE_USER_9

This constant has the following syntax:

```
public static final char JOB_TYPE_USER_9
```

DAY_SUNDAY

This constant has the following syntax:

```
public static final int DAY_SUNDAY
```

DAY_MONDAY

This constant has the following syntax:

```
public static final int DAY_MONDAY
```

DAY_TUESDAY

This constant has the following syntax:

```
public static final int DAY_TUESDAY
```

DAY_WEDNESDAY

This constant has the following syntax:

```
public static final int DAY_WEDNESDAY
```

DAY_THURSDAY

This constant has the following syntax:

```
public static final int DAY_THURSDAY
```

DAY_FRIDAY

This constant has the following syntax:

```
public static final int DAY_FRIDAY
```

DAY_SATURDAY

This constant has the following syntax:

```
public static final int DAY_SATURDAY
```

DAY_ALL

This constant has the following syntax:

```
public static final int DAY_ALL
```

PERMISSION_OWNER_EDIT

This constant has the following syntax:

```
public static final int PERMISSION_OWNER_EDIT
```

PERMISSION_OWNER_EXECUTE

This constant has the following syntax:

```
public static final int PERMISSION_OWNER_EXECUTE
```

PERMISSION_GROUP_EDIT

This constant has the following syntax:

```
public static final int PERMISSION_GROUP_EDIT
```

PERMISSION_GROUP_EXECUTE

This constant has the following syntax:

```
public static final int PERMISSION_GROUP_EXECUTE
```

PERMISSION_WORLD_EDIT

This constant has the following syntax:

```
public static final int PERMISSION_WORLD_EDIT
```

PERMISSION_WORLD_EXECUTE

This constant has the following syntax:

```
public static final int PERMISSION_WORLD_EXECUTE
```

CALENDAR_ACTION_DEFAULT

This constant has the following syntax:

```
public static final int CALENDAR_ACTION_DEFAULT
```

CALENDAR_ACTION_ONLY

This constant has the following syntax:

```
public static final int CALENDAR_ACTION_ONLY
```

CALENDAR_ACTION_SCHEDULE

This constant has the following syntax:

```
public static final int CALENDAR_ACTION_SCHEDULE
```

CALENDAR_ACTION_NEXT_DAY

This constant has the following syntax:

```
public static final int CALENDAR_ACTION_NEXT_DAY
```

CALENDAR_ACTION_NEXT_WORK_DAY

This constant has the following syntax:

```
public static final int CALENDAR_ACTION_NEXT_WORK_DAY
```

CALENDAR_ACTION_PREVIOUS_WORK_DAY

This constant has the following syntax:

```
public static final int CALENDAR_ACTION_PREVIOUS_WORK_DAY
```

JOB_BLOB_TYPE_STDIN

This constant has the following syntax:

```
public static final int JOB_BLOB_TYPE_STDIN
```

JOB_BLOB_TYPE_STDOUT

This constant has the following syntax:

```
public static final int JOB_BLOB_TYPE_STDOUT
```

JOB_BLOB_TYPE_STDERR

This constant has the following syntax:

```
public static final int JOB_BLOB_TYPE_STDERR
```

EXTERNAL_INSTANCE_TYPE_APPSERVER

This constant has the following syntax:

```
public static final java.lang.String EXTERNAL_INSTANCE_TYPE_APPSERVER
```

EXTERNAL_INSTANCE_TYPE_CONNECT

This constant has the following syntax:

```
public static final java.lang.String EXTERNAL_INSTANCE_TYPE_CONNECT
```

EXTERNAL_INSTANCE_TYPE_WORKLOAD

This constant has the following syntax:

```
public static final java.lang.String EXTERNAL_INSTANCE_TYPE_WORKLOAD
```

EXTERNAL_INSTANCE_UPDATE

This constant has the following syntax:

```
public static final int EXTERNAL_INSTANCE_UPDATE
```

EXTERNAL_INSTANCE_INSERT

This constant has the following syntax:

```
public static final int EXTERNAL_INSTANCE_INSERT
```

JOB_TYPE_OPERATION_INSERT

This constant has the following syntax:

```
public static final int JOB_TYPE_OPERATION_INSERT
```

JOB_TYPE_OPERATION_UPDATE

This constant has the following syntax:

```
public static final int JOB_TYPE_OPERATION_UPDATE
```

JOB_TYPE_OPERATION_DELETE

This constant has the following syntax:

```
public static final int JOB_TYPE_OPERATION_DELETE
```

NOTIFY_ALWAYS

This constant has the following syntax:

```
public static final char NOTIFY_ALWAYS
```

NOTIFY_ON_FAILURE

This constant has the following syntax:

```
public static final char NOTIFY_ON_FAILURE
```

NOTIFY_NEVER

This constant has the following syntax:

```
public static final char NOTIFY_NEVER
```

ENCRYPTION_TYPE_AES

This constant has the following syntax:

```
public static final int ENCRYPTION_TYPE_AES
```

ENCRYPTION_TYPE_CONFIG

This constant has the following syntax:

```
public static final int ENCRYPTION_TYPE_CONFIG
```

ENCRYPTION_TYPE_DEFAULT

This constant has the following syntax:

```
public static final int ENCRYPTION_TYPE_DEFAULT
```

ENCRYPTION_TYPE_NONE

This constant has the following syntax:

```
public static final int ENCRYPTION_TYPE_NONE
```

ATTRIBUTE_TYPE_INT

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_INT
```

ATTRIBUTE_TYPE_STRING

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_STRING
```

ATTRIBUTE_TYPE_DATE

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_DATE
```

ATTRIBUTE_TYPE_BOOLEAN

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_BOOLEAN
```

ATTRIBUTE_TYPE_INT_LIST

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_INT_LIST
```

ATTRIBUTE_TYPE_TIMEOFDAY_LIST

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_TIMEOFDAY_LIST
```

ATTRIBUTE_TYPE_CHKFILES_LIST

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_CHKFILES_LIST
```

ATTRIBUTE_TYPE_DOUBLE

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_DOUBLE
```

ATTRIBUTE_TYPE_CHAR

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_CHAR
```

ATTRIBUTE_TYPE_JOBITEM

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_JOBITEM
```

ATTRIBUTE_TYPE_LONG

This constant has the following syntax:

```
public static final int ATTRIBUTE_TYPE_LONG
```

AsErrorException

extends

AsRuntimeException

AsErrorException is thrown to relay error messages from the CA Workload Automation AE API to the Client applications.

AsErrorException

The default AsErrorException constructor has the following syntax:

```
public AsErrorException()
```

The cause and message are not set for the default constructor.

AsErrorException

This AsErrorException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsErrorException(java.lang.String message)
```

message

Defines the message.

AsErrorException

The AsErrorException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsErrorException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsErrorException

The AsErrorException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsErrorException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsException

extends

```
java.lang.Exception
```

AsException is the base checked exception for the CA Workload Automation AE API. All checked API exceptions should extend this class. Checked exceptions are used to relay API errors, return codes, and warning messages to the Client application.

AsException

The AsException default constructor has the following syntax:

```
public AsException()
```

The cause and message are not set for the default constructor.

AsException

The AsException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsException(java.lang.String message)
```

message

Defines the message.

AsException

The AsException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsException

The AsException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

parseMessage

The parseMessage method in the AsException class parses the API message string into key value pairs, returning an array of message objects. This method has the following syntax:

```
static AsMessage[] parseMessage(java.lang.String messageContent)
```

messageContent

Defines the message string.

getMessages

The getMessages method in the AsException class returns an array of key/value message objects. If the response does not have any warnings, the array is empty. This method has the following syntax:

```
AsMessage[] getMessages()
```

AsExternalValidationException

extends

AsException

AsExternalValidationException is thrown to relay external validation failure messages from the CA Workload Automation AE API to the Client applications. This failure occurs when the external job validation callout indicates a failure.

AsExternalValidationException

The default AsExternalValidationException constructor has the following syntax:

```
public AsExternalValidationException()
```

The cause and message are not set for the default constructor.

AsExternalValidationException

This AsExternalValidationException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsExternalValidationException(java.lang.String message)
```

message

Defines the message.

AsExternalValidationException

This AsExternalValidationException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsExternalValidationException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsExternalValidationException

This AsExternalValidationException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsExternalValidationException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the `Throwable.getMessage()` method.

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsFieldValidationException

extends

AsException

AsFieldValidationException is thrown to relay field validation failure messages from the CA Workload Automation AE API to the client applications. For example, when creating a job, this exception occurs if you try to create a command job using the `InsertUpdateJob` API but do not specify a machine.

AsFieldValidationException

This AsFieldValidationException constructor is the default constructor for which the cause and message are not set. This constructor has the following syntax:

```
AsFieldValidationException()
```

AsFieldValidationException

This AsFieldValidationException constructor creates a new exception with the specified detail message. This constructor has the following syntax:

```
AsFieldValidationException(java.lang.String message)
```

message

Specifies the message string.

AsFieldValidationException

This AsFieldValidationException constructor creates a new exception with the specified cause and a detail message of the cause. This constructor has the following syntax:

```
AsFieldValidationException(java.lang.Throwable cause)
```

cause

Indicates the cause of the exception. The cause is retrievable by calling the `Throwable.getCause()` method.

AsFieldValidationException

This AsFieldValidationException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
AsFieldValidationException(java.lang.String message, java.lang.Throwable cause)
```

message

Specifies the detailed message. The message is retrievable by calling the `Throwable.getMessage()` method.

cause

Indicates the cause of the exception. The cause is retrievable by calling the `Throwable.getCause()` method.

getMessages

The `getMessages` method in the `AsFieldValidationException` class returns the error messages. If the response does not have any warnings, the array will be empty. This method has the following syntax:

```
public AsMessage[] getMessages()
```

parseMessage

The `parseMessage` method in the `AsFieldValidationException` class parses the API message string into key value pairs and returns an array of message objects. This method has the following syntax:

```
public static AsMessage[] parseMessage(java.lang.String messageContent)
```

messageContent

Specifies the message string.

AsGeneralErrorException

extends

AsException

AsGeneralErrorException is thrown to relay general error messages from the CA Workload Automation AE API to Client applications. You must examine the contents in order to understand the nature of the error that generated this exception.

AsGeneralErrorException

The default AsGeneralErrorException constructor has the following syntax:

```
public AsGeneralErrorException()
```

The cause and message are not set for the default constructor.

AsGeneralErrorException

This AsGeneralErrorException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsGeneralErrorException(java.lang.String message)
```

message

Defines the message.

AsGeneralErrorException

This AsGeneralErrorException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsGeneralErrorException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsGeneralErrorException

This AsGeneralErrorException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsGeneralErrorException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsInitializationException

extends

```
AsRuntimeException
```

AsInitializationException is thrown to notify the Client application of API misuse. An example of misuse of the API is using an uninitialized object in a method which has an initialization prerequisite.

AsInitializationException

The AsInitializationException default constructor has the following syntax:

```
public AsInitializationException()
```

The cause and message are not set for the default constructor.

AsInitializationException

The AsInitializationException default constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsInitializationException(java.lang.String message)
```

message

Defines the message.

AsInitializationException

The AsInitializationException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsInitializationException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsInitializationException

The AsInitializationException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsInitializationException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsInvalidFilterResponseAttributeException

extends

AsRuntimeException

AsInvalidFilterResponseAttributeException is the exception thrown by the filter response when a client program tries to retrieve an attribute value that was not returned in the response.

AsInvalidFilterResponseAttributeException

This AsInvalidFilterResponseAttributeException is the default constructor. The cause and message are not set. This constructor has the following syntax:

```
public AsInvalidFilterResponseAttributeException()
```

AsInvalidFilterResponseAttributeException

This AsInvalidFilterResponseAttributeException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsInvalidFilterResponseAttributeException(java.lang.String message)
```

message

Specifies the message.

AsInvalidFilterResponseAttributeException

This AsInvalidFilterResponseAttributeException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsInvalidFilterResponseAttributeException(java.lang.Throwable cause)
```

cause

Indicates the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsInvalidFilterResponseAttributeException

This AsInvalidFilterResponseAttributeException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsInvalidFilterResponseAttributeException(java.lang.String message,  
java.lang.Throwable cause)
```

message

Specifies the detailed message. The message is retrievable by calling the `Throwable.getMessage()` method.

cause

Indicates the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

getMessages

The `getMessages` method in the `AsInvalidFilterResponseAttributeException` class returns error messages. If the response does not contain any warnings, the array will be empty. This method has the following syntax:

```
public AsMessage[] getMessages()
```

AsJobConstants

extends

`java.lang.Object`

The `AsJobConstants` class defines job constants for defining, updating, and reporting job definitions.

The job types are defined as:

`JOB_TYPE_Job_Type`

The job definition keywords are defined as:

`KEYWORD_Job_Definition_Keyword`

The keyword values are defines as:

`VALUE__Associated_Keyword__Value`

AsJobConstants

The `AsJobConstants` constructor has the following syntax:

```
public AsJobConstants()
```

AsJobProperties

extends

java.util.Properties

The AsJobProperties utility class retrieves information about job definition keywords. The data in this class is static, read-only, and tied to the CA Workload Automation AE release. The data is re-usable across instances that are running the same release of CA Workload Automation AE.

Note: The creation of this object is resource-intensive. You may want to create it once, and re-use that reference throughout other applications.

AsJobProperties

The AsJobProperties constructor has the following syntax:

```
public AsJobProperties()
```

getConstantFromKeyword

The getConstantFromKeyword method in the AsJobProperties class returns the integer constant from the specified keyword depending on the specified job type, or returns -1 if the keyword is undefined. This method has the following syntax:

```
public int getConstantFromKeyword(java.lang.String keyword, int job_type)  
keyword
```

Defines the keyword for the required constant.

```
job_type
```

Defines the job type that qualifies the keyword.

getKeywordFromConstant

The getKeywordFromConstant method in the AsJobProperties class returns the keyword from the specified integer constant, or null if the constant is undefined. This method has the following syntax:

```
public java.lang.String getKeywordFromConstant(int constant)  
constant
```

Defines the integer constant for the required keyword.

getJobTypeString

The `getJobTypeString` method in the `AsJobProperties` class returns the string value for the given job type constant. This method has the following syntax:

```
public java.lang.String getJobTypeString(int jobType)  
jobType
```

Defines the job type to get the string value for.

getJobTypeInt

The `getJobTypeInt` method in the `AsJobProperties` class returns the numeric value for the given job type string. This method has the following syntax:

```
public int getJobTypeInt(java.lang.String jobType)  
jobType
```

Defines the string format of job type.

getKeywordsForJobType

The `getKeywordsForJobType` method in the `AsJobProperties` class returns a list of all valid keywords for a given job type, or null if job type is undefined. This method has the following syntax:

```
public java.lang.String[] getKeywordsForJobType(int job_type)  
job_type
```

Defines the job type to get valid keywords for.

getKeywordDataType

The `getKeywordDataType` method in the `AsJobProperties` class returns the data type of a keyword value. This method has the following syntax:

```
public int getKeywordDataType(int keyword)  
keyword
```

Defines the integer constant for the desired keyword.

AsMachineDeletionException

extends

AsException

AsMachineDeletionException is thrown when the DeleteMachine request fails on component machines.

AsMachineDeletionException

The AsMachineDeletionException is a constructor for an error with known failing machines. This constructor has the following syntax:

```
public AsMachineDeletionException(java.lang.String message, java.lang.String[] failedMachines, java.lang.String[] nonExistentMachines)
```

message

Defines the error message.

failedMachines

Defines the list of machines that failed deletion with a General Error.

nonExistentMachines

Defines the list of machines that failed deletion with an Object Does Not Exist Error.

AsMachineDeletionException

The AsMachineDeletionException base constructor has the following syntax:

```
public AsMachineDeletionException()
```

AsMachineDeletionException

The AsMachineDeletionException constructor with a message has the following syntax:

```
public AsMachineDeletionException(java.lang.String message)
```

message

Defines the error Message.

AsMachineDeletionException

The AsMachineDeletionException constructor with a cause has the following syntax:

```
public AsMachineDeletionException(java.lang.Throwable cause)
```

cause

Defines the error cause.

AsMachineDeletionException

The AsMachineDeletionException constructor with a message and a cause has the following syntax:

```
public AsMachineDeletionException(java.lang.String message, java.lang.Throwable  
cause)
```

message

Defines the error Message.

cause

Defines the error cause.

getGeneralFailures

The getGeneralFailures method in the AsMachineDeletionException class returns the machine names that failed deletion with a general error. This method has the following syntax:

```
java.lang.String[] getGeneralFailures()
```

getObjectDoesNotExistFailures

The getObjectDoesNotExistFailures method in the AsMachineDeletionException class returns names of the machines that failed deletion with an object does not exist error. This method has the following syntax:

```
java.lang.String[] getObjectDoesNotExistFailures()
```

AsMessage

extends

`java.lang.Object`

AsMessage is a wrapper for the messages that the API returns with the key parsed out so that the user of the API can act programmatically based on the message ID and not need to inspect/parse the internationalized message.

getKey

The getKey method in the AsMessage class returns the key. This method has the following syntax:

`java.lang.String getKey()`

getMessage

The getMessage method in the AsMessage class returns the message. This method has the following syntax:

`java.lang.String getMessage()`

toString

The toString method in the AsMessage class has the following syntax:

`java.lang.String toString()`

AsNoAttributesException

extends

`AsException`

AsNoAttributesException is thrown to relay no attribute messages from the CA Workload Automation AE API to the Client applications. For example, if no attributes are specified when executing the GetJobsWithFilter API, this exception is thrown.

AsNoAttributesException

The default AsNoAttributesException constructor has the following syntax:

`public AsNoAttributesException()`

The cause and message are not set for the default constructor.

AsNoAttributesException

This AsNoAttributesException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsNoAttributesException(java.lang.String message)
```

message

Defines the message.

AsNoAttributesException

This AsNoAttributesException constructor creates a new exception with the specified cause and detailed message of the cause. This constructor has the following syntax:

```
public AsNoAttributesException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsNoAttributesException

This AsNoAttributesException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsNoAttributesException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the `Throwable.getMessage()` method.

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsObjectDoesNotExistException

extends

AsException

AsObjectDoesNotExistException is thrown to indicate that the requested object does not exist.

AsObjectDoesNotExistException

The AsObjectDoesNotExistException default constructor has the following syntax:

```
public AsObjectDoesNotExistException()
```

The cause and message are not set for the default constructor.

AsObjectDoesNotExistException

This AsObjectDoesNotExistException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsObjectDoesNotExistException(java.lang.String message)
```

message

Defines the message.

AsObjectDoesNotExistException

This AsObjectDoesNotExistException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsObjectDoesNotExistException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsObjectDoesNotExistException

This AsObjectDoesNotExistException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsObjectDoesNotExistException(java.lang.String message, java.lang.Throwable  
cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsObjectDoesNotExistException

This AsObjectDoesNotExistException constructor creates a new exception with the specified detailed message and cause.

```
public AsObjectDoesNotExistException(java.lang.String message, java.lang.String[]  
badObjects)
```

message

Defines the detailed message.

badObjects

Defines a list of bad objects.

getBadObjects

The getBadObjects method in the AsObjectDoesNotExistException class returns the list of requested objects that did not exist. This method has the following syntax:

```
a.lang.String[] getBadObjects()
```

AsRuntimeException

extends

`java.lang.RuntimeException`

AsRuntimeException is the base run-time exception for the CA Workload Automation AE API. All run-time API exceptions extend this class. Run-time exceptions are used to notify the Client application of API misuse. An example of misuse of the API is calling an execute method before calling a setRequest method.

AsRuntimeException

The AsRuntimeException default constructor has the following syntax:

```
public AsRuntimeException()
```

The cause and message are not set for the default constructor.

AsRuntimeException

This AsRuntimeException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsRuntimeException(java.lang.String message)
```

message

Defines the message.

AsRuntimeException

This AsRuntimeException constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsRuntimeException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsRuntimeException

This AsRuntimeException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsRuntimeException(java.lang.String message, java.lang.Throwable cause)  
message
```

message Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

getMessages

The getMessages method in the AsRuntimeException class returns an array of key/value message objects. If the response does not have any warnings, the array is empty. This method has the following syntax:

```
AsMessage[] getMessages()
```

AsSecurityException

extends

AsException

AsSecurityException is thrown to relay security related error and warning messages from the CA Workload Automation AE API to the Client applications.

AsSecurityException

The AsSecurityException default constructor has the following syntax:

```
public AsSecurityException()
```

The cause and message are not set for the default constructor.

AsSecurityException

This AsSecurityException default constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsSecurityException(java.lang.String message)
```

message

Defines the detailed message.

AsSecurityException

This AsSecurityException default constructor creates a new exception with the specified cause and a detail message of the cause. This constructor has the following syntax:

```
public AsSecurityException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsSecurityException

This AsSecurityException default constructor creates a new exception with the specified detail message and cause. This constructor has the following syntax:

```
public AsSecurityException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsTimeoutException

extends

AsRuntimeException

AsTimeoutException is thrown to notify Client applications of a CA Workload Automation AE API server time-out.

AsTimeoutException

The AsTimeoutException default constructor has the following syntax:

```
public AsTimeoutException()
```

The cause and message are not set for the default constructor.

AsTimeoutException

This AsTimeoutException constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsTimeoutException(java.lang.String message)
```

message

Defines the message.

AsTimeoutException

This AsTimeoutException constructor creates a new exception with the specified cause and a detailed message of the cause.

```
public AsTimeoutException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsTimeoutException

This AsTimeoutException constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsTimeoutException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the `Throwable.getMessage()` method.

cause

Defines the cause. The cause is retrievable by calling the `Throwable.getCause()` method.

AsUnknownErrorException

extends

AsRuntimeException

AsUnknownErrorException catches all exceptions for unknown API errors. This exception should not be thrown, but is in place as a fail-safe for unexpected errors. One possible reason for this exception to be thrown is if the Client libraries in use are communicating with a server that implements a newer version of the API that has additional error codes.

AsUnknownErrorException

The AsUnknownErrorException default constructor has the following syntax:

```
public AsUnknownErrorException()
```

The cause and message are not set for the default constructor.

AsUnknownErrorException

This AsUnknownErrorException default constructor creates a new exception with the specified detailed message. This constructor has the following syntax:

```
public AsUnknownErrorException(java.lang.String message)  
message
```

Defines the message.

AsUnknownErrorException

This AsUnknownErrorException default constructor creates a new exception with the specified cause and a detailed message of the cause. This constructor has the following syntax:

```
public AsUnknownErrorException(java.lang.Throwable cause)
```

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AsUnknownErrorException

This AsUnknownErrorException default constructor creates a new exception with the specified detailed message and cause. This constructor has the following syntax:

```
public AsUnknownErrorException(java.lang.String message, java.lang.Throwable cause)
```

message

Defines the detailed message. The message is retrievable by calling the Throwable.getMessage() method.

cause

Defines the cause. The cause is retrievable by calling the Throwable.getCause() method.

AuthenticatedUser

extends

java.lang.Object

implements

IAuthenticatedUser

The AuthenticatedUser class indicates the object that is returned from the AsApi.authenticateWithArtifact method. The object contains the user and node that were authenticated in eIAM with the artifact that was passed in.

getUser

The getUser method in the AuthenticatedUser class returns the user that was authenticated with the artifact. This method has the following syntax:

```
public java.lang.String getUser()
```

getNode

The getNode method in the AuthenticatedUser class returns the node that was authenticated with the artifact. This method has the following syntax:

```
public java.lang.String getNode()
```

Cat1Request

extends

 ApiRequest

implements

 ICat1Request

Cat1Request is an abstract class representing common functionality for all Category 1 requests.

execute

The execute method in the Cat1Request class performs the request and returns the corresponding response object. This method has the following syntax:

`IApiResponse execute(IAsApi api) throws AsException`

api

Specifies the API object to use for execution. This should be a valid connection.

AsException

Specifies the super class exception for any API specific exceptions that can occur. See specific requests for the individual exceptions that can be thrown.

setUser

The setUser method in the Cat1Request class sets the user for the request. This user is used for impersonation and potentially owns any objects created by this request. This method has the following syntax:

`void setUser(java.lang.String userName, java.lang.String machine)`

userName

Specifies the user name to execute this request.

machine

Specifies the computer on which this user is defined.

Cat2Request

extends

 ApiRequest

implements

 ICat2Request

Cat2Request is an abstract class representing common functionality for all Category 2 requests.

execute

The execute method in the Cat2Request class performs the request and returns a corresponding set of response objects. This method has the following syntax:

`IApiResponseSet execute(IAsApi api) throws AsInitializationException`

api

Specifies the API object to use for execution. This should be a valid connection.

AsInitializationException

Notifies the Client application of API misuse.

setUser

The setUser method in the Cat2Request class sets the user for the request. This user is used for impersonation and potentially owns any objects created by this request. This method has the following syntax:

`void setUser(java.lang.String userName, java.lang.String machine)`

userName

Specifies the user name to execute this request.

machine

Specifies the computer on which this user is defined.

ComponentMachine

extends

`java.lang.Object`

The ComponentMachine class describes a machine that is a part of a virtual machine.

ComponentMachine

This ComponentMachine constructor has the following syntax:

```
public ComponentMachine(java.lang.String machineName)
```

ComponentMachine

This ComponentMachine constructor has the following syntax:

```
public ComponentMachine(java.lang.String machineName, int maxload, double factor)
```

setMaxLoad

The setMaxLoad method in the ComponentMachine class has the following syntax:

```
public void setMaxLoad(int maxload)
```

setFactor

The setFactor method in the ComponentMachine class has the following syntax:

```
public void setFactor(double factor)
```

getMachineName

The getMachineName method in the ComponentMachine class has the following syntax:

```
public java.lang.String getMachineName()
```

getMaxLoad

The getMaxLoad method in the ComponentMachine class has the following syntax:

```
public int getMaxLoad()
```

getFactor

The getFactor method in the ComponentMachine class has the following syntax:

```
public double getFactor()
```

CreateParameterItem

extends

JobItem

CreateParameterItem is the container class for specifying create parameter arguments in job definitions. This JobItem subclass directly maps to the create_parameter jil keyword.

Example: Sample Code for Using the CreateParameterItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;
    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
                    AsJobConstants.JOB_TYPE_J2EE_SESSION_BEAN);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_BEAN_NAME, "myBean");
        insertReq.set(AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,
                    "myContext");
        insertReq.set(AsJobConstants.KEYWORD_METHOD_NAME, "myMethod");
        insertReq.set(AsJobConstants.KEYWORD_PROVIDER_URL, "myUrl");

        // Set up the create_parameter part of the job definition and
        // attach it to the base job definition.
        CreateParameterItem cpi = new CreateParameterItem();
        cpi.set(AsJobConstants.KEYWORD_CREATE_PARAMETER, "param1=p1");
        cpi.set(AsJobConstants.KEYWORD_CREATE_PARAMETER, "param2=p2");
        cpi.set(AsJobConstants.KEYWORD_CREATE_PARAMETER, "param3=p3");
        insertReq.add(cpi);
        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_CREATE_METHOD,  
        AsJobConstants.KEYWORD_CREATE_PARAMETER  
    };  
  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();  
  
        // Process job level properties.  
        System.out.println("----- Begin Response -----");  
        for(int i=0; i<rspProperties.length; ++i) {  
            String keyword =  
                jobProperties.getKeywordFromConstant(rspProperties[i]);  
            switch(getRsp.getAttributeType(rspProperties[i])) {  
                case AsConstants.ATTRIBUTE_TYPE_STRING:  
                    System.out.println(keyword + ": " +  
                        getRsp.getString(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_INT:  
                    System.out.println(keyword + ": " +  
                        getRsp.getInt(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                    System.out.println(keyword + ": " +  
                        getRsp.getBoolean(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_LONG:  
                    System.out.println(keyword + ": " +  
                        getRsp.getLong(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_JOBITEM:  
                    System.out.println("\n----- Begin " + keyword + " JobItem  
-----");  
                    JobItem ji = getRsp.getJobItem(rspProperties[i]);  
                    int itemProperties[] = ji.getAttributes();
```

```

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {
            System.out.print(keyword + ": ");

            switch(jobProperties.getKeywordDataType(itemProperties[j])) {
                case AsConstants.ATTRIBUTE_TYPE_STRING:
                    System.out.println(ji.getString(itemProperties[j]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_INT:
                    System.out.println(ji.getInt(itemProperties[j]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                    System.out.println(ji.getBool(itemProperties[j]));
                    break;
            }
            System.out.println("----- End " + keyword + " JobItem
-----\n");
            break;
        }
    }
    System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
}

```

Sample output:

```

----- Begin Response -----
job_name: create_parm_sample
alarm_if_fail: true
create_method: create

----- Begin create_parameter JobItem -----
create_parameter: param1=p1
create_parameter: param2=p2
create_parameter: param3=p3
----- End create_parameter JobItem -----

----- End Response -----

```

CreateParameterItem

This constructor has the following syntax:

```
public CreateParameterItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

DeleteBox

DeleteBox deletes a box and all of the jobs it contains. DeleteBox is used by JIL for the delete_box command.

DeleteBox verifies delete permissions for the specified box, then retrieves all contained jobs and deletes them. DeleteBox returns the name of each job as it is deleted, and is therefore a cat2 API.

When requested, the DeleteBox API also performs the following:

- The API returns a list of jobs that depend on each job deleted.
- The API fails when a job to be deleted is in ACTIVATED, STARTING, or RUNNING state. In jil, this is controlled by the RESTRICT_DELETE_JOB environment variable.
- The API fails when a job to be deleted still has dependent jobs. In jil, this is controlled by the RESTRICT_DELETE_DEPENDENT_JOB environment variable.

Example: Sample Code for Executing the DeleteBox API

```
private void sample(AsApi appServer, String boxName) {  
  
    // Set up the request.  
    IDelateBoxReq req = new DeleteBoxReq();  
    req.setRequest(boxName, true, true, true);  
  
    // Execute the request, check for errors, display results.  
    try {  
        IApiResponseSet set = req.execute(appServer);  
  
        while(set.hasNext()) {  
            DeleteBoxRsp rsp = (DeleteBoxRsp) set.next();  
  
            if(rsp.hasDependentJobs()) {  
                System.out.print("Jobs which are dependent on the\n");  
            }  
  
            System.out.println("DELETED Job: " + rsp.getDeletedJobName());  
  
            if(rsp.hasDependentJobs()) {  
                String[] dependentJobs = rsp.getDependentJobs();  
                for(int i = 0; i < dependentJobs.length; i++) {  
                    System.out.println("\t" + dependentJobs[i]);  
                }  
            }  
        }  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
}
```

```
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

DeleteBoxReq

extends

Cat2Request

implements

IDeleteBoxReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - You do not have authority to delete the box.

DeleteBoxReq

The DeleteBoxReq constructor creates a DeleteBoxReq object. This constructor has the following syntax:

```
public DeleteBoxReq()
```

setRequest

The setRequest method in the DeleteBoxReq class configures the request. This method has the following syntax:

```
void SetRequest (java.lang.String boxname, boolean verify, boolean RestrictDelJob,  
boolean RestrictDelDepJob,)
```

boxname

Specifies the name of the box to delete. Wildcards are not supported.

verify

Specifies whether to return the names of jobs that depend on each job deleted.

Possible values are:

- true—Returns a list of jobs that depend on each job deleted.
- false—Does not return a list of jobs that depend on each job deleted.

RestrictDelJob

Specifies whether the API should fail if the job being deleted is in the ACTIVATED, STARTED, or RUNNING state. Possible values are:

- true—Fails on deletion if the job is in ACTIVATED, STARTING, or RUNNING state.
- false—Deletes the job without regard to its state.

RestrictDelDepJob

Specifies whether the API should fail if the job being deleted has dependent jobs. Possible values are:

- true—Fails on deletion if a job is encountered that still has dependent jobs.
- false—Deletes the job without regard to its dependents.

[DeleteBoxRsp](#)

extends

ApiResponse

implements

IDeleteBoxRsp

This is an individual response that is returned when iterating through a DeleteBoxRspSet. If the initial delete box request is set to verify, each job contains a list of job names that are dependent on the deleted job represented by this response.

[DeleteBoxRsp](#)

The DeleteBoxRsp constructor creates a DeleteBoxRsp object. This constructor has the following syntax:

```
public DeleteBoxRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

[getDeletedJobName](#)

The getDeletedJobName method in the DeleteBoxRsp class returns the name of the deleted job. This method has the following syntax:

```
public java.lang.String getDeletedJobName()
```

[getDependentJobs](#)

The getDependentJobs method in the DeleteBoxRsp class returns an array containing the names of the successors for the deleted job. These strings are included in the response only if the request called for them. This method has the following syntax:

```
public java.lang.String[] getDependentJobs()
```

hasDependentJobs

The hasDependentJobs method in the DeleteBoxRsp class returns true if the number of dependent jobs associated with the deleted job is greater than zero. This method has the following syntax:

```
public boolean hasDependentJobs()
```

DeleteBoxRspSet

extends

ApiResponseSet

This is a response set produced from a delete box request. This set returns response objects of type DeleteBoxRsp. The first response represents the requested box, subsequent responses represent names of jobs that have been deleted as a result of this box deletion.

The following exceptions can be thrown when iterating over this set:

- AsApiGeneralException
- AsSecurityException—The set user does not have permission to remove the named box job.

DeleteBoxRspSet

The DeleteBoxRspSet constructor creates a DeleteBoxRspSet object. This constructor has the following syntax:

```
public DeleteBoxRspSet(com.ca.autosys.services.JResponseQ responseQ)
```

responseQ

Defines the raw responseQ object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

DeleteCalendarDate

DeleteCalendarDate deletes dates from a calendar. DeleteCalendarDate is used by autocal_asc.

DeleteCalendarDate checks delete permissions for a specified calendar name. After the calendar dates are deleted, audit information is appropriately modified, and events are sent or deleted as necessary for jobs affected by this calendar change.

Example: Sample Code for Executing the DeleteCalendarDate API

```
private void sample(AsApi api, String calendarName, Date[] dates2) {  
    // Set up the request.  
    ICalendarDateReq req = new DeleteCalendarDateReq();  
    req.setRequest(calendarName, dates2);  
  
    // Execute the request, check for errors.  
    try {  
        req.execute(api);  
        System.out.println("Dates successfully deleted.");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsObjectDoesNotExistException noObjEx) {  
        System.out.println("AsObjectDoesNotExistException: " +  
noObjEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

DeleteCalendarDateReq

extends

Cat1Request

implements

ICalendarDateReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to modify this calendar.
- AsObjectDoesNotExistException—At least one of the requested dates was not defined. No dates will have been deleted. The dates in error will be available in the response.

DeleteCalendarDateReq

The DeleteCalendarDateReq constructor creates a DeleteCalendarDateReq object. This constructor has the following syntax:

```
public DeleteCalendarDateReq()
```

setRequest

The setRequest method in the DeleteCalendarDateReq class specifies the name of the calendar and the dates to delete in this request. This method has the following syntax:

```
void setRequest(java.lang.String calendarName, java.util.Date[] calendarDates)
```

calendarName

Specifies the name of the calendar from which to delete dates.

calendarDates

Specifies the dates to delete.

DeleteCalendarDateRsp

extends

ApiResponse

The DeleteCalendarDateRsp class comprises the response for the delete calendar date request. This response contains no special information. An exception occurs if an error is detected.

DeleteCalendarDateRsp

The DeleteCalendarDateRsp constructor creates a DeleteCalendarDateRsp object. This constructor has the following syntax:

```
public DeleteCalendarDateRsp(com.ca.autosys.services.JResponse rsp)
```

A java.lang.IllegalArgumentException is thrown if a null response is given.

DeleteCycleCalendarDays

DeleteCycleCalendarDays deletes a cycle calendar from the CA Workload Automation AE database.

Example: Sample Code for Executing the DeleteCycleCalendarDays API

```
void sample(AsApi api, String cycleName) {  
    DeleteCycleCalendarDaysReq req = new DeleteCycleCalendarDaysReq();  
    req.setRequest(cycleName);  
  
    try {  
        req.execute(api);  
        System.out.println("Cycle successfully deleted.");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("General Exception Caught: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("Security Exception Caught: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AutoSys Exception Caught: " + asEx.getMessage());  
    }  
}
```

DeleteCycleCalendarDaysReq

extends

Cat1Request

implements

IDeleteCycleCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have authority to delete the cycle.

DeleteCycleCalendarDaysReq

The DeleteCycleCalendarDaysReq constructor creates a DeleteCycleCalendarDaysReq object. This constructor has the following syntax:

```
public DeleteCycleCalendarDaysReq()
```

setRequest

The setRequest method in the DeleteCycleCalendarDaysReq class sets up the request to delete a cycle calendar from the CA Workload Automation AE database. This method has the following syntax:

```
public void setRequest(java.lang.String cycleName)
```

cycleName

Specifies the name of the cycle calendar to delete.

DeleteCycleCalendarDaysRsp

extends

ApiResponse

implements

IDeleteCycleCalendarDaysRsp

The DeleteCycleCalendarDaysRsp class comprises the response for the delete cycle calendar days request. This response contains no special information.

DeleteCycleCalendarDaysRsp

The DeleteCycleCalendarDaysRsp constructor creates a DeleteCycleCalendarDaysRsp object. This constructor has the following syntax:

```
public DeleteCycleCalendarDaysRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

DeleteEntireCalendarDate

DeleteEntireCalendarDate removes a standard calendar from the CA Workload Automation AE database.

Example: Sample Code for Executing the DeleteEntireCalendarDate API

```
void Sample(String serverName, int serverPort, String calendarName) {
    // create the api.
    AsApi api = new AsApi(serverName, serverPort);

    // create the request.
    DeleteEntireCalendarDateReq req = new DeleteEntireCalendarDateReq();
    req.setRequest(calendarName);

    // execute it and get any return information
    try {
        req.execute(api);
        System.out.println("\nCalendar \\" + calendarName + "\\ has been deleted.");
    } catch (AsSecurityException secEx) {
        System.out.println("\nSecurity Violation: " + secEx.getMessage().trim());
    } catch (AsGeneralErrorException genEx) {
        System.out.println("\nGeneral Error: " + genEx.getMessage().trim());
    } catch (AsException e) {
        System.out.println("\nAutoSys Error: " + e.getMessage().trim());
    }
}
```

DeleteEntireCalendarDateReq

extends

Cat1Request

implements

IDeleteEntireCalendarDateReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - You do not have authority to delete the calendar.

DeleteEntireCalendarDateReq

The DeleteEntireCalendarDateReq constructor creates a DeleteEntireCalendarDateReq object. This constructor has the following syntax:

```
public DeleteEntireCalendarDateReq()
```

setRequest

The `setRequest` method in the `DeleteEntireCalendarDate` class creates the request to delete a calendar. This method has the following syntax:

```
public void setRequest(java.lang.String calendarName)  
calendarName
```

Defines the name of the calendar to delete.

DeleteEntireCalendarDateRsp

```
extends  
    ApiResponse  
implements  
    IDeleteEntireCalendarDateRsp
```

The `DeleteEntireCalendarDateRsp` class comprises the response for the delete entire calendar date request. This response contains no special information.

DeleteEntireCalendarDateRsp

The `DeleteEntireCalendarDateRsp` constructor creates a `DeleteEntireCalendarDateRsp` object. This constructor has the following syntax:

```
public DeleteEntireCalendarDateRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

A `java.lang.IllegalArgumentException` is thrown if a null response is given.

DeleteExtendedCalendarDays

DeleteExtendedCalendarDays deletes an extended calendar from the CA Workload Automation AE database.

Example: Sample Code for Executing the DeleteExtendedCalendarDays API

```
void Sample(AsApi api, String serverName, int serverPort, String calendarName) {  
    // create the request  
    DeleteExtendedCalendarDaysReq req = new DeleteExtendedCalendarDaysReq();  
    req.setRequest(calendarName);  
  
    // execute it and get any return information  
    try {  
        req.execute(api);  
        System.out.println("\nCalendar \\" + calendarName + "\" has been deleted.");  
    } catch (AsSecurityException secEx) {  
        System.out.println("\nSecurity Violation: " + secEx.getMessage().trim());  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("\nGeneral Error: " + genEx.getMessage().trim());  
    } catch (AsException e) {  
        System.out.println("\nAutoSys Error: " + e.getMessage().trim());  
    }  
}
```

DeleteExtendedCalendarDaysReq

extends

Cat1Request

implements

IDeleteExtendedCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have authority to delete the calendar.

DeleteExtendedCalendarDaysReq

The DeleteExtendedCalendarDaysReq constructor creates a DeleteExtendedCalendarDaysReq object. This constructor has the following syntax:

```
public DeleteExtendedCalendarDaysReq()
```

setRequest

The `setRequest` method in the `DeleteExtendedCalendarDaysReq` class sets up the request to delete an extended calendar. This method has the following syntax:

```
public void setRequest(java.lang.String calendarName)
```

calendarName

Specifies the calendar to delete.

DeleteExtendedCalendarDaysRsp

extends

ApiResponse

implements

IDeleteExtendedCalendarDaysRsp

The `DeleteExtendedCalendarDaysRsp` class comprises the response for the delete extended calendar days request. This response contains no special information. An exception occurs if an error is detected.

DeleteExtendedCalendarDaysRsp

The `DeleteExtendedCalendarDaysRsp` constructor creates a `DeleteExtendedCalendarDaysRsp` object. This constructor has the following syntax:

```
public DeleteExtendedCalendarDaysRsp(com.ca.autosys.services.JResponse rsp)
```

rsp

Defines the raw response object.

A `java.lang.IllegalArgumentException` is thrown if a null response is given.

DeleteExternalInstance

DeleteExternalInstance deletes external instances from the CA Workload Automation AE database.

Example: Sample Code for Executing the DeleteExternalInstance API

```
void sample(AsApi api, String instanceName) {
    DeleteExternalInstanceReq req = new DeleteExternalInstanceReq();
    req.setRequest( instanceName );

    try {
        req.execute( api );
        System.out.println( "Instance successfully deleted." );
    } catch ( AsGeneralErrorException genEx ) {
        System.out.println( "AsGeneralErrorException: " + genEx.getMessage() );
    } catch ( AsException asEx ) {
        System.out.println( "AsException: " + asEx.getMessage() );
    }
}
```

DeleteExternalInstanceReq

extends

Cat1Request

implements

IDeleteExternalInstanceReq

Execution of this request can result in AsGeneralErrorException being thrown.

DeleteExternalInstanceReq

The DeleteExternalInstanceReq constructor creates a DeleteExternalInstanceReq object. This constructor has the following syntax:

```
public DeleteExternalInstanceReq()
```

setRequest

The setRequest method in the DeleteExternalInstanceReq class configures the request for execution. This method has the following syntax:

```
public void setRequest(java.lang.String instanceName)
```

instanceName

Specifies the external instance name to delete.

DeleteExternalInstanceRsp

extends
 ApiResponse
implements
 IDeleteExternalInstanceRsp

DeleteExternalInstanceRsp is the response object for the DeleteExternalInstance API. This response contains no special information. An exception occurs if an error is detected.

DeleteExternalInstanceRsp

The DeleteExternalInstanceRsp constructor creates a DeleteExternalInstanceRsp object. This constructor has the following syntax:

```
public DeleteExternalInstanceRsp(com.ca.autosys.services.JResponse response)
```

DeleteGbloc

DeleteGbloc deletes global BLOBs from the CA Workload Automation AE database.

Example: Sample Code for Executing the DeleteGbloc API

```
void sample(AsApi api, String globalBlobName) {  
    DeleteGblocReq req = new DeleteGblocReq();  
    req.setRequest( globalBlobName );  
  
    try {  
        // execute the request and check for errors  
        req.execute(api);  
        System.out.println("\nBlob successfully deleted from the database.\n");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AutoSys General Error. " + genEx.getMessage());  
    } catch (AsObjectDoesNotExistException noObjEx) {  
        System.out.println("AutoSys No Attributes Error. " + noObjEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AutoSys Error. " + asEx.getMessage());  
    }  
}
```

DeleteGblobReq

extends
Cat1Request
implements
IDeleteGblobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsObjectDoesNotExistException—An invalid global BLOB name was specified.

DeleteGblobReq

The DeleteGblobReq constructor creates a DeleteGblobReq object. This constructor has the following syntax:

```
public DeleteGblobReq()
```

setRequest

The setRequest method in the DeleteGblobReq class configures the request to delete a global BLOB. This method has the following syntax:

```
void setRequest(java.lang.String blobName)
```

blobName

Specifies the name of the global BLOB to delete.

DeleteGblobRsp

extends
ApiResponse
implements
IDeleteGblobRsp

This is a response for the DeleteGblob API.

DeleteGblobRsp

The DeleteGblobRsp constructor creates a DeleteGblobRsp object. This constructor has the following syntax:

```
public DeleteGblobRsp(com.ca.autosys.services.JResponse response)
```

DeleteJblob

DeleteJblob deletes CA Workload Automation AE job blobs.

Example: Sample Code for Executing the DeleteJblob API

```
void sample( AsApi api, String jobName ) {
    // set up a response object
    DeleteJblobRsp response = null;

    // set up the request to delete the blob
    IDeleteJblobReq req = new DeleteJblobReq();
    req.setRequest( jobName, true );

    // execute the request and check for errors
    try {
        response = ( DeleteJblobRsp ) req.execute( api );
        System.out.println("\nDelete successful. The latest input blob was deleted,
and there are " + response.getNumberOfVersions() + " version(s) remaining.\n\n");
    } catch ( AsGeneralErrorException genEx ) {
        System.out.println( "AsGeneralErrorException: " + genEx.getMessage() );
    } catch ( AsBadAttributesException badAttEx ) {
        System.out.println( "AsBadAttributesException: " + badAttEx.getMessage()
);
    } catch ( AsBadFilterFieldsException badFiltFldsEx ) {
        System.out.println( "AsBadFilterFieldsException: " +
badFiltFldsEx.getMessage() );
    } catch ( AsObjectDoesNotExistException objNoExistEx ) {
        System.out.println( "AsObjectDoesNotExistException: " +
objNoExistEx.getMessage() );
    } catch ( AsException asEx ) {
        System.out.println( "AsException: " + asEx.getMessage());
    }
}
```

DeleteJblobReq

extends

Cat1Request

implements

IDeleteJblobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsBadAttributesException—An invalid job name was passed.
- AsObjectDoesNotExistException—The BLOB does not exist.

DeleteJblobReq

The DeleteJblobReq constructor creates a DeleteJblobReq object. This constructor has the following syntax:

```
public DeleteJblobReq()
```

setRequest

The setRequest method in the DeleteJblobReq class configures the DeleteJblob request. This method has the following syntax:

```
public void setRequest(java.lang.String jobName, boolean latestInputBlob)  
jobName
```

Specifies the job name to which the BLOB is tied.

latestInputBlob

Specifies the BLOB to delete. Possible values are:

- true—Deletes the latest version of the input BLOB.
- false—Deletes both the STDOUT and STDERR BLOBS.

DeleteJblobRsp

extends

ApiResponse

implements

IDeleteJblobRsp

DeleteJblobRsp is the response object for the DeleteJblob API.

DeleteJblobRsp

The DeleteJblobRsp constructor creates a DeleteJblobRsp object. This constructor has the following syntax:

```
public DeleteJblobRsp(com.ca.autosys.services.JResponse response)
```

getNumberOfVersions

The getNumberOfVersions method in the DeleteJblobRsp class returns the number of input BLOB versions. This is valid only when latest InputBlob is set to true in the deletion request. This method has the following syntax:

```
public int getNumberOfVersions()
```

DeleteJob

DeleteJob deletes the specified job. DeleteJob verifies delete permissions for the specified job before deleting it. Dependent jobs are orphaned.

When requested, the DeleteJob API also performs the following:

- The API returns a list of jobs that depend on the job being deleted.
- The API fails when a job to delete is in the ACTIVATED, STARTING, or RUNNING state. In jil, this is controlled by the RESTRICT_DELETE_JOB environment variable.
- The API fails when the job to delete still has dependent jobs. In jil, this is controlled by the RESTRICT_DELETE_DEPENDENT_JOB environment variable.

Example: Sample Code for Executing the DeleteJob API

```
public void sample(AsApi api, String jobName) {  
    // Set up the request.  
    IDeleteJobReq req = new DeleteJobReq();  
    req.setRequest(jobName, true, true, false);  
  
    // Execute the request, check for errors, and display the results.  
    try {  
        DeleteJobRsp rsp = (DeleteJobRsp) req.execute(api);  
  
        if(rsp.hasDependentJobs()) {  
            System.out.println("Jobs which are dependent on the");  
        }  
  
        System.out.println("DELETED Job: " + jobName);  
  
        if(rsp.hasDependentJobs()) {  
            String[] dependentJobs = rsp.getDependentJobs();  
            for(int i = 0; i < dependentJobs.length; i++) {  
                System.out.println("\t" + dependentJobs[i]);  
            }  
        }  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

DeleteJobReq

extends

Cat1Request

implements

IDeleteJobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to modify this job.

DeleteJobReq

The DeleteJobReq constructor creates a DeleteJobReq object. This constructor has the following syntax:

```
public DeleteJobReq()
```

setRequest

The setRequest method in the DeleteJobReq class configures the request to delete the job. This method has the following syntax:

```
public void setRequest(java.lang.String jobname, boolean listDep, boolean  
                  restrictDelJob, boolean restrictDelDepJob,)
```

jobname

Specifies the name of the job to delete. Wildcards are not supported.

listDep

Specifies whether to return the names of jobs that depend on a job being deleted.
Possible values are:

- True—Returns a list of jobs that depend on the job being deleted.
- False—Does not return a list of jobs that depend on the job being deleted.

restrictDelJob

Specifies whether the API should fail if the job being deleted is in ACTIVATED, STARTED, or RUNNING state. Possible values are:

- True—Fails the API if the job to delete is in ACTIVATED, STARTED, or RUNNING state.
- False—Deletes the job without regard to its state.

restrictDelDepJob

Specifies whether the API should fail if the job being deleted has dependent jobs.
Possible values are:

- True—Fails the API if the job to delete still has dependent jobs.
- False—Deletes the job without regard to its dependents.

DeleteJobRsp

extends

ApiResponse

implements

IDeleteJobRsp

This is a response for the delete job request.

DeleteJobRsp

The DeleteJobRsp constructor creates a DeleteJobRsp object. This constructor has the following syntax:

```
public DeleteJobRsp(com.ca.autosys.services.JResponse rsp)
```

rsp

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

getDependentJobs

The getDependentJobs method in the DeleteJobRsp class returns an array containing the names of the successors of all deleted jobs. This method has the following syntax:

```
public java.lang.String[] getDependentJobs()
```

hasDependentJobs

The hasDependentJobs method in the DeleteJobRsp class returns true if the number of dependent jobs associated with the deleted job is greater than zero. This can only return true if listDep was specified in the setRequest() method. This method has the following syntax:

```
public boolean hasDependentJobs()
```

DeleteMachine

DeleteMachine removes machines from the database. DeleteMachine is used by JIL to delete machines.

DeleteMachine lets you delete real machines, entire virtual machines, or components of a virtual machine. DeleteMachine verifies delete access for the machine, then deletes the machine and updates audit information.

Example: Sample Code for Executing the DeleteMachine API

```
void sample(AsApi api, String machineName) {
    // Set up the request.
    IDeleteMachineReq req = new DeleteMachineReq();
    req.setRequest(machineName);

    // Execute the request, check for errors.
    try {
        req.execute(api);
        System.out.println("Machine \\" + machineName + "\\ has been deleted.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsObjectDoesNotExistException noObjEx) {
        System.out.println("AsObjectDoesNotExistException: " +
noObjEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

DeleteMachineReq

extends

Cat1Request

implements

IDeleteMachineReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to modify this machine.
- AsObjectDoesNotExistException—The specified machine does not exist.

DeleteMachineReq

The DeleteMachineReq constructor creates a DeleteMachineReq object. This constructor has the following syntax:

```
public DeleteMachineReq()
```

setRequest

This setRequest method in the DeleteMachineReq class specifies which machine to delete. This method has the following syntax:

```
public void setRequest(java.lang.String machine)
```

machine

Specifies the name of real or virtual machine to delete.

When you set machine to the name of a virtual machine, the API deletes the virtual machine and all of its components.

setRequest

This setRequest method in the DeleteMachineReq class specifies which machine and references of a real machine in virtual containers to delete. This method has the following syntax:

```
public void setRequest(java.lang.String machine, boolean remove_references)
```

machine

Specifies the name of real or virtual machine to delete.

When you set machine to the name of a virtual machine, the API deletes the virtual machine and all of its components.

remove_references

Specifies whether to remove references of a real machine in all virtual containers. Possible values are:

- True—Remove all reference of a real machine in all virtual containers.
- False—Does not remove references of a real machine in all virtual containers.

setRequest

This setRequest method in the DeleteMachineReq class specifies which virtual machine and specific component machines to delete. This method has the following syntax:

```
public void setRequest(java.lang.String machine, java.lang.String[]  
                      componentMachines)
```

machine

Specifies the name virtual machine to delete.

When you set machine to the name of a virtual machine but you do not specify a *componentMachines* value, the API deletes the virtual machine and all of its components.

componentMachines

Specifies the names of component machines to delete from the specified virtual machine.

DeleteMachineRsp

extends
 ApiResponse
implements
 IDeleteMachineRsp

DeleteMachineRsp

The DeleteMachineRsp constructor creates a DeleteMachineRsp object. This constructor has the following syntax:

```
public DeleteMachineRsp(com.ca.autosys.services.JResponse rsp)
```

rsp

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

getDeletedMachines

The getDeletedMachines method in the DeleteMachineRsp class returns an array containing the names of component machines that were deleted. This method has the following syntax:

```
java.lang.String[] getDeletedMachines()
```

hasDeletedMachines

The hasDeletedMachines method in the DeleteMachineRsp class returns true if there are names in the deleted machines list. This method has the following syntax:

```
boolean hasDeletedMachines()
```

DeleteResource

DeleteResource lets you delete resource information from the CA Workload Automation AE database.

Example: Sample Code for Using the DeleteResourceReq API

```
private void sample(AsApi api, String RESOURCE_NAME, String MACHINE_NAME) {  
    // Set up the request.  
    IDeleteResourceReq req = new DeleteResourceReq();  
    req.setName(RESOURCE_NAME);  
    req.setMachine(MACHINE_NAME);  
    req.setRequest();  
    req.setUser(TEST_USER, SERVER_NAME);  
  
    // Execute the request, check for errors.  
    try {  
        req.execute(api);  
        System.out.println("Resource " + RESOURCE_NAME + " has been deleted from  
Machine " + MACHINE_NAME);  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

DeleteResourceReq

extends
Cat1Request
implements
IDeleteResourceReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to insert the resource.

DeleteResourceReq

The DeleteResourceReq constructor creates a DeleteResourceReq object. This constructor has the following syntax:

```
public DeleteResourceReq()
```

setRequest

The setRequest method in the DeleteResourceReq class configures the request to delete a resource. This method has the following syntax:

```
public void setRequest()
```

setMachine

This is a setter method for the machine name on which the resource to be deleted exists. This method has the following syntax:

```
public void setMachine(java.lang.String resource_machine)  
resource_machine
```

Specifies the name of the machine on which the resource to be deleted exists.

setName

This is a setter method for the resource name. This method has the following syntax:

```
public void setName(java.lang.String resource_name)  
resource_name
```

Specifies the name of the resource to be deleted.

DeleteResourceRsp

extends

ApiResponse

implements

IDeleteResourceRsp

This is a response for delete resource request.

DeleteResourceRsp

The DeleteResourceRsp constructor creates a DeleteResourceRsp object. This constructor has the following syntax:

```
public DeleteResourceRsp(com.ca.autosys.services.JResponse response)
```

EnvironmentVariableItem

extends

JobItem

EnvironmentVariableItem is the container class for specifying environment variables in job definitions. This JobItem subclass directly maps to the envvars jil keyword.

Example: Sample Code for Using the EnvironmentVariableItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_COMMAND);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_COMMAND, "myCommand");

        // Set up the envvars part of the job definition and
        // attach it to the base job definition.
        EnvironmentVariableItem evi = new EnvironmentVariableItem();
        evi.set(AsJobConstants.KEYWORD_ENVARS, "var1=value1");
        evi.set(AsJobConstants.KEYWORD_ENVARS, "var2=value2");
        evi.set(AsJobConstants.KEYWORD_ENVARS, "var3=value3");
        insertReq.add(evi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
        JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);
    }
}
```

```
// Set up what attributes are to be returned.  
int attributes[] = {  
    AsJobConstants.KEYWORD_JOB_NAME,  
    AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
    AsJobConstants.KEYWORD_MACHINE,  
    AsJobConstants.KEYWORD_COMMAND,  
    AsJobConstants.KEYWORD_ENVARS  
};  
  
// Ready the request and execute it.  
getReq.setRequest(filter, attributes);  
set = getReq.execute(appServer);  
  
// Process the returned responses.  
while(set.hasNext()) {  
    getRsp = (GetJobsWithFilterRsp)set.next();  
    int rspProperties[] = getRsp.getAttributes();  
  
    // Process job level properties.  
    System.out.println("----- Begin Response -----");  
    for(int i=0; i<rspProperties.length; ++i) {  
        String keyword =  
jobProperties.getKeywordFromConstant(rspProperties[i]);  
        switch(getRsp.getAttributeType(rspProperties[i])) {  
            case AsConstants.ATTRIBUTE_TYPE_STRING:  
                System.out.println(keyword + ": " +  
getRsp.getString(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_INT:  
                System.out.println(keyword + ": " +  
getRsp.getInt(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                System.out.println(keyword + ": " +  
getRsp.getBoolean(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_LONG:  
                System.out.println(keyword + ": " +  
getRsp.getLong(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_JOBITEM:  
                System.out.println("\n----- Begin " + keyword + " JobItem  
-----");  
                JobItem ji = getRsp.getJobItem(rspProperties[i]);  
                int itemProperties[] = ji.getAttributes();  
  
                // Process job item level properties.  
                for(int j=0; j<itemProperties.length; ++j) {  
                    System.out.print(keyword + ": ");
```

```
switch(jobProperties.getKeywordDataType(itemProperties[j])) {  
    case AsConstants.ATTRIBUTE_TYPE_STRING:  
  
        System.out.println(ji.getString(itemProperties[j]));  
        break;  
    case AsConstants.ATTRIBUTE_TYPE_INT:  
  
        System.out.println(ji.getInt(itemProperties[j]));  
        break;  
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
  
        System.out.println(ji.getBool(itemProperties[j]));  
        break;  
    }  
}  
System.out.println("----- End " + keyword + " JobItem  
-----\n");  
break;  
}  
}  
System.out.println("----- End Response -----");  
}  
} catch (AsInitializationException e) {  
    e.printStackTrace();  
} catch (AsException e) {  
    e.printStackTrace();  
}  
}  
}
```

Sample output

```
----- Begin Response -----  
job_name: envvars_sample  
alarm_if_fail: true  
machine: localhost  
command: myCommand  
  
----- Begin envvars JobItem -----  
envvars: var1=value1  
envvars: var2=value2  
envvars: var3=value3  
----- End envvars JobItem -----  
  
----- End Response -----
```

EnvironmentVariableItem

This constructor has the following syntax:

```
public EnvironmentVariableItem()
```

EventRequest

extends

Cat2Request

implements

IEventRequest

EventRequest

This EventRequest constructor creates an EventRequest object. This constructor has the following syntax:

```
public EventRequest()
```

setJobName

The setJobName method in the EventRequest class sets the job name in the event request and is valid for all events except the following:

- STOP_DEMON_EVENT
- COMMENT_EVENT
- ALARM_EVENT
- SET_GLOBAL_EVENT.

This method has the following syntax:

```
public void setJobName(java.lang.String job)
```

job

Defines the job name for the request.

setEventType

The `setEventType` method in the `EventRequest` class sets the event type of the request and is valid only for the following event types:

- `EVENT_CHANGE_STATUS`
- `EVENT_KILLJOB`
- `EVENT_ALARM`
- `EVENT_STARTJOB`
- `EVENT_FORCE_STARTJOB`
- `EVENT_STOP_DEMON`
- `EVENT_JOB_ON_ICE`
- `EVENT_JOB_OFF_ICE`
- `EVENT_JOB_ON_HOLD`
- `EVENT_JOB_OFF_HOLD`
- `EVENT_COMMENT`
- `EVENT_DELETEJOB`
- `EVENT_CHANGE_PRIORITY`
- `EVENT_SET_GLOBAL`
- `EVENT_SEND_SIGNAL`
- `EVENT_MACH_OFFLINE`
- `EVENT_MACH_ONLINE`
- `EVENT_RELEASE_RESOURCE`
- `EVENT_REPLY_RESPONSE`
- `EVENT_RESTARTJOB`

This method has the following syntax:

```
public void setEventType(int type)
```

type

Defines the event type for the request.

setSignals

The `setSignals` method in the `EventRequest` class sets the signals in the event request and is valid only for the `SEND_SIGNAL` and `KILLJOB` types.

This method has the following syntax:

```
public void setSignals(int[] signals)
```

signals

Defines the signals for the request.

setStatus

The `setStatus` method in the `EventRequest` class sets the status in the event request and is valid only for the `CHANGE_STATUS` type. This method has the following syntax:

```
public void setStatus(int status)
```

status

Defines the status in the request.

Valid status values are:

- `STATUS_RUNNING`
- `STATUS_STARTING`
- `STATUS_ASSUCCESS`
- `STATUS_FAILURE`
- `STATUS_TERMINATED`
- `STATUS_ON_ICE`
- `STATUS_INACTIVE`
- `STATUS_ACTIVATED`
- `STATUS_RESTART`
- `STATUS_ON_HOLD`
- `STATUS_QUE_WAIT`
- `STATUS_PEND_MACH`

setAlarmId

The `setAlarmId` method in the `EventRequest` class sets the alarm ID in the event request and is valid only for the `ALARM` type. This method has the following syntax:

```
public void setAlarmId(int alarm)
```

alarm

Defines the alarm ID in the request.

Valid alarm values are:

- `ALARM_FORKFAIL`
- `ALARM_MINRUNALARM`
- `ALARM_JOBFAILURE`
- `ALARM_MAX_RETRY`
- `ALARM_STARTJOBFAIL`

- ALARM_EVENT_HDLR_ERROR
- ALARM_EVENT_QUE_ERROR
- ALARM_JOBNOT_ONICEHOLD
- ALARM_MAXRUNALARM
- ALARM_RESOURCE
- ALARM_MISSING_HEARTBEAT
- ALARMCHASE
- ALARM_DATABASE_COMM
- ALARM_VERSION_MISMATCH
- ALARM_DB_ROLLOVER
- ALARM_EP_ROLLOVER
- ALARM_EP_SHUTDOWN
- ALARM_EP_HIGH_AVAIL
- ALARM_DB_PROBLEM
- ALARM_DUPLICATE_EVENT
- ALARM_INSTANCE_UNAVAILABLE
- ALARM_AUTO_PING
- ALARM_OB_SHUTDOWN
- ALARM_EXTERN_DEPS_ERROR
- ALARM_MACHINE_UNAVAILABLE
- ALARM_SERVICEDESK_FAILURE
- ALARM_UNINOTIFY_FAILURE
- ALARM_CPI_JOBNAME_INVALID
- ALARM_CPI_UNAVAILABLE

setGlobalVariableName

The `setGlobalVariableName` method in the `EventRequest` class sets the global variable name in the event request and is valid only for the `SET_GLOBAL` type. This method has the following syntax:

```
public void setGlobalVariableName(java.lang.String name)  
name
```

Defines the global variable name in the request.

setGlobalVariableValue

The `setGlobalVariableValue` method in the `EventRequest` class sets the global variable value in the event request and is valid only for the `SET_GLOBAL` type. This method has the following syntax:

```
public void setGlobalVariableValue(java.lang.String value)  
value
```

Defines the global variable value in the request.

setMachineName

The `setMachineName` method in the `EventRequest` class sets the machine name in the event request and is valid only for the `MACH_ONLINE` and `MACH_OFFLINE` types. This method has the following syntax:

```
public void setMachineName(java.lang.String machine)  
machine
```

Defines the machine name in the request.

setComment

The `setComment` method in the `EventRequest` class sets the comment in the event request and is valid only for the `COMMENT` type. This method has the following syntax:

```
public void setComment(java.lang.String comment)  
comment
```

Defines the comment in the request.

setPriority

The `setPriority` method in the `EventRequest` class sets the priority of the event request and is valid for all event types. This method has the following syntax:

```
public void setPriority(int priority)
```

priority

Defines the priority of the request.

setMaxSendTries

The `setMaxSendTries` method in the `EventRequest` class sets the maximum number of send tries in the event request and is valid for all event types. This method has the following syntax:

```
public void setMaxSendTries(int maxTries)
```

maxTries

Defines the maximum number of send tries in the request.

setJobQuePriority

The `setJobQuePriority` method in the `EventRequest` class sets the job que priority in the event request and is valid for all event types. This method has the following syntax:

```
public void setJobQuePriority(int qPriority)
```

qPriority

Defines the job queue priority in the request.

setEventTime

The `setEventTime` method in the `EventRequest` class sets the event time in the event request and is valid for all event types. This method has the following syntax:

```
public void setEventTime(java.util.Date time)
```

time

Defines the event time in the request.

setApplication

The `setApplication` method in the `EventRequest` class sets the application in the event request and is valid for all events except the following:

- `STOP_DEMON_EVENT`
- `COMMENT_EVENT`
- `ALARM_EVENT`
- `SET_GLOBAL_EVENT`.

This method has the following syntax:

```
public void setApplication(java.lang.String application)  
application
```

Defines the application in the request.

setGroup

The `setGroup` method in the `EventRequest` class sets the group in the event request and is valid for all events except the following:

- `STOP_DEMON_EVENT`
- `COMMENT_EVENT`
- `ALARM_EVENT`
- `SET_GLOBAL_EVENT`.

This method has the following syntax:

```
public void setGroup(java.lang.String group)  
group
```

Defines the group in the request.

setReply

The `setReply` method in the `EventRequest` class sets the reply to an outstanding response and is valid only for `REPLY_RESPONSE` events. This method has the following syntax:

```
public void setReply(java.lang.String reply)  
reply
```

Defines the reply data in the request.

setStepName

The setStepName method in the EventRequest class sets the Step Name to restart a failed MicroFocus job and is valid only for RESTARTJOB events. This method has the following syntax:

```
public void setStepName(java.lang.String stepname)  
stepname
```

Defines the step name in the request.

FetchGblob

FetchGblob retrieves a global BLOB from the CA Workload Automation AE database.

Example: Sample Code for Executing the FetchGblob API

```
void sample(AsApi api, String globalBlobName) {  
    FetchGblobRsp rsp = null;  
    FetchGblobReq req = new FetchGblobReq();  
    req.setRequest( globalBlobName );  
  
    try {  
        // execute the request and check for errors  
        rsp = (FetchGblobRsp) req.execute(api);  
        System.out.println( "\nGlobal blob " + globalBlobName +  
            " successfully fetched. The blob was " +  
            rsp.getBlob().length + " bytes in size.");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AutoSys General Error. " + genEx.getMessage());  
    } catch (AsObjectDoesNotExistException noObjEx) {  
        System.out.println("AutoSys No Attributes Error. " + noObjEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AutoSys Error. " + asEx.getMessage());  
    }  
}
```

FetchGblocReq

extends
Cat1Request
implements
IFetchGblocReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsObjectDoesNotExistException—Invalid global BLOB name specified.

FetchGblocReq

The FetchGblocReq constructor creates a FetchGblocReq object. This constructor has the following syntax:

```
public FetchGblocReq()
```

setRequest

The setRequest method in the FetchGblocReq class configures the request for execution. This method has the following syntax:

```
public void setRequest(java.lang.String blobName)
```

blobName

Specifies the BLOB name to fetch.

FetchGblocRsp

extends
ApiResponse
implements
IFetchGblocRsp

FetchGblocRsp is the response object for the FetchGbloc API.

FetchGblocRsp

The FetchGblocRsp constructor creates a FetchGblocRsp object. This constructor has the following syntax:

```
public FetchGblocRsp(com.ca.autosys.services.JResponse response)
```

getBlob

The getBlob method in the FetchGblobRsp class returns the requested global BLOB. This method has the following syntax:

```
public byte[] getBlob()
```

getCreationUser

The getCreationUser method in the FetchGblobRsp class returns the name of the user that created the BLOB. This method has the following syntax:

```
public java.lang.String getCreationUser()
```

getModificationUser

The getModificationUser method in the FetchGblobRsp class returns the name of the user that modified the BLOB. This method has the following syntax:

```
public java.lang.String getModificationUser()
```

getCreationTime

The getCreationTime method in the FetchGblobRsp class returns the creation time for the BLOB. This method has the following syntax:

```
public java.lang.String getCreationTime()
```

getModificationTime

The getModificationTime method in the FetchGblobRsp class returns the modification time for the BLOB. This method has the following syntax:

```
public java.lang.String getModificationTime()
```

FetchJblob

FetchJblob retrieves JBLOBs that are tied to a CA Workload Automation AE job.

Example: Sample Code for Executing the FetchJblob API

```
void sample( AsApi api, String jobName ) {
    // set up a response object
    FetchJblobRsp response = null;

    // set up the request to fetch the blob
    IFetchJblobReq req = new FetchJblobReq();
    req.setRequest( jobName, AsConstants.JOB_BLOB_TYPE_STDIN, 1 );

    // execute the request and check for errors
    try {
        response = ( FetchJblobRsp ) req.execute( api );
        System.out.println("\nFetch successful. Blob was " +
response.getBlob().length + " bytes.\n\n");
    } catch ( AsGeneralErrorException genEx ) {
        System.out.println( "AsGeneralErrorException: " + genEx.getMessage() );
    } catch ( AsBadAttributesException badAttEx ) {
        System.out.println( "AsBadAttributesException: " + badAttEx.getMessage()
);
    } catch ( AsBadFilterFieldsException badFiltFldsEx ) {
        System.out.println( "AsBadFilterFieldsException: " +
badFiltFldsEx.getMessage() );
    } catch ( AsObjectDoesNotExistException objNoExistEx ) {
        System.out.println( "AsObjectDoesNotExistException: " +
objNoExistEx.getMessage() );
    } catch ( AsException asEx ) {
        System.out.println( "AsException: " + asEx.getMessage());
    }
}
```

FetchJblobReq

extends

Cat1Request

implements

IFetchJblobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsBadAttributesException - An invalid job name was specified.
- AsBadFilterFieldsException - An invalid BLOB version was specified.
- AsObjectDoesNotExistException - The specified job name does not have any BLOBS associated with it.

setRequest

The setRequest method in the FetchJblobReq class configures the request to fetch a JBLOB from a job. This method has the following syntax:

```
void setRequest(java.lang.String jobName, int blobType, int blobVersion)
```

jobName

Specifies the existing job name to which the BLOB is tied.

blobType

Specifies the type of BLOB to retrieve. Possible values are:

- AsConstants.JOB_BLOB_TYPE_STDIN
- AsConstants.JOB_BLOB_TYPE_STDOUT
- AsConstants.JOB_BLOB_TYPE_STDERR

blobVersion

Specifies the version number of the BLOB to fetch. Version number 0 fetches the latest version.

Note: This parameter is ignored if the BLOB type is not equal to JOB_BLOB_TYPE_STDIN.

FetchJblobRsp

extends
 ApiResponse
implements
 IFetchJblobRsp

FetchJblobRsp is the response for fetching JBLOBs from CA Workload Automation AE jobs.

getBlob

The getBlob method in the FetchJblobRsp class returns the byte array that represents the BLOB. This method has the following syntax:

```
byte[] getBlob()
```

FilterChar

FilterChar is a common class used to track the construction of the filter.

extends

com.ca.autosys.services.request.filter.Filter

implements

IFilterLeaf

FilterChar

This FilterChar constructor creates a FilterChar object. This constructor has the following syntax:

```
public FilterChar(int id, char value, char[] values, int operator)
```

getFilterId

The getFilterId method in the FilterChar class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getValue

The getValue method in the FilterChar class returns the value. This method has the following syntax:

```
public char getValue()
```

getValues

The getValues method in the FilterChar class returns the values. This method has the following syntax:

```
public char[] getValues()
```

getOperator

The getOperator method in the FilterChar class returns the operator. This method has the following syntax:

```
public int getOperator()
```

FilterSort

The FilterSort object allows sort criteria to be added to filtering API requests.

extends

java.lang.Object

implements

IFilterSort

FilterSort

This FilterSort constructor creates a FilterSort object. This constructor has the following syntax:

```
public FilterSort()
```

FilterSort

This FilterSort constructor creates a FilterSort object with a known sort order and sort attribute. This constructor has the following syntax:

```
public FilterSort(int sortOrder, int sortAttribute)
```

sortOrder

Defines the sort order. Possible values are:

- ORDER_ASCENDING
- ORDER_DESCENDING

sortAttribute

Defines the attribute to sort by.

add

The add method in the FilterSort class adds sorting criteria to a sorting filter. This method has the following syntax:

```
public void add(int sortOrder, int sortAttribute)
```

sortOrder

Defines the sort order. Possible values are:

- ORDER_ASCENDING
- ORDER_DESCENDING

sortAttribute

Defines the attribute to sort by.

FinderParameterItem

extends

JobItem

FinderParameterItem is the container class for specifying finder parameter arguments in job definitions. This JobItem subclass directly maps to the finder_parameter jil keyword.

Example: Sample Code for Using the FinderParameterItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;
    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_J2EE_JMS_ENTITY_BEAN);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_BEAN_NAME, "myBean");
        insertReq.set(AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,
        "myContext");
        insertReq.set(AsJobConstants.KEYWORD_METHOD_NAME, "myMethod");
        insertReq.set(AsJobConstants.KEYWORD_PROVIDER_URL, "myUrl");
        insertReq.set(AsJobConstants.KEYWORD_OPERATION_TYPE,
        AsJobConstants.VALUE__OPERATION_TYPE__UPDATE);
        // Set up the finder_parameter part of the job definition and
        // attach it to the base job definition.
        FinderParameterItem fpi = new FinderParameterItem();
        fpi.set(AsJobConstants.KEYWORD_FINDER_PARAMETER, "param1=p1");
        fpi.set(AsJobConstants.KEYWORD_FINDER_PARAMETER, "param2=p2");
        fpi.set(AsJobConstants.KEYWORD_FINDER_PARAMETER, "param3=p3");
        insertReq.add(fpi);
        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_BEAN_NAME,  
        AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,  
        AsJobConstants.KEYWORD_METHOD_NAME,  
        AsJobConstants.KEYWORD_PROVIDER_URL,  
        AsJobConstants.KEYWORD_OPERATION_TYPE,  
        AsJobConstants.KEYWORD_FINDER_PARAMETER  
    };  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();
```

```
// Process job level properties.  
System.out.println("----- Begin Response -----");  
for(int i=0; i<rspProperties.length; ++i) {  
    String keyword =  
    jobProperties.getKeywordFromConstant(rspProperties[i]);  
    switch(getRsp.getAttributeType(rspProperties[i])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
            System.out.println(keyword + ": " +  
            getRsp.getString(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
            System.out.println(keyword + ": " +  
            getRsp.getInt(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
            System.out.println(keyword + ": " +  
            getRsp.getBoolean(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_LONG:  
            System.out.println(keyword + ": " +  
            getRsp.getLong(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:  
            System.out.println("\n----- Begin " + keyword + " JobItem  
-----");  
            JobItem ji = getRsp.getJobItem(rspProperties[i]);  
            int itemProperties[] = ji.getAttributes();
```

```
// Process job item level properties.  
for(int j=0; j<itemProperties.length; ++j) {  
    System.out.print(keyword + ": ");  
  
    switch(jobProperties.getKeywordDataType(itemProperties[j])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
  
            System.out.println(ji.getString(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
  
            System.out.println(ji.getInt(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
  
            System.out.println(ji.getBool(itemProperties[j]));  
            break;  
    }  
    System.out.println("----- End " + keyword + " JobItem  
-----\n");  
    break;  
}  
}  
System.out.println("----- End Response -----");  
}  
} catch (AsInitializationException e) {  
    e.printStackTrace();  
} catch (AsException e) {  
    e.printStackTrace();  
}  
}  
}
```

Sample output

```
----- Begin Response -----  
job_name: finder_parm_sample  
alarm_if_fail: true  
    machine: localhost  
    bean_name: myBean  
    initial_context_factory: myContext  
    method_name: myMethod  
    provider_url: myUrl  
    operation_type: 2
```

```
----- Begin finder_parameter JobItem -----
finder_parameter: param1=p1
finder_parameter: param2=p2
finder_parameter: param3=p3
----- End finder_parameter JobItem -----

----- End Response -----
```

[FinderParameterItem](#)

This constructor has the following syntax:

```
public FinderParameterItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- [add](#)
- [getAttributes](#)
- [getBool](#)
- [getInt](#)
- [getItemType](#)
- [getJobItem](#)
- [getLoggableItem](#)
- [getString](#)
- [hasSubItems](#)
- [set\(int keyword, boolean value\)](#)
- [set\(int keyword, int value\)](#)
- [set\(int keyword, java.lang.String value\)](#)
- [toString](#)

FtpCommandItem

extends

JobItem

FtpCommandItem is the container class for specifying FTP command arguments in job definitions. This JobItem subclass directly maps to the jil keyword ftp_command.

Example: Sample Code for Using the FtpCommandItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_FTP);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_FTP_COMPRESSION, 9);
        insertReq.set(AsJobConstants.KEYWORD_FTP_LOCAL_NAME, "myLocal");
        insertReq.set(AsJobConstants.KEYWORD_FTP_REMOTE_NAME, "myRemote");
        insertReq.set(AsJobConstants.KEYWORD_FTP_SERVER_NAME, "myServer");
        insertReq.set(AsJobConstants.KEYWORD_FTP_TRANSFER_DIRECTION,
        AsJobConstants.VALUE__FTP_DIRECTION__UPLOAD);
        insertReq.set(AsJobConstants.KEYWORD_FTP_TRANSFER_TYPE,
        AsJobConstants.VALUE__FTP_TRANSFER_TYPE__BINARY);

        // Set up the ftp_command part of the job definition and
        // attach it to the base job definition.
        FtpCommandItem fci = new FtpCommandItem();
        fci.set(AsJobConstants.KEYWORD_FTP_COMMAND, "cd public");
        fci.set(AsJobConstants.KEYWORD_FTP_COMMAND, "cd client");
        fci.set(AsJobConstants.KEYWORD_FTP_COMMAND, "ls");
        insertReq.add(fci);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_FTP_COMPRESSION,  
        AsJobConstants.KEYWORD_FTP_LOCAL_NAME,  
        AsJobConstants.KEYWORD_FTP_REMOTE_NAME,  
        AsJobConstants.KEYWORD_FTP_SERVER_NAME,  
        AsJobConstants.KEYWORD_FTP_TRANSFER_DIRECTION,  
        AsJobConstants.KEYWORD_FTP_TRANSFER_TYPE,  
        AsJobConstants.KEYWORD_FTP_COMMAND  
    };  
  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();  
  
        // Process job level properties.  
        System.out.println("----- Begin Response -----");  
        for(int i=0; i<rspProperties.length; ++i) {  
            String keyword =  
                jobProperties.getKeywordFromConstant(rspProperties[i]);  
            switch(getRsp.getAttributeType(rspProperties[i])) {  
                case AsConstants.ATTRIBUTE_TYPE_STRING:  
                    System.out.println(keyword + ": " +  
                        getRsp.getString(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_INT:  
                    System.out.println(keyword + ": " +  
                        getRsp.getInt(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                    System.out.println(keyword + ": " +  
                        getRsp.getBoolean(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_LONG:  
                    System.out.println(keyword + ": " +  
                        getRsp.getLong(rspProperties[i]));  
                    break;  
            }  
        }  
    }  
}
```

```

        System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
        System.out.println("\n----- Begin " + keyword + " JobItem
-----");
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {
            System.out.print(keyword + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
    case AsConstants.ATTRIBUTE_TYPE_STRING:
        System.out.println(ji.getString(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_INT:
        System.out.println(ji.getInt(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
        System.out.println(ji.getBool(itemProperties[j]));
        break;
    }
}
System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}

```

Sample output

```

----- Begin Response -----
job_name: ftp_command_sample
alarm_if_fail: true
machine: localhost
ftp_compression: 9
ftp_local_name: myLocal

```

```
ftp_remote_name: myRemote
ftp_server_name: myServer
ftp_transfer_direction: 1
ftp_transfer_type: 2
----- Begin ftp_command JobItem -----
ftp_command: cd public
ftp_command: cd client
ftp_command: ls
----- End ftp_command JobItem -----

----- End Response -----
```

FtpCommandItem

This constructor has the following syntax:

```
public FtpCommandItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- `add`
- `getAttributes`
- `getBool`
- `getInt`
- `getItemType`
- `getJobItem`
- `getLoggableItem`
- `getString`
- `hasSubItems`
- `set(int keyword, boolean value)`
- `set(int keyword, int value)`
- `set(int keyword, java.lang.String value)`
- `toString`

GetAdapterJobStats

GetAdapterJobStats retrieves the statuses of CA Workload Automation AE adapter jobs. GetAdapterJobStats is used by the autostatad command.

GetAdapterJobStats verifies list security for the AUTOSTAT security resource. If that passes, the API does not verify security for each job. Otherwise, the API verifies read access for each job. If read access verification fails, GetAdapterJobStats returns an empty response (zero-length job name).

Example: Sample Code for Executing the GetAdapterJobStats API

```
void sample(AsApi appServer, String jobName) {

    // Set up the request.
    IGetAdapterJobStatsReq req = new GetAdapterJobStatsReq();
    req.setRequest(jobName);

    // Execute the request, check for errors, and display the results.
    try {
        IApiResponseSet stats = req.execute(appServer);

        while(stats.hasNext()) {
            GetAdapterJobStatsRsp rsp = (GetAdapterJobStatsRsp) stats.next();
            System.out.println("\nJob Name: " + rsp.getJobName() +
                "Job Status: " + rsp.getJobStatus() +
                "Job Time: " + Tools.getStringFromDate(rsp.getTime()));
        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetAdapterJobStatsReq

extends

Cat2Request

implements

IGetAdapterJobStatsReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The `setRequest` method in the `GetAdapterJobStatsReq` class specifies the name of an adapter job for which to retrieve the status. This method has the following syntax:

```
void setRequest(java.lang.String name)
```

name

Specifies the name of the adapter job whose status to retrieve. Wildcards are accepted, for example, oap%.

GetAdapterJobStatsRsp

extends

ApiResponse

implements

IGetAdapterJobStatsRsp

The `GetAdapterJobStatsRsp` class is an individual response that is returned when iterating through a `GetAdapterJobStatsRspSet`.

getJobName

The `getJobName` method in the `GetAdapterJobStatsRsp` class returns the name of the adapter job for which status was requested. This method has the following syntax:

```
java.lang.String getJobName()
```

getJobStatus

The `getJobStatus` method in the `GetAdapaterJobStatsRsp` class returns the status of the adapter job for which status was requested. This method has the following syntax:

```
java.lang.String getJobStatus()
```

getTime

The `getTime` method in the `GetAdapterJobStatsRsp` class returns the time that the status was set for the specified adapter job. This method has the following syntax:

```
java.util.Date getTime()
```

GetAdapterJobStatsRspSet

extends

ApiResponseSet

The GetAdapterJobStatsRspSet class is a response set produced from a get adapter job status request. This set returns response objects of type GetAdapterJobStatsRsp.

Iterating over this set can produce the AsApiGeneralException exception.

GetAdapterJobStatsRspSet

The GetAdapterJobStatsRspSet constructor has the following syntax:

```
public GetAdapterJobStatsRspSet(com.ca.autosys.services.JResponseQ responseQ)  
responseQ
```

Defines the raw ResponseQ object.

The GetAdapterJobStatsRspSet constructor throws java.lang.IllegalArgumentException if a null response is given.

GetCalendarDays

GetCalendarDays retrieves all the dates in the requested standard calendar.
GetCalendarDays is used by autocal_asc.

GetCalendarDays checks for read access to the requested calendar and then returns each date.

Example: Sample Code for Executing the GetCalendarDays API

```
private void sample(AsApi api, String calendarName) {
    // Set up the request.
    IGetCalendarDaysReq req = new GetCalendarDaysReq();
    req.setRequest(calendarName);

    // Execute the request, check for errors, display the results.
    try {
        IApiResponseSet days = req.execute(api);

        // Days returned, show header.
        if(days.hasNext()) {
            System.out.println("\nDates in calendar \"\" + calendarName + "\"":\n");
        }

        // Loop and process the days.
        while(days.hasNext()) {
            GetCalendarDaysRsp rsp = (GetCalendarDaysRsp) days.next();
            System.out.println("\t" +
Tools.getStringFromDate(rsp.getCalendarDate()));
        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsObjectDoesNotExistException noObjEx) {
        System.out.println("AsObjectDoesNotExistException: " +
noObjEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetCalendarDaysReq

extends

Cat2Request

implements

IGetCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have authority to access the calendar.
- AsObjectDoesNotExistException - The specified calendar does not exist.

setRequest

The setRequest method in the GetCalendarDaysReq class specifies the calendar name to use in the request. This method has the following syntax:

```
void setRequest(java.lang.String name)
```

name

Specifies the calendar name to retrieve. Wildcards are not supported.

GetCalendarDaysRsp

extends

ApiResponse

implements

IGetCalendarDaysRsp

The GetCalendarDaysRsp class defines an individual response that is returned when iterating through a GetCalendarDaysRspSet.

getCalendarDate

The getCalendarDate method in the GetCalendarDaysRsp class returns the dates in the calendar named in the setRequest method in the GetCalendarDaysReq class. This method has the following syntax:

```
java.lang.Date getCalendarDate()
```

GetCalendarNames

GetCalendarNames requests and returns a list of all defined calendar names from the CA Workload Automation AE database.

Example: Sample Code for Executing the GetCalendarNames API

```
private void sample( AsApi api ) {
    GetCalendarNamesReq req = new GetCalendarNamesReq();

    IApiResponseSet calendarList;
    try {
        calendarList = req.execute( api );

        System.out.println("\nDefined Calendars:\n");

        while ( calendarList.hasNext() ) {
            GetCalendarNamesRsp rsp = ( GetCalendarNamesRsp ) calendarList.next();
            System.out.println(rsp.getCalendarName());
        }
    } catch ( AsInitializationException initEx ) {
        System.out.println("Caught Initialization Exception: " +
initEx.getMessage());
    } catch ( AsException asEx ) {
        System.out.println("Caught AutoSys Exception: " + asEx.getMessage());
    }
}
```

GetCalendarNamesReq

extends

Cat2Request

implements

IGetCalendarNamesReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have authority to list the calendars.

GetCalendarNamesRsp

extends
 ApiResponse
implements
 IGetCalendarNamesRsp

getCalendarName

The getCalendarName method in the GetCalendarNamesRsp class returns the calendar name in the response. This method has the following syntax:

```
java.lang.String getCalendarName()
```

GetCalendarNamesRspSet

extends
 ApiResponseSet

GetCalendarNamesRspSet

The GetCalendarNamesRspSet constructor has the following syntax:

```
public GetCalendarNamesRspSet(com.ca.autosys.services.JResponseQ responseQ)  
responseQ
```

Defines the raw responseQ object.

The GetCalendarNamesRspSet constructor throws java.lang.IllegalArgumentException if the responseQ is null.

GetCycleCalendarDays

GetCycleCalendarDays returns the definition of a cycle calendar from the CA Workload Automation AE database.

Example: Sample Code for Executing the GetCycleCalendarDays API

```
private void sample(AsApi api, String cycle) {
    GetCycleCalendarDaysReq req = new GetCycleCalendarDaysReq();
    req.setRequest(cycle);

    IApiResponseSet set;
    try {
        set = req.execute(api);

        System.out.println("Cycle Calendar: " + cycle);
        while(set.hasNext()) {
            GetCycleCalendarDaysRsp rsp = (GetCycleCalendarDaysRsp) set.next();
            System.out.println();
            System.out.println("Start Date: " +
Tools.getStringFromDate(rsp.getStartDate()));
            System.out.println("End Date : " +
Tools.getStringFromDate(rsp.getEndDate()));
            System.out.println();
        }
    }

    } catch (AsInitializationException initEx) {
        System.out.println("Caught Initialization Exception. " +
initEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("Caught AutoSys Exception. " + asEx.getMessage());
    }
}
```

GetCycleCalendarDaysReq

extends

Cat2Request

implements

IGetCycleCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have the authority to retrieve the cycle calendar.

setRequest

The `setRequest` method in the `GetCycleCalendarDaysReq` class configures the `GetCycleCalendarDaysReq` request. This method has the following syntax:

```
void setRequest(java.lang.String cycleName)
```

cycleName

Specifies the cycle name to retrieve.

GetCycleCalendarDaysRsp

extends

`ApiResponse`

implements

`IGetCycleCalendarDaysRsp`

getCycCalName

The `getCycCalName` method in the `GetCycleCalendarDaysRsp` class returns the cycle calendar name. This method has the following syntax:

```
java.lang.String getCycCalName()
```

getStartDate

The `getStartDate` method in the `GetCycleCalendarDaysRsp` class returns the start date of the cycle. This method has the following syntax:

```
java.util.Date getStartDate()
```

getEndDate

The `getEndDate` method in the `GetCycleCalendarDaysRsp` class returns the end date of the cycle. This method has the following syntax:

```
java.util.Date getEndDate()
```

GetExtendedCalendarDays

GetExtendedCalendarDays retrieves the days that are defined in an extended calendar.

Example: Sample Code for Executing the GetExtendedCalendarDays API

```
void Sample( AsApi api, String server, int port, String calendar ) {
    // set up the request
    GetExtendedCalendarDaysReq req = new GetExtendedCalendarDaysReq();
    req.setRequest( calendar );

    // execute the request and check return
    try {
        req.execute( api );
    } catch ( AsGeneralErrorException genEx ) {
        System.out.println( "General Exception Caught: "
            + genEx.getMessage().trim() );
    } catch ( AsSecurityException secEx ) {
        System.out.println( "Security Exception Caught: "
            + secEx.getMessage().trim() );
    } catch ( AsObjectDoesNotExistException objEx ) {
        System.out.println( "ObjectDoesNotExist Exception Caught: "
            + objEx.getMessage().trim() );
    } catch ( AsException asEx ) {
        System.out.println( "AutoSys Exception Caught: "
            + asEx.getMessage().trim() );
    }
}
```

GetExtendedCalendarDaysReq

extends

Cat1Request

implements

IGetExtendedCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The user does not have authority to get the dates from the calendar.
- AsObjectDoesNotExistException - The requested calendar does not exist.

setRequest

The setRequest method in the GetExtendedCalendarDaysReq class sets the get extended calendar days request. This method has the following syntax:

```
void setRequest(java.lang.String calendarName)
```

calendarName

Specifies the extended calendar name.

GetExtendedCalendarDaysRsp

extends

ApiResponse

implements

IGetExtendedCalendarDaysRsp

getAdjustment

The getAdjustment method in the GetExtendedCalendarDaysRsp class retrieves the adjustment used to add or subtract days from the condition. This method has the following syntax:

```
int getAdjustment()
```

getCalendarID

The getCalendarID method in the GetExtendedCalendarDaysRsp class retrieves the calendar ID. This method has the following syntax:

```
int getCalendarID()
```

getCalendarName

The getCalendarName method in the GetExtendedCalendarDaysRsp class retrieves the name of the calendar. This method has the following syntax:

```
java.lang.String getCalendarName()
```

getCondition

The getCondition method in the GetExtendedCalendarDaysRsp class retrieves the condition used to decide holiday and cycle usage. This method has the following syntax:

```
java.lang.String getCondition()
```

getCycleCalName

The `getCycleCalName` method in the `GetExtendedCalendarDaysRsp` class retrieves the cycle calendar name. This method has the following syntax:

```
java.lang.String getCycleCalName()
```

getHolidayAction

The `getHolidayAction` method in the `GetExtendedCalendarDaysRsp` class retrieves the holiday action. This method has the following syntax:

```
int getHolidayAction()
```

getHolidayCalName

The `getHolidayCalName` method in the `GetExtendedCalendarDaysRsp` class retrieves the holiday calendar name. This method has the following syntax:

```
java.lang.String getHolidayCalName()
```

getWeekendAction

The `getWeekendAction` method in the `GetExtendedCalendarDaysRsp` class retrieves the weekend action. This method has the following syntax:

```
int getWeekendAction()
```

getWorkDays

The `getWorkDays` method in the `GetExtendedCalendarDaysRsp` class retrieves the workdays in the calendar. This method has the following syntax:

```
int[] getWorkDays()
```

GetExternalInstances

GetExternalInstances retrieves instance information for all external instances that match the given pattern.

Example: Sample Code for Executing the GetExternalInstances API

```
private void sample(AsApi api, String instance) {
    GetExternalInstancesReq req = new GetExternalInstancesReq();
    req.setRequest(instance);

    try {
        ApiResponseSet rspSet = (ApiResponseSet)req.execute(api);

        while(rspSet.hasNext()) {
            GetExternalInstancesRsp rsp = (GetExternalInstancesRsp)rspSet.next();
            System.out.println("\nInstance ID: " + rsp.getInstanceId());
            System.out.println("Instance Type: " + rsp.getInstanceType());
            System.out.println("Connect Info: " + rsp.getConnectInfo() + "\n");
        }
    } catch (AsGeneralErrorException ge) {
        System.out.println("AsGeneralErrorException: " + ge.getMessage());
    } catch (AsException e) {
        System.out.println("AsException: " + e.getMessage());
    }
}
```

GetExternalInstancesReq

extends
Cat2Request
implements
IGetExternalInstancesReq

Execution of this request can result in AsGeneralErrorException being thrown.

GetExternalInstancesReq

The GetExternalInstancesReq constructor creates a GetExternalInstancesReq object.
This constructor has the following syntax:

```
public GetExternalInstancesReq()
```

setRequest

The setRequest method in the GetExternalInstancesReq class configures the request.
This method has the following syntax:

```
public void setRequest(java.lang.String instanceId)
```

instanceId

Specifies the ID pattern, for example, ACE, P%, %.

GetExternalInstancesRsp

extends
 ApiResponse
implements
 IGetExternalInstancesRsp

GetExternalInstancesRsp

The GetExternalInstancesRsp constructor has the following syntax:

```
public GetExternalInstancesRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

The GetExternalInstancesRsp constructor throws java.lang.IllegalArgumentException if a null response is given.

getInstanceType

The getInstanceType method in the GetExternalInstancesRsp class returns the instance type. This method has the following syntax:

```
public java.lang.String getInstanceType()
```

getInstanceTypeAsInt

The getInstanceTypeAsInt method in the GetExternalInstancesRsp class has the following syntax:

```
public int getInstanceTypeAsInt()
```

getInstanceId

The getInstanceId method in the GetExternalInstancesRsp class returns the instance ID. This method has the following syntax:

```
public java.lang.String getInstanceId()
```

getConnectInfo

The getConnectInfo method in the GetExternalInstancesRsp class returns the instance connect information. This method has the following syntax:

```
public java.lang.String getConnectInfo()
```

getMachine

The getMachine method in the GetExternalInstancesRsp class returns the instance machine names. This method has the following syntax:

```
public java.lang.String getMachine()
```

getManager

The getManager method in the GetExternalInstancesRsp class returns the instance manager name. This method has the following syntax:

```
public java.lang.String getManager()
```

getPort

The getPort method in the GetExternalInstancesRsp class returns the instance port number. This method has the following syntax:

```
public int getPort()
```

getCryptType

The getCryptType method in the GetExternalInstancesRsp class returns the instance encryption type. This method has the following syntax:

```
public int getCryptType()
```

GetGblobList

GetGblobList requests listings of global blobs from the CA Workload Automation AE database.

Example: Sample Code for Executing the GetGblobList API

```
private void sample(AsApi api) {
    // Set up request and response set.
    IApiResponseSet list = null;
    GetGblobListReq req = new GetGblobListReq();

    // Request information on ALL global blobs.
    req.setRequest("ALL");

    // Execute the request, display results, and check for errors.
    try {
        list = req.execute(api);
```

```

        while(list.hasNext()) {
            GetGblobListRsp rsp = (GetGblobListRsp) list.next();
            System.out.println("\nBlob Detail:\n\n" +
                "Blob Name:      " + rsp.getGblobName() + "\n" +
                "Blob Creator:   " + rsp.getCreationUser() + "\n" +
                "Blob Create Time: " + rsp.getCreationTime() + "\n" +
                "Blob Modifier:   " + rsp.getModificationUser() + "\n" +
                "Blob Modify Time: " + rsp.getModificationTime() + "\n");
        }
        System.out.println("\n");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " +
            genEx.getMessage());
    } catch (AsObjectDoesNotExistException noObjEx) {
        System.out.println("AsObjectDoesNotExistException: " +
            noObjEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " +
            asEx.getMessage());
    }
}
}

```

GetGblobListReq

extends

Cat2Request

implements

IGetGblobListReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsObjectDoesNotExistException - An invalid global blob name was passed.

setRequest

The setRequest method in the GetGblobListReq class configures the request for execution. This method has the following syntax:

public void setRequest(java.lang.String *blobName*)

blobName

Specifies the name of the global blob to list. Specify ALL to return all global blobs.

GetGblistRsp

extends
 ApiResponse
implements
 IGblistRsp

getGblistName

The getGblistName method in the GetGblistRsp class returns the global blob name. This method has the following syntax:

```
public java.lang.String getGblistName()
```

getCreationUser

The getCreationUser method in the GetGblistRsp class returns the name of the user that created the blob. This method has the following syntax:

```
public java.lang.String getCreationUser()
```

getCreationTime

The getCreationTime method in the GetGblistRsp class returns the creation time for the blob. This method has the following syntax:

```
public java.lang.String getCreationTime()
```

getModificationUser

The getModificationUser method in the GetGblistRsp class returns the the name of the user that modified the blob. This method has the following syntax:

```
public java.lang.String getModificationUser()
```

getModificationTime

The getModificationTime method in the GetGblistRsp class returns the modification time for the blob. This method has the following syntax:

```
public java.lang.String getModificationTime()
```

GetGeneratedExtendedCalendarDays

GetGeneratedExtendedCalendarDays returns the dates that would be generated for the given extended calendar definition.

Example: Sample Code for Executing the GetGeneratedExtendedCalendarDays API

```
private void sample(AsApi api, String calendarName) {
    int[] workDays = {
        AsConstants.DAY_MONDAY,
        AsConstants.DAY_WEDNESDAY,
        AsConstants.DAY_FRIDAY,
        AsConstants.DAY_SUNDAY
    };
    String holidayCal = "holiday_calendar";
    String cycleCal = "cycle_calendar";
    int holidayAction = AsConstantsCALENDAR_ACTION_NEXT_WORK_DAY;
    int weekendAction = AsConstantsCALENDAR_ACTION_PREVIOUS_WORK_DAY;
    int conditionAdjustment = 0;
    String conditionKeyword = "WORKDAYS";
    GetGeneratedExtendedCalendarDaysReq req = new
    GetGeneratedExtendedCalendarDaysReq();
    req.setRequest(calendarName, workDays, holidayCal, cycleCal, holidayAction,
weekendAction, conditionAdjustment, conditionKeyword);

    IApiResponseSet dateInCalendar;
    try {
        dateInCalendar = req.execute(api);

        while (dateInCalendar.hasNext()) {
            GetGeneratedExtendedCalendarDaysRsp rsp =
(GetGeneratedExtendedCalendarDaysRsp) dateInCalendar.next();

System.out.println(GetGeneratedExtendedCalendarDaysRsp.getStringFromDate(rsp.getCalendarDate()));
        }
    } catch (AsInitializationException e) {
        System.out.println("Initialization Exception Caught: " + e.getMessage());
    } catch (AsException asEx) {
        System.out.println("AutoSys Exception Caught: " + asEx.getMessage());
    }
}
```

GetGeneratedExtendedCalendarDaysReq

extends

Cat2Request

implements

IGetGeneratedExtendedCalendarDaysReq

Execution of this request can result in AsInitializationException being thrown.

setRequest

The setRequest method in the GetGeneratedExtendedCalendarDaysReq class configures the request to get the dates that would be generated for the given extended calendar. This method has the following syntax:

```
void setRequest(java.lang.String calendarName, int[] workDays, java.lang.String holidayCalendar, java.lang.String cycleCalendar, int holidayAction, int weekendAction, int conditionAdjustment, java.lang.String conditionKeyword)
```

calendarName

Specifies the name of the calendar.

workDays

Specifies the requested workdays. Valid workdays are:

- AsConstants.DAY_SUNDAY - Set Sunday as a workday.
- AsConstants.DAY_MONDAY - Set Monday as a workday.
- AsConstants.DAY_TUESDAY - Set Tuesday as a workday.
- AsConstants.DAY_WEDNESDAY - Set Wednesday as a workday.
- AsConstants.DAY_THURSDAY - Set Thursday as a workday.
- AsConstants.DAY_FRIDAY - Set Friday as a workday.
- AsConstants.DAY_SATURDAY - Set Saturday as a workday.

holidayCalendar

Specifies the name of the holiday calendar to use.

cycleCalendar

Specifies the name of the cycle calendar to use.

holidayAction

Specifies the action to take when the work falls on a holiday.

- AsConstants.CALENDAR_ACTION_DEFAULT - Do not schedule the job.
- AsConstants.CALENDAR_ACTION_ONLY - Schedule only on a holiday.
- AsConstants.CALENDAR_ACTION_SCHEDULE - Schedule always.
- AsConstants.CALENDAR_ACTION_NEXT_DAY - Schedule on the next day (day could be a holiday).
- AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY - Schedule on the next workday.
- AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY - Schedule on the previous workday.

weekendAction

Specifies the action to take when the work falls on a non-workday.

- AsConstants.CALENDAR_ACTION_DEFAULT - Schedule always.
- AsConstants.CALENDAR_ACTION_ONLY - Schedule only on a non-workday.
- AsConstants.CALENDAR_ACTION_NEXT_DAY - Schedule on the next day (day could be a non-workday).
- AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY - Schedule on the next workday.
- AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY - Schedule on the previous workday.

conditionAdjustment

Specifies the negative or positive number defining the days before or after the condition setting.

conditionKeyword

Specifies the date keyword to use as a scheduling condition.

Note: For more information about condition keywords, see the *Reference Guide*.

GetGeneratedExtendedCalendarDaysRsp

extends

ApiResponse

implements

IGetGeneratedExtendedCalendarDaysRsp

getCalendarDate

The `getCalendarDate` method in the `GetGeneratedExtendedCalendarDaysRsp` class retrieves a date object representing a single date defined in the extended calendar. This method has the following syntax:

```
java.util.Date getCalendarDate()
```

GetGeneratedExtendedCalendarDaysRspSet

extends

`ApiResponseSet`

GetGeneratedExtendedCalendarDaysRspSet

The `GetGeneratedExtendedCalendarDaysRspSet` constructor has the following syntax:

```
public GetGeneratedExtendedCalendarDaysRspSet(com.ca.autosys.services.JResponseQ  
responseQ)
```

responseQ

Defines the raw `ResponseQ` object.

The `GetGeneratedExtendedCalendarDaysRspSet` constructor throws `java.lang.IllegalArgumentException` if the response is null.

GetGlobals

GetGlobals retrieves global variables and associated attributes based on the requested name pattern. Either ALL or % retrieves all defined global variables. GetGlobals is used by autorep-G.

If the requesting user has appropriate access to the AUTOREP list security resource, GetGlobals does not verify security for each global variable returned. Otherwise, the API verifies read access for each global variable. If security is denied, GetGlobals returns an empty response (that is, the name and value fields will be zero-length strings).

Example: Sample Code for Executing the GetGlobals API

```
private void sample(AsApi appServer) {
    // Set up the request.
    IGetGlobalsReq req = new GetGlobalsReq();
    req.setRequest("ALL");

    // Set up response set.
    IApiResponseSet globals = null;

    // Execute the request, process responses, and check for errors.
    try {
        globals = req.execute(appServer);

        while(globals.hasNext()) {
            GetGlobalsRsp rsp = (GetGlobalsRsp) globals.next();

            System.out.println();
            System.out.println("Global Name: " + rsp.getName());
            System.out.println("Global Value: " + rsp.getValue());
            System.out.println("Global Modification Time: " + rsp.getModTime());
        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetGlobalsReq

extends
Cat2Request
implements
IGetGlobalsReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The setRequest method in the GetGlobalsReq class sets the name or pattern of the global variables to retrieve. This method has the following syntax:

```
void setRequest(java.lang.String name)
```

name

Specifies the name or pattern of the global variables to retrieve. When name is set to ALL or %, the API returns all global variables.

GetGlobalsRsp

extends
ApiResponse
implements
IGetGlobalsRsp

The GetGlobalsRsp class is an individual response that is returned when iterating through a GetGlobalRspSet. This response represents one global variable consisting of the name, value, and modification time.

getModTime

The getModTime method in the GetGlobalsRsp class returns the time at which the global variable was last changed. This method has the following syntax:

```
java.util.Date getModTime()
```

getName

The getName method in the GetGlobalsRsp class returns the global variable name. This method has the following syntax:

```
java.lang.String getName()
```

getValue

The `getValue` method in the `GetGlobalsRsp` class returns the global variable value. This method has the following syntax:

```
java.lang.String getValue()
```

GetIntCodes

In CA Workload Automation AE, there are certain integer properties that have special meaning. These are commonly referred to as intcodes. An intcode has a type, integer value, and a string representation. An example is job status. The status of a job is an integer value, for example, 1. This indicates that the job is in RUNNING state. That knowledge is embodied in the intcode concept. In this example, the intcodes type is status; its integer value is 1; and its string representation is RUNNING.

These values are stored in the database, and are subject to change. The entire list of intcodes can be retrieved using this cat2 API.

The intent is to call `GetIntCodes` once and cache the values as a table. When an intcode is retrieved, the corresponding string representation of that integer can be retrieved.

Example: Sample Code for Executing the GetIntCodes API

```
private void sample(AsApi appServer) {
    GetIntCodesReq req = new GetIntCodesReq();
    req.setRequest();

    try {
        ApiResponseSet rspSet = (ApiResponseSet)req.execute(appServer);

        while(rspSet.hasNext()) {
            System.out.println("----- Begin Response -----");
            GetIntCodesRsp rsp = (GetIntCodesRsp)rspSet.next();
            System.out.println("Intcode text: " + rsp.getText());
            System.out.println("Intcode type: " + rsp.getType());
            System.out.println("Intcode value: " +
String.valueOf(rsp.getValue()));
            System.out.println("----- End Response -----\\n");
        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetIntCodesReq

extends
Cat2Request
implements
IGetIntCodesReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The setRequest method in the GetIntCodesReq class configures the request. This method has the following syntax:

```
void setRequest ()
```

GetIntCodesRsp

extends
ApiResponse
implements
IGetIntCodesRsp

The GetIntCodesRsp class is an individual response that is returned when iterating through a GetIntCodesRspSet.

getType

The getType method in the GetIntCodesRsp class returns the int code type. Possible values are:

- alarm
- event
- sound
- status

This method has the following syntax:

```
java.lang.String getType()
```

getText

The `getText` method in the `GetIntCodesRsp` class returns the int code text, for example, `FORKFAIL`, `KILLJOB`, or `ACTIVATED`. This method has the following syntax:

```
java.lang.String getText()
```

getValue

The `getValue` method in the `GetIntCodesRsp` class returns the int code value. This method has the following syntax:

```
int getValue()
```

GetJobLog

`GetJobLog` retrieves the job log for a specified CA Workload Automation AE job.

Example: Sample Code for Executing the GetJobLog API

```
private void sample(AsApi appServer, String jobName) {  
    GetJobLogReq req = new GetJobLogReq();  
    req.setRequest(IGetJobLogReq.AGENT, jobName, -1, -1);  
  
    try {  
        GetLogRsp rsp = (GetLogRsp)req.execute(appServer);  
        String log = rsp.getLog();  
        System.out.println(log);  
    } catch (AsGeneralErrorException ge) {  
        System.out.println("AsGeneralErrorException: " + ge.getMessage());  
    } catch (AsSecurityException se) {  
        System.out.println("AsSecurityException: " + se.getMessage());  
    } catch (AsObjectDoesNotExistException noObjEx) {  
        System.out.println("AsObjectDoesNotExistException: " +  
noObjEx.getMessage());  
    } catch (AsException e) {  
        System.out.println("AsException: " + e.getMessage());  
    }  
}
```

GetJobLogReq

extends

Cat1Request

implements

IGetJobLogReq

GetJobLogReq requests a specific job log from the configured server. Execution of this request produces a GetLogRsp.

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have authority to view this object.
- AsObjectDoesNotExistException - The specified jobname is invalid.

setRequest

The setRequest method in the GetJobLogReq class configures the request to retrieve a job log. This method has the following syntax:

```
void setRequest(int logType, java.lang.String jobName, int runNumber, int nTry)  
logType
```

Specifies the log type to retrieve. Possible values are:

STANDARD_OUT

Specifies the standard out file.

STANDARD_ERR

Specifies the standard error file.

AGENT

Specifies the remote Agent file corresponding to the given run number and nTry.

AGENT_PROFILE

Specifies the profile used to source this job.

jobName

Specifies the name of the job for which to retrieve the log.

runNumber

Specifies the run number.

nTry

Specifies the number of job run retries.

GetJobLogRsp

extends
 ApiResponse
implements
 IGetLogRsp

getLog

The getLog method in the GetLogRsp class returns the requested log contents. This method has the following syntax:

```
public java.lang.String getLog()
```

GetJobRunsWithFilter

GetJobRunsWithFilter retrieves job runs information with filtering.

Example: Sample Code for Executing the GetJobRunsWithFilter API

```
private void sample(AsApi api) {
    int statusCode = 0;

    try {
        // Get the status code for the "SUCCESS" status.
        statusCode = Tools.getIntCode(api, Tools.INTCODE_TYPE_STATUS, "SUCCESS");
    } catch (AsException e) {    // Error getting status code...
        e.printStackTrace();
    }

    // Set up the filter to request records for all jobnames
    // with the status of "SUCCESS".
    JobRunsFilterAnd filter = new JobRunsFilterAnd();
    filter.add(new JobRunsFilterString(IJobRunsFilterStringFLT_JOB_NAME, "%"));
    filter.add(new JobRunsFilterInt(IJobRunsFilterIntFLT_EVENT_STATUS,
        statusCode, IJobRunsFilterIntEQUAL));

    // Set up the requested attributes.
    int[] attributes = {
        IJobRunAttributesATTR_JOB_NAME,
        IJobRunAttributesATTR_EVENT,
        IJobRunAttributesATTR_EVENT_TEXT,
        IJobRunAttributesATTR_EVENT_TIME,
        IJobRunAttributesATTR_EVENT_STATUS
    };

    // Set up the request and response set.
    IGetJobRunsWithFilterReq req = new GetJobRunsWithFilterReq();
    req.setRequest(filter, attributes);
    IApiResponseSet set;

    // Try and execute the request, check for errors, print the data.
    try {
        set = req.execute(api);

        // Loop and print data.
        while(set.hasNext()) {
            System.out.println("\n----- Begin Response -----");
            IGetJobRunsWithFilterRsp rsp = (GetJobRunsWithFilterRsp)set.next();

            // Returned attributes.
            int[] rspAttributes = rsp.getAttributes();
    
```

```
// Loop and display the returned data.
for(int i=0; i<attributes.length; i++) {
    String attributeName = rsp.getAttributeName(rspAttributes[i]);
    int attributeType = rsp.getAttributeType(rspAttributes[i]);
    String attributeValue = "";

    switch(attributeType) {
        case AsApiConstants.AtChar:
            attributeValue =
                String.valueOf(rsp.getChar(rspAttributes[i]));
            break;
        case AsApiConstants.AtDate:
            attributeValue =
                String.valueOf(rsp.getDate(rspAttributes[i]));
            break;
        case AsApiConstants.AtInt:
            attributeValue =
                String.valueOf(rsp.getInt(rspAttributes[i]));
            break;
        case AsApiConstants.AtString:
            attributeValue =
                String.valueOf(rsp.getString(rspAttributes[i]));
            break;
    }
    System.out.println(attributeName + " = " + attributeValue);
}
System.out.println("----- End Response -----\\n");
}

} catch (AsGeneralErrorException genEx) {
    System.out.println("AsGeneralErrorException: " + genEx.getMessage());
} catch (AsSecurityException secEx) {
    System.out.println("AsSecurityException: " + secEx.getMessage());
} catch (AsNoAttributesException noAttrEx) {
    System.out.println("AsNoAttributesException: " + noAttrEx.getMessage());
} catch (AsBadAttributesException badAttrEx) {
    System.out.println("AsBadAttributesException: " +
badAttrEx.getMessage());
} catch (AsBadFilterFieldsException badFilterEx) {
    System.out.println("AsBadFilterFieldsException: " +
badFilterEx.getMessage());
} catch (AsException asEx) {
    System.out.println("AsException: " + asEx.getMessage());
}
}
```

GetJobRunsWithFilterReq

extends

Cat2Request

implements

IGetJobRunsWithFilterReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - You do not have authority to process the request.
- AsNoAttributesException - No attributes were specified.
- AsBadAttributesException - Invalid attributes were specified.
- AsBadFilterFieldsException - Invalid filtering fields were specified.

setRequest

The setRequest method in the GetJobRunsWithFilterReq class sets up the request for execution. This method has the following syntax:

public void setRequest(IJobRunsFilter *filter*, int[] *attributes*)

filter

Defines the job runs set.

attributes

Specifies the attributes for the job. Possible values are:

- ATTR_EVENT: Event ID.
- ATTR_JOB_NAME: Job name.
- ATTR_GROUP: Group name.
- ATTR_APPLICATION: Application name.
- ATTR_JOB_TYPE: Job type
- ATTR_TIME_ZONE: Timezone reference.
- ATTR_TARGET_MACHINE: Target machine for job to run on.
- ATTR_JOID: Job ID number.
- ATTR_RUN_NUM: Job run number.
- ATTR_NTRY: Job retry number.
- ATTR_START_TIME: Job start time.
- ATTR_END_TIME: Job end time.
- ATTR_STATUS: Job ending status.

- ATTR_EXIT_CODE: Job exit code.
- ATTR_RUN_MACHINE: Machine where job ran.
- ATTR_SVC_DESK_STATUS: Service desk status.
- ATTR_SVC_DESK_HANDLE: Service desk contact ID.
- ATTR_OVERRIDE_NUM: Job override number.
- ATTR_BOX_JOID: ID of box job.
- ATTR_BOX_NAME: Name of box job.
- ATTR_EVENT_TIME: Time the event occurred.
- ATTR_QUE_STATUS: Dispatch submission status.
- ATTR_EVENT_TEXT: Text associated with the event.
- ATTR_EVENT_STATUS: Status of the event.
- ATTR_OWNER: Owner.
- ATTR_DESCRIPTION: Job Description.
- ATTR_EVENT_QUE_STAMP: Time the event was processed.
- ATTR_EXT_RUNTIME_WOBID: Internal Workload object id.
- ATTR_EXT_RUNTIME_JOBID: Internal job id.
- ATTR_EXT_RUNTIME_JOBLOGID: Internal Job log id.
- ATTR_EXT_RUNTIME_JOBNAME: Internal job name.
- ATTR_EXT_RUNTIME_JOBNO: Internal job number assigned by agent.
- ATTR_EVENT_START_TIME: Time event processing started.
- ATTR_EVENT_END_TIME: Time event processing ended.

GetJobRunsWithFilterRsp

extends
 ApiResponse
implements
 IGetJobRunsWithFilterRsp

getAttributeName

The `getAttributeName` method in the `GetJobRunsWithFilterRsp` class returns the string name resembling the column name for this attribute. This method has the following syntax:

```
public java.lang.String getAttributeName(int attribute)
```

attribute

Specifies the attribute.

getAttributes

The `getAttributes` method in the `GetJobRunsWithFilterRsp` class returns the list of attributes present in this response. This list should correspond to the attributes set in the initial request. This method has the following syntax:

```
public int[] getAttributes()
```

getAttributeType

The `getAttributeType` method in the `GetJobRunsWithFilterRsp` class returns the attribute type for the given attribute. This type returns the corresponding get function to call. This method has the following syntax:

```
public int getAttributeType(int attribute)
```

attribute

Specifies the attribute.

Possible returns and the corresponding get functions are:

- TYPE_INT: `getInt()`
- TYPE_STRING: `getString()`
- TYPE_BOOLEAN: `getBoolean()`
- TYPE_DATE: `getDate()`
- TYPE_INT_LIST: `getInts()`
- TYPE_TIMEOFDAY_LIST: `getTimesOfDay()`
- TYPE_CHKFILE_LIST: `getChkFiles()`

- TYPE_CHAR: getChar()

getChar

The getChar method in the GetJobRunsWithFilterRsp class returns the character value for the specified attribute. This method has the following syntax:

```
public char getChar(int attribute)
```

attribute

Specifies the job run attribute. The valid attribute is ATTR_JOB_TYPE.

getDate

The getDate method in the GetJobRunsWithFilterRsp class returns the date value for the specified attribute. This method has the following syntax:

```
public java.util.Date getDate(int attribute)
```

attribute

Specifies the job run attribute. Valid attributes are:

- ATTR_START_TIME: Job start time.
- ATTR_END_TIME: Job end time.
- ATTR_EVENT_TIME: Time the event occurred.
- ATTR_EVENT_QUE_STAMP: Time the event was processed.
- ATTR_REPLY_MSG: Message text associated with reply.
- ATTR_REPLY_RSP: Text of reply response.
- ATTR_STATUS_LONG: Long job status text.
- ATTR_STATUS_SHORT: Short job status text

getInt

The getInt method in the GetJobRunsWithFilterRsp class returns the integer value for the specified attribute. This method has the following syntax:

```
public int getInt(int attribute)
```

attribute

Specifies the job run attribute. Valid attributes are:

- ATTR_EVENT: Event ID.
- ATTR_JOB_TYPE: Job type
- ATTR_JOID: Job ID number.
- ATTR_RUN_NUM: Job run number.

- ATTR_NTRY: Job retry number.
- ATTR_STATUS: Job ending status.
- ATTR_EXIT_CODE: Job exit code.
- ATTR_SVC_DESK_STATUS: Service desk status.
- ATTR_BOX_JOID: ID of box job.
- ATTR_QUE_STATUS: Dispatch submission status.
- ATTR_EVENT_STATUS: Status of the event.
- ATTR_EVENT_ALARM: Alarm number associated with event

getString

The getString method in the GetJobRunsWithFilterRsp class returns the string value for the specified attribute. This method has the following syntax:

```
public java.lang.String getString(int attribute)
```

attribute

Specifies the job run attribute. Valid attributes are:

- ATTR_EVENT_TEXT: Text associated with the event.
- ATTR_JOB_NAME: Job name.
- ATTR_GROUP: Group name.
- ATTR_APPLICATION: Application name.
- ATTR_TIME_ZONE: Timezone reference.
- ATTR_TARGET_MACHINE: Target machine for job to run on.
- ATTR_RUN_MACHINE: Machine where job ran.
- ATTR_SVC_DESK_HANDLE: Service desk contact ID.
- ATTR_OVERRIDE_NUM: Job override number.
- ATTR_BOX_NAME: Name of box job.
- ATTR_OWNER: Owner.
- ATTR_DESCRIPTION: Job description

GetJobStatus

GetJobStatus retrieves the status of a job and checks for read access to the requested job. GetJobStatus is used by the product binary autostatus.

Example: Sample Code for Executing the GetJobStatus API

```
private void sample(AsApi appServer, String jobName) {
    // Set up the request object.
    IGetJobStatusReq req = new GetJobStatusReq();
    req.setRequest(jobName);

    // Execute the request, and check for errors.
    try {
        // Execute the request, and get the response.
        GetJobStatusRsp rsp = (GetJobStatusRsp) req.execute(appServer);

        // Display the job status int that was returned.
        //
        // If the status is desired in string format, the GetIntCodesReq api
        // can be called to map this status int to a string counterpart.
        System.out.println("Job status for job (" + jobName + ") is " +
                           rsp.getStatus());

    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetJobStatusReq

extends

Cat1Request

implements

IGetJobStatusReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have permission to view the status of the named job.

setRequest

The `setRequest` method in the `GetJobStatusReq` class configures the request for the job status. This method has the following syntax:

```
void setRequest(java.lang.String jobName)
```

jobName

Specifies the name of the job whose status to retrieve.

GetJobStatusRsp

extends

ApiResponse

implements

IGetJobStatusRsp

getStatus

The `getStatus` method in the `GetJobStatusRsp` class returns the status of the requested job (as an intcode). This method has the following syntax:

```
int getStatus()
```

GetJobsWithFilter

The filtering APIs are different from the other APIs in the SDK. They involve more classes which go together to allow for more complicated requests and responses. The following classes comprise the Job Filtering API:

- GetJobsWithFilterReq
- GetJobsWithFilterRsp
- JobFilterAnd
- JobFilterOr
- JobFilterDate
- JobFilterInt
- JobFilterString
- JobFilterBool
- JobFilterChar

Job filtering retrieves jobs from the CA Workload Automation AE database. It lets you specify complex filters based on simple field comparisons that can be logically connected and nested.

To accomplish this, the Job Filtering hierarchy uses the Composite Pattern. Following that pattern are five leaf objects. These objects are used to compare on a specific field of the given type (date, integer, string, Boolean, character):

- JobFilterDate
- JobFilterInt
- JobFilterString
- JobFilterBool
- JobFilterChar

The following two composite types combine leaf or composite objects logically:

- JobFilterAnd
- JobFilterOr

Example: Sample code for executing the GetJobsWithFilter API

```
private void sample(AsApi appServer) {
    GetJobsWithFilterReq req = new GetJobsWithFilterReq();
    JobFilterString filter =
        new JobFilterString(IJobFilterString.FLT_JOB_NAME, "sample%");
    int[] attributes =
        {GetJobsWithFilterReq.ATTR_JOB_NAME,
         GetJobsWithFilterReq.ATTR_APPLICATION,
         GetJobsWithFilterReq.ATTR_JOB_TYPE,
         GetJobsWithFilterReq.ATTR_GROUP
        };
    req.setRequest(filter, attributes);

    try {
        ApiResponseSet rspSet = (ApiResponseSet)req.execute(appServer);

        while(rspSet.hasNext()) {
            System.out.println("----- Begin Response -----");
            GetJobsWithFilterRsp rsp = (GetJobsWithFilterRsp)rspSet.next();
            int[] rspAttributes = rsp.getAttributes();
            String name, output;

            for(int i=0; i<rspAttributes.length; i++) {
                int type = rsp.getAttributeType(rspAttributes[i]);
                switch(type) {
                    case IFilterRsp.TYPE_INT:
                        name = rsp.getAttributeName(rspAttributes[i]);
                        output = name + " = " + rsp.getInt(rspAttributes[i]);
                        System.out.println(output);
                        break;
                    case IFilterRsp.TYPE_STRING:
                        name = rsp.getAttributeName(rspAttributes[i]);
                        output = name + " = " + rsp.getString(rspAttributes[i]);
                        System.out.println(output);
                        break;
                    case IFilterRsp.TYPE_DATE:
                        name = rsp.getAttributeName(rspAttributes[i]);
                        output = name + " = " +
Tools.getStringFromDate(rsp.getDate(rspAttributes[i]));
                        System.out.println(output);
                        break;
                    case IFilterRsp.TYPE_CHAR:
                        name = rsp.getAttributeName(rspAttributes[i]);
                        output = name + " = " + rsp.getChar(rspAttributes[i]);
                        System.out.println(output);
                        break;
                }
            }
            System.out.println("----- End Response -----");
        }
    }
```

```

        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsNoAttributesException noAttrEx) {
        System.out.println("AsNoAttributesException: " + noAttrEx.getMessage());
    } catch (AsBadAttributesException badAttrEx) {
        System.out.println("AsBadAttributesException: " +
badAttrEx.getMessage());
    } catch (AsBadFilterFieldsException badFiltEx) {
        System.out.println("AsBadFilterFieldsException: " +
badFiltEx.getMessage());
    } catch (AsException ex) {
        System.out.println("AsException: " + ex.getMessage());
    }
}

```

GetJobsWithFilterReq

extends

Cat2Request

implements

IGetJobsWithFilterReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsNoAttributesException—No attributes were specified.
- AsBadAttributesException—Invalid attributes were specified.
- AsBadFilterFieldsException—Invalid filter fields were specified.

GetJobsWithFilterReq retrieves a set of job attributes based on the provided job filter. It creates an SQL query based on the passed in filter object. It binds to and returns the variables specified in the request. Security is checked. If you do not have read permission, the jobs are not returned.

GetJobsWithFilterReq

The GetJobsWithFilterReq constructor has the following syntax:

```
public GetJobsWithFilterReq()
```

setRequest

This setRequest method in the GetJobsWithFilterReq class configures the request. This method has the following syntax:

```
void setRequest(IJobFilter filter, int[] attributes)
```

filter

Specifies the job set.

attributes

Specifies an array of integers representing attributes of the job set to be returned. The attribute values are contained in IJobFilterAttributes.

setRequest

This setRequest method in the GetJobsWithFilterReq class configures the request. This method has the following syntax:

```
public void setRequest(IJobFilter filter, int[] attributes, IJobFilterSort sort)
```

filter

Defines a constructed filter object that defines the job set. Passing a null pointer as a filter will return the requested attributes for all jobs.

attributes

Defines an array of integers representing attributes of the job set to be returned. IJobFilterAttributes defines all the possible attributes.

sort

Specifies an object that represents the required sort order of responses.

setMaxCount

The setMaxCount method in the GetJobsWithFilterReq class sets the maximum number of records to return. This method has the following syntax:

```
public void setMaxCount(int maxCount)
```

maxCount

Defines the maximum number of records to return.

setScrollDirection

The `setScrollDirection` method in the `GetJobsWithFilterReq` class sets the scroll direction of the returned items. This method has the following syntax:

```
public void setScrollDirection(int scrollDirection)  
scrollDirection
```

Specifies the direction to scroll in the returned record set. Valid values are the following:

- `SCROLL_FORWARD`
- `SCROLL_BACKWARD`

setSkipCount

The `setSkipCount` sets the number of returnable records to skip.

```
public void setSkipCount(int skipCount)  
skipCount
```

Defines the number of records to skip.

IJobFilterAttributes

The IJobFilterAttributes interface consists of the following constants, which encompasses all possible attributes you can retrieve with filtering:

- ATTR_ABAP_LANG
- ATTR_ABAP_NAME
- ATTR_ACTION
- ATTR_ADMINISTRATOR
- ATTR_AFTER_TIME
- ATTR_AGENT_NAME
- ATTR_ALARM
- ATTR_ALARM_IF_FAIL
- ATTR_ALARM_VERIF
- ATTR_ALERT
- ATTR_ALL_EVENTS
- ATTR_ALL_STATUS
- ATTR_AMOUNT
- ATTR_APPL_NTRY
- ATTR_APPLICATION
- ATTR_ARC_CLIENT
- ATTR_ARC_CONNECT
- ATTR_ARC_DATE
- ATTR_ARC_DOC_CLASS
- ATTR_ARC_DOC_TYPE
- ATTR_ARC_FORMAT
- ATTR_ARC_HOST_LINK
- ATTR_ARC_INFO
- ATTR_ARC_OBJ_NAME
- ATTR_ARC_OBJ_TYPE
- ATTR_ARC_OBJ_VARIANT
- ATTR_ARC_PARMS
- ATTR_ARC_PATH
- ATTR_ARC_PRINTER
- ATTR_ARC_PROTOCOL

- ATTR_ARC_REPORT
- ATTR_ARC_SERVICE
- ATTR_ARC_STORAGE
- ATTR_ARC_TEXT
- ATTR_ARC_USERID
- ATTR_ARC_VERSION
- ATTR_ARGS
- ATTR_ARGS_NUM
- ATTR_ARGLTYPE
- ATTR_AUTH_STRING
- ATTR_AUTHORIZATION
- ATTR_AUTO_DELETE
- ATTR_AUTO_HOLD
- ATTR_AVG_RUNTIME
- ATTR_BANNER_PAGE
- ATTR_BDC_ERR_RATE
- ATTR_BDC_EXT_LOG
- ATTR_BDC_PROC_RATE
- ATTR_BDC_SYSTEM
- ATTR_BEAN_NAME
- ATTR_BLIND_COPY
- ATTR_BLOB_INPUT
- ATTR_BLOB_TYPE
- ATTR_BOX_FAILURE
- ATTR_BOX_JOID
- ATTR_BOX_NAME
- ATTR_BOX_SUCCESS
- ATTR_BOX_TERMINATOR
- ATTR_BROADCAST_ADDRESS
- ATTR_CHARACTER_CODE
- ATTR_CHK_FILES
- ATTR_CHK_FILES_STRING
- ATTR_CLASS_NAME

- ATTR_COMMAND
- ATTR_COND_INDEX
- ATTR_COND_INDEX_POINTER
- ATTR_COND_JOB_AUTOSERV
- ATTR_COND_JOB_NAME
- ATTR_COND_LOOKBACK_SECS
- ATTR_COND_MODE
- ATTR_COND_OPERATOR
- ATTR_COND_TEST_GLOBAL_VALUE
- ATTR_COND_TYPE
- ATTR_COND_VALUE
- ATTR_CONDITION
- ATTR_CONDITION_CODE
- ATTR_CONNECT_STRING
- ATTR_CONNECTION_FACTORY
- ATTR_CONTINUOUS
- ATTR_COPIES
- ATTR_COPY
- ATTR_COPY_GENERATION
- ATTR_COPY_JCL
- ATTR_CPU_USAGE
- ATTR_CREATE_METHOD
- ATTR_CREATE_NAME
- ATTR_CREATE_PARAMETER
- ATTR_CREATE_PARAMETER_NUM
- ATTR_CREATE_STAMP
- ATTR_CREATE_USERID
- ATTR_CURRUN
- ATTR_DATASET
- ATTR_DATATYPE
- ATTR_DATE_CONDITIONS
- ATTR_DAYS_OF_WEEK
- ATTR_DAYS_OF_WEEK_STRING

- ATTR_DBTYPE
- ATTR_DEPT
- ATTR_DESCRIPTION
- ATTR_DEST
- ATTR_DESTINATION_FILE
- ATTR_DESTINATION_NAME
- ATTR_DISK_DRIVE
- ATTR_DISK_FORMAT
- ATTR_DISK_SPACE
- ATTR_ENCODING
- ATTR_ENCRYPTION_TYPE
- ATTR_ENDPOINT_URL
- ATTR_ENVVARS
- ATTR_EXCLUDE_CALENDAR
- ATTR_EXIT_CODE
- ATTR_EXPIRATION
- ATTR_EXPRESS
- ATTR_FACTOR
- ATTR_FAIL_CODES
- ATTR_FAILURE
- ATTR_FIELD_NAME
- ATTR_FILTER
- ATTR_FILTER_TYPE
- ATTR_FINDER_NAME
- ATTR_FINDER_PARAMETER
- ATTR_FOOTER
- ATTR_FORMAT
- ATTR_FREE
- ATTR_FTP_COMMAND
- ATTR_FTP_COMPRESSION
- ATTR_FTP_LOCAL_NAME
- ATTR_FTP_LOCAL_USER
- ATTR_FTP_REMOTE_NAME

- ATTR_FTP_SERVER_NAME
- ATTR_FTP_SERVER_PORT
- ATTR_FTP_TRANSFER_DIRECTION
- ATTR_FTP_TRANSFER_TYPE
- ATTR_FTP_USE_SSL
- ATTR_GLOBAL_IND
- ATTR_GROUP
- ATTR_HAS_NOTIFICATION
- ATTR_HAS_OVERRIDE
- ATTR_HAS_SERVICE_DESK
- ATTR_HEARTBEAT_ATTEMPTS
- ATTR_HEARTBEAT_FREQ
- ATTR_HEARTBEAT_INTERVAL
- ATTR_HIGH
- ATTR_HOST_PAGE
- ATTR_I5_ACTION
- ATTR_I5_CC_EXIT
- ATTR_I5_CURR_LIB
- ATTR_I5_JOB_DESC
- ATTR_I5_JOB_NAME
- ATTR_I5_JOB_QUEUE
- ATTR_I5_LDA
- ATTR_I5_LIB
- ATTR_I5_LIBRARY_LIST
- ATTR_I5_NAME
- ATTR_I5_OTHERS
- ATTR_I5_PARAMS
- ATTR_I5_PROCESS_PRIORITY
- ATTR_IGNORE
- ATTR_INDEX
- ATTR_INITIAL_CONTEXT_FACTORY
- ATTR_INSIDE_RANGE
- ATTR_INSTANCE

- ATTR_INTERACTIVE
- ATTR_INVOCATION_TYPE
- ATTR_IP_HOST
- ATTR_IP_PORT
- ATTR_IP_STATUS
- ATTR_J2EE_AUTHENTICATION_ORDER
- ATTR_J2EE_CONN_DOMAIN
- ATTR_J2EE_CONN_ORIGIN
- ATTR_J2EE_NO_GLOBAL_PROXY_DEFAULTS
- ATTR_J2EE_PARAMETER
- ATTR_J2EE_PROXY_DOMAIN
- ATTR_J2EE_PROXY_HOST
- ATTR_J2EE_PROXY_ORIGIN_HOST
- ATTR_J2EE_PROXY_PORT
- ATTR_J2EE_PROXY_USER
- ATTR_J2EE_USER
- ATTR_JC_PID
- ATTR_JCL_LIBRARY
- ATTR_JCL_MEMBER
- ATTR_JMX_PARAMETER
- ATTR_JMX_USER
- ATTR_JOB_CLASS
- ATTR_JOB_FILTER
- ATTR_JOB_LINEAGE
- ATTR_JOB_LOAD
- ATTR_JOB_NAME
- ATTR_JOB_TERMINATOR
- ATTR_JOB_TYPE
- ATTR_JOID
- ATTR_KEY_TO_AGENT
- ATTR_LAST_END
- ATTR_LAST_HEARTBEAT
- ATTR_LAST_START

- ATTR_LOW
- ATTR_LOWER_BOUNDARY
- ATTR_MAC_ADDRESS
- ATTR_MACH_STATUS
- ATTR_MACHINE
- ATTR_MACHINE_NAME
- ATTR_MAX_EXIT_SUCCESS
- ATTR_MAX_LOAD
- ATTR_MAX_RUN_ALARM
- ATTR_MBEAN_ATTR
- ATTR_MBEAN_NAME
- ATTR_MBEAN_OPERATION
- ATTR_MESSAGE_CLASS
- ATTR_METHOD_NAME
- ATTR_MF_JCL_NAME
- ATTR_MF_JCL_TYPE
- ATTR_MF_SERVER
- ATTR_MF_SERVER_ADDRESS_TYPE
- ATTR_MF_USER
- ATTR_MF_VERSION
- ATTR_MIN_RUN_ALARM
- ATTR_MODE
- ATTR MODIFY_PARAMETER
- ATTR MODIFY_PARAMETER_NUM
- ATTR_MONITOR_COND
- ATTR_MONITOR_MODE
- ATTR_MONITOR_TYPE
- ATTR_MUST_COMPLETE_TIMES
- ATTR_MUST_START_TIMES
- ATTR_N_RETRY
- ATTR_NAME
- ATTR_NEW_SPOOL
- ATTR_NEXT_PRIORITY

- ATTR_NEXT_START
- ATTR_NO_CHANGE
- ATTR_NO_FORWARD
- ATTR_NO_PRINT
- ATTR_NODE_NAME
- ATTR_NOTIFICATION_ID
- ATTR_NOTIFICATION_MSG
- ATTR_NTRY
- ATTR_NUM_COLUMNS
- ATTR_NUM_LINES
- ATTR_OBJECT_NAME
- ATTR_ONE_WAY
- ATTR_OPERATION
- ATTR_OPERATION_TYPE
- ATTR_OPSYS
- ATTR_ORACLE_APPL_NAME
- ATTR_ORACLE_APPL_NAME_TYPE
- ATTR_ORACLE_ARGS
- ATTR_ORACLE_DESC
- ATTR_ORACLE_MON_CHILDREN
- ATTR_ORACLE_MON_CHILDREN_DELAY
- ATTR_ORACLE_PRINT_COPIES
- ATTR_ORACLE_PRINT_STYLE
- ATTR_ORACLE_PRINTER
- ATTR_ORACLE_PROGRAM
- ATTR_ORACLE_PROGRAMDATA
- ATTR_ORACLE_REQ_SET
- ATTR_ORACLE_RESP
- ATTR_ORACLE_SAVE_OUTPUT
- ATTR_ORACLE_USE_ARG_DEF
- ATTR_ORACLE_USER
- ATTR_OS_USER
- ATTR_OWNER

- ATTR_PARAMETER_NUM
- ATTR_PARENT_NAME
- ATTR_PERMISSION
- ATTR_PERMISSION_STRING
- ATTR_PID
- ATTR_POLL_INTERVAL
- ATTR_PORT
- ATTR_PORT_NAME
- ATTR_PORTS
- ATTR_PRINT_COPIES
- ATTR_PRINT_IMM
- ATTR_PRINT_PRIORITY
- ATTR_PRINT_STYLE
- ATTR_PRINTER
- ATTR_PRIORITY
- ATTR_PROC_STEP
- ATTR_PROCESS_NAME
- ATTR_PROCESS_STATUS
- ATTR_PROFILE
- ATTR_PROGRAM
- ATTR_PROVIDER_URL
- ATTR_PROVISION
- ATTR_PRT_ARC_MODE
- ATTR_PS_ARGS
- ATTR_PS_DEST_FORMAT
- ATTR_PS_DEST_TYPE
- ATTR_PS_DETAIL_FOLDER
- ATTR_PS_DLST_ROLES
- ATTR_PS_DLST_USERS
- ATTR_PS_EMAIL_ADDRESS
- ATTR_PS_EMAIL_ADDRESS_EXPANDED
- ATTR_PS_EMAIL_LOG
- ATTR_PS_EMAIL_SUBJECT

- ATTR_PS_EMAIL_TEXT
- ATTR_PS_EMAIL_WEB_REPORT
- ATTR_PS_OPERATOR_ID
- ATTR_PS_OUTPUT_DEST
- ATTR_PS_PROCESS_NAME
- ATTR_PS_PROCESS_TYPE
- ATTR_PS_RESTARTS
- ATTR_PS_RUN_CNTRL_ARGS
- ATTR_PS_RUN_CNTRL_ID
- ATTR_PS_RUN_CONTROL_TABLE
- ATTR_PS_SERVER_NAME
- ATTR_PS_SKIP_PARM_UPDATES
- ATTR_PS_TIME_ZONE
- ATTR_QUANTITY
- ATTR_QUE_NAME
- ATTR_QUEUE_NAME
- ATTR_RC
- ATTR_RCP_TYPE
- ATTR_RECIPIENT
- ATTR_RECIPIENT_NAME
- ATTR_RELEASE
- ATTR_REMOTE_NAME
- ATTR_REMOVE_REFERENCES
- ATTR_REPLY_MSG
- ATTR_REPLY_RSP
- ATTR_REQ_TYPE
- ATTR_REQUEST_ID
- ATTR_RES_NAME
- ATTR_RES_TYPE
- ATTR_RESOURCE_NAME
- ATTR_RESOURCES
- ATTR_RESTART
- ATTR_RESULT

- ATTR_RESULT_TYPE
- ATTR_RETURN_CLASS_NAME
- ATTR_RETURN_NAMESPACE
- ATTR_RETURN_XML_NAME
- ATTR_ROID
- ATTR_RUN_CALENDAR
- ATTR_RUN_MACHINE
- ATTR_RUN_NUM
- ATTR_RUN_PRIORITY
- ATTR_RUN_WINDOW
- ATTR_RUN_WINDOW_STRING
- ATTR_RUNNING
- ATTR_SAP_ABAP_NAME
- ATTR_SAP_BANNER
- ATTR_SAP_CHAIN_ID
- ATTR_SAP_CLIENT
- ATTR_SAP_EVENT_ID
- ATTR_SAP_EVENT_PARM
- ATTR_SAP_EXT_TABLE
- ATTR_SAP_FAIL_MSG
- ATTR_SAP_INFO_PACK
- ATTR_SAP_IS_TRIGGER
- ATTR_SAP_JOB_CLASS
- ATTR_SAP_JOB_COUNT
- ATTR_SAP_JOB_NAME
- ATTR_SAP_LANG
- ATTR_SAP_MAILLIST
- ATTR_SAP_MON_CHILD
- ATTR_SAP_OFFICE
- ATTR_SAP_PRINT_PARMS
- ATTR_SAP_PROC_TYPE
- ATTR_SAP_PROC_USER
- ATTR_SAP_PROCESS_CLIENT

- ATTR_SAP_PROCESS_STATUS
- ATTR_SAP_RECIPIENTS
- ATTR_SAP_RELEASE_OPTION
- ATTR_SAP_RFC_DEST
- ATTR_SAP_STEP_NUM
- ATTR_SAP_STEP_PARMS
- ATTR_SAP_SUCCESS_MSG
- ATTR_SAP_TARGET_JOBNAME
- ATTR_SAP_TARGET_SYS
- ATTR_SAVEOP
- ATTR SCP LOCAL NAME
- ATTR SCP LOCAL USER
- ATTR SCP PROTOCOL
- ATTR SCP REMOTE_DIR
- ATTR SCP REMOTE_NAME
- ATTR SCP SERVER_NAME
- ATTR SCP SERVER_PORT
- ATTR SCP TARGET_OS
- ATTR SCP TRANSFER_DIRECTION
- ATTR SEARCH_BW
- ATTR SEND_NOTIFICATION
- ATTR SEQ_NUM
- ATTR SERVICE_DESK
- ATTR SERVICE_NAME
- ATTR SHELL
- ATTR SIGN
- ATTR SNMP AUTH_PROTOCOL
- ATTR SNMP COMM_STRING
- ATTR SNMP HOST
- ATTR SNMP MIB
- ATTR SNMP OID
- ATTR SNMP PARAMETER_NUM
- ATTR SNMP PRIVACY

- ATTR_SNMP_PRIVACY_USER
- ATTR_SNMP_VALUE
- ATTR_SNMP_VERSION
- ATTR_SP_ARG
- ATTR_SP_NAME
- ATTR_SPOOL_NAME
- ATTR_SQL_COMMAND
- ATTR_START_MINS
- ATTR_START_MINS_STRING
- ATTR_START_TIMES
- ATTR_START_TIMES_STRING
- ATTR_STARTING
- ATTR_STATUS
- ATTR_STATUS_LONG
- ATTR_STATUS_SHORT
- ATTR_STATUS_TIME
- ATTR_STD_ERR_FILE
- ATTR_STD_IN_FILE
- ATTR_STD_OUT_FILE
- ATTR_STEP_NAME
- ATTR_STEP_USER
- ATTR_STR_OID
- ATTR_STRING_VALUES
- ATTR_SUCCESS
- ATTR_SUCCESS_CODES
- ATTR_SUCCESS_CRITERIA
- ATTR_SUCCESS_PATTERN
- ATTR_SVCDESK_ATTR
- ATTR_SVCDESK_DESC
- ATTR_SVCDESK_IMP
- ATTR_SVCDESK_PRI
- ATTR_SVCDESK_SEV
- ATTR_TABLENAME

- ATTR_TARGET_NAMESPACE
- ATTR_TERM_RUN_TIME
- ATTR_TERMINATED
- ATTR_TEXT_FILE_FILTER
- ATTR_TEXT_FILE_FILTER_EXISTS
- ATTR_TEXT_FILE_MODE
- ATTR_TEXT_FILE_NAME
- ATTR_TIME_FORMAT
- ATTR_TIME_POSITION
- ATTR_TIMEZONE
- ATTR_TITLE
- ATTR_TRIGGER_COND
- ATTR_TRIGGER_TYPE
- ATTR_TYPE
- ATTR_ULIMIT
- ATTR_UPDATE_STAMP
- ATTR_UPPER_BOUNDARY
- ATTR_URL
- ATTR_USE_TOPIC
- ATTR_USER
- ATTR_USER_ROLE
- ATTR_VALUE
- ATTR_VALUEOP
- ATTR_VARIANT
- ATTR_VERSION
- ATTR_WAKE_PASSWORD
- ATTR_WATCH_FILE
- ATTR_WATCH_FILE_CHANGE_TYPE
- ATTR_WATCH_FILE_CHANGE_VALUE
- ATTR_WATCH_FILE_GROUPNAME
- ATTR_WATCH_FILE_MIN_SIZE
- ATTR_WATCH_FILE_OWNER
- ATTR_WATCH_FILE_RECURSIVE

- ATTR_WATCH_FILE_TYPE
- ATTR_WATCH_FILE_WIN_USER
- ATTR_WATCH_INTERVAL
- ATTR_WATCH_NO_CHANGE
- ATTR_WEB_PARAMETER
- ATTR_WEB_USER
- ATTR_WIN_EVENT_CATEGORY
- ATTR_WIN_EVENT_COMPUTER
- ATTR_WIN_EVENT_DATETIME
- ATTR_WIN_EVENT_DESCRIPTION
- ATTR_WIN_EVENT_ID
- ATTR_WIN_EVENT_OP
- ATTR_WIN_EVENT_SOURCE
- ATTR_WIN_EVENT_TYPE
- ATTR_WIN_LOG_NAME
- ATTR_WIN_SERVICE_NAME
- ATTR_WIN_SERVICE_STATUS
- ATTR_WSDL_OPERATION
- ATTR_WSDL_URL
- ATTR_XCRYPT_TYPE
- ATTR_XKEY_TO_MANAGER
- ATTR_XMACHINE
- ATTR_XMANAGER
- ATTR_XPORT
- ATTR_XTYPE
- ATTR_ZOS_DATASET
- ATTR_ZOS_DSN_RENAMED
- ATTR_ZOS_DSN_UPDATED
- ATTR_ZOS_EXPLICIT_DSN
- ATTR_ZOS_FTP_DIRECTION
- ATTR_ZOS_FTP_HOST
- ATTR_ZOS_FTP_USERID
- ATTR_ZOS_JOBNAME

- ATTR_ZOS_TRIGGER_BY
- ATTR_ZOS_TRIGGER_ON
- ATTR_ZOS_TRIGGER_TYPE

JobFilterAnd

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobFilterComposite, IAndComposite

The JobFilterAnd class connects all contained filter objects (composite or leaf) with a logical AND.

JobFilterAnd

The JobFilterAnd constructor creates a JobFilterAnd object. This constructor has the following syntax:

```
public JobFilterAnd()
```

add

The add method in the JobFilterAnd class adds filter objects (composite or leaf) to this AND construct. For example, if you have three filter objects that you want to connect with a logical AND, call this method three times, once with each object. It results in an expression similar to this: "x and y and z". This method has the following syntax:

```
void add (IJobFilter filter)
```

filter

Defines the filter to AND.

toString

The toString method in the JobFilterAnd class has the following syntax:

```
java.lang.String toString()
```

getComponents

The getComponents method in the JobFilterAnd class returns the list of cached leaf filter components. This method has the following syntax:

```
java.util.ArrayList getComponents ()
```

JobFilterOr

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobFilterComposite, IOrComposite

JobFilterOr connects all contained filter objects (composite or leaf) with a logical OR.

JobFilterOr

The JobFilterOr constructor creates a JobFilterOr object. This constructor has the following syntax:

```
public JobFilterOr()
```

add

The add method in the JobFilterOr class adds filter objects (composite or leaf) to this OR construct. For example, if you have three filters you want to connect with a logical OR, call this method three times, once for each object. It results in an expression similar to this: "x or y or z". This method has the following syntax:

```
void add (IJobFilter filter)
```

filter

The filter to OR.

toString

The toString method in the JobFilterOr class has the following syntax:

```
java.lang.String toString()
```

getComponents

The getComponents method in the JobFilterOr class returns the list of cached leaf filter components. This method has the following syntax:

```
java.util.ArrayList getComponents ()
```

JobFilterDate

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobFilterDate

JobFilterDate is a JobFilter leaf that filters on dates.

JobFilterDate

This JobFilterDate constructor creates a JobFilterDate that matches the specified job field according to the specified operator. For example, to match jobs whose last start time is more recent than April 4, 2004, create the object as follows:

```
Date fdate = new Date(104, 3, 4);
JobFilterDate jfd = new JobFilterDate(JobFilterDate.FLT_LAST_START, jfd,
JobFilterDate.GREATER);
```

This constructor has the following syntax:

```
JobFilterDate(int id,java.util.Date date, int operator)
```

id

Specifies the ID of the date property. Possible values are:

- FLT_CREATE_STAMP
- FLT_UPDATE_STAMP
- FLT_LAST_END
- FLT_LAST_HEARTBEAT
- FLT_LAST_START
- FLT_NEXT_START
- FLT_STATUS_TIME

date

Defines the date to filter against.

operator

Specifies the operator to apply to this date. Possible values are:

- EQUAL
- GREATER
- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

JobFilterDate throws the following exception if an invalid operator is used:
`java.lang.IllegalArgumentException`

JobFilterDate

This JobFilterDate constructor is a copy constructor. This constructor has the following syntax:

`JobFilterDate(IJobFilterDate filter)`

filter

Defines an object that implements the IJobFilterDate interface.

toString

The `toString` method in the JobFilterDate class has the following syntax:

`java.lang.String toString()`

getFilterId

The `getFilterId` method in the JobFilterDate class returns the filter ID. This method has the following syntax:

`int getFilterId()`

getDate

The `getDate` method in the JobFilterDate class returns the date for this filter. This method has the following syntax:

`java.util.Date getDate()`

getOperator

The `getOperator` method in the `JobFilterDate` class returns the operator. This method has the following syntax:

```
int getOperator()
```

JobFilterInt

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobFilterInt

JobFilterInt creates a JobFilterInt that filters integer properties according to the given operator.

JobFilterInt

This JobFilterInt constructor is a copy constructor. This constructor has the following syntax:

JobFilterInt(IJobFilterInt *filter*)

filter

Defines an object that implements the IJobFilterInt interface.

JobFilterInt

This JobFilterInt constructor matches against a single integer. This constructor has the following syntax:

JobFilterInt(int *id*,int *value*,int *operator*)

id

Specifies the ID of the int property. Possible values are:

- FLT_BOX_JOID
- FLT_EXIT_CODE
- FLT_HEARTBEAT_INTERVAL
- FLT_JOB_LOAD
- FLT_JOID
- FLT_MAX_EXIT_SUCCESS
- FLT_MAX_RUN_ALARM
- FLT_MIN_RUN_ALARM
- FLT_N_RETRY
- FLT_NTRY
- FLT_PRIORITY
- FLT_RUN_NUM
- FLT_RUN_PRIORITY

- FLT_STATUS
- FLT_TERM_RUN_TIME
- FLT_WATCH_FILE_MIN_SIZE
- FLT_WATCH_INTERVAL
- ATTR_ACTION
- ATTR_ARGS_NUM
- ATTR_ARGLTYPE
- ATTR_BLIND_COPY
- ATTR_CONTINUOUS
- ATTR_COPY
- ATTR_CPU_USAGE
- ATTR_CREATE_PARAMETER_NUM
- ATTR_DATATYPE
- ATTR_DISK_FORMAT
- ATTR_DISK_SPACE
- ATTR_ENCRYPTION_TYPE
- ATTR_FILTER_TYPE
- ATTR_FREE
- ATTR_FTP_TRANSFER_DIRECTION
- ATTR_FTP_TRANSFER_TYPE
- ATTR_HAS_OVERRIDE
- ATTR_HOST_PAGE
- ATTR_I5_ACTION
- ATTR_I5_PROCESS_PRIORITY
- ATTR_IGNORE
- ATTR_INTERACTIVE
- ATTR_INVOCATION_TYPE
- ATTR_IP_STATUS
- ATTR_JOB_FILTER
- ATTR_JOB_TYPE
- ATTR_MAX_LOAD
- ATTR_MF_JCL_TYPE
- ATTR_MF_SERVER_ADDRESS_TYPE

- ATTR_MF_VERSION
- ATTR_MODE
- ATTR MODIFY_PARAMETER_NUM
- ATTR_MONITOR_MODE
- ATTR_MONITOR_TYPE
- ATTR_ONE_WAY
- ATTR_OPERATION
- ATTR_OPERATION_TYPE
- ATTR_OPSYS
- ATTR_ATTR_ORACLE_APPL_NAME_TYPE
- ATTR_PARAMETER_NUM
- ATTR_PROCESS_STATUS
- ATTR_PRT_ARC_MODE
- ATTR_PS_DEST_FORMAT
- ATTR_PS_DEST_TYPE
- ATTR_RCP_TYPE
- ATTR_RES_TYPE
- ATTR_RESULT
- ATTR_SAP_BANNER
- ATTR_SAP_PROC_TYPE
- ATTR_SAP_PROCESS_STATUS
- ATTR_SAP_RELEASE_OPTION
- ATTR_SCP_PROTOCOL
- ATTR_SCP_TRANSFER_DIRECTION
- ATTR_SCP_TARGET_OS
- ATTR_SCP_SERVER_PORT
- ATTR_SEND_NOTIFICATION
- ATTR_SEQ_NUM
- ATTR_SIGN
- ATTR_SNMP_AUTH_PROTOCOL
- ATTR_SNMP_PARAMETER_NUM
- ATTR_SNMP_PRIVACY
- ATTR_TEXT_FILE_MODE

- ATTR_TRIGGER_TYPE
- ATTR_VALUEOP
- ATTR_WATCH_FILE_CHANGE_TYPE
- ATTR_WATCH_FILE_RECURSIVE
- ATTR_WATCH_FILE_TYPE
- ATTR_WATCH_NO_CHANGE
- ATTR_WIN_EVENT_OP
- ATTR_WIN_EVENT_TYPE
- ATTR_WIN_SERVICE_STATUS
- ATTR_XCRYPT_TYPE
- ATTR_XPORT
- ATTR_ZOS_FTP_DIRECTION
- ATTR_ZOS_TRIGGER_TYPE

value

Defines the integer to filter against.

operator

Specifies the operator to apply to this integer. Possible values are:

- EQUAL
- GREATER
- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

JobFilterInt throws the following exception if an invalid operator is used:
java.lang.IllegalArgumentException

JobFilterInt

This JobFilterInt constructor matches against an array of integer values. If the particular field matches any of the integers in the list, it is a match (like a logical OR). This constructor has the following syntax:

`JobFilterInt(int id, int[] values, int operator)`

id

Specifies the ID of the int property. Possible values are:

- FLT_BOX_JOID

- FLT_EXIT_CODE
- FLT_HEARTBEAT_INTERVAL
- FLT_JOB_LOAD
- FLT_JOID
- FLT_MAX_EXIT_SUCCESS
- FLT_MAX_RUN_ALARM
- FLT_MIN_RUN_ALARM
- FLT_N_RETRYS
- FLT_NTRY
- FLT_PRIORITY
- FLT_RUN_NUM
- FLT_RUN_PRIORITY
- FLT_STATUS
- FLT_TERM_RUN_TIME
- FLT_WATCH_FILE_MIN_SIZE
- FLT_WATCH_INTERVAL
- ATTR_ACTION
- ATTR_ARGS_NUM
- ATTR_ARGLTYPE
- ATTR_BLIND_COPY
- ATTR_CONTINUOUS
- ATTR_COPY
- ATTR_CPU_USAGE
- ATTR_CREATE_PARAMETER_NUM
- ATTR_DATATYPE
- ATTR_DISK_FORMAT
- ATTR_DISK_SPACE
- ATTR_ENCRYPTION_TYPE
- ATTR_FILTER_TYPE
- ATTR_FREE
- ATTR_FTP_TRANSFER_DIRECTION
- ATTR_FTP_TRANSFER_TYPE
- ATTR_HAS_OVERRIDE

- ATTR_HOST_PAGE
- ATTR_I5_ACTION
- ATTR_I5_PROCESS_PRIORITY
- ATTR_IGNORE
- ATTR_INTERACTIVE
- ATTR_INVOCATION_TYPE
- ATTR_IP_STATUS
- ATTR_JOB_FILTER
- ATTR_JOB_TYPE
- ATTR_MAX_LOAD
- ATTR_MF_JCL_TYPE
- ATTR_MF_SERVER_ADDRESS_TYPE
- ATTR_MF_VERSION
- ATTR_MODE
- ATTR MODIFY_PARAMETER_NUM
- ATTR_MONITOR_MODE
- ATTR_MONITOR_TYPE
- ATTR_ONE_WAY
- ATTR_OPERATION
- ATTR_OPERATION_TYPE
- ATTR_OPSYS
- ATTR_ATTR_ORACLE_APPL_NAME_TYPE
- ATTR_PARAMETER_NUM
- ATTR_PROCESS_STATUS
- ATTR_PRT_ARC_MODE
- ATTR_PS_DEST_FORMAT
- ATTR_PS_DEST_TYPE
- ATTR_RCP_TYPE
- ATTR_RES_TYPE
- ATTR_RESULT
- ATTR_SAP_BANNER
- ATTR_SAP_PROC_TYPE
- ATTR_SAP_PROCESS_STATUS

- ATTR_SAP_RELEASE_OPTION
- ATTR SCP_PROTOCOL
- ATTR SCP_TRANSFER_DIRECTION
- ATTR SCP_TARGET_OS
- ATTR SCP_SERVER_PORT
- ATTR_SEND_NOTIFICATION
- ATTR SEQ_NUM
- ATTR SIGN
- ATTR SNMP_AUTH_PROTOCOL
- ATTR SNMP_PARAMETER_NUM
- ATTR SNMP_PRIVACY
- ATTR TEXT_FILE_MODE
- ATTR_TRIGGER_TYPE
- ATTR_VALUEOP
- ATTR_WATCH_FILE_CHANGE_TYPE
- ATTR_WATCH_FILE_RECURSIVE
- ATTR_WATCH_FILE_TYPE
- ATTR_WATCH_NO_CHANGE
- ATTR_WIN_EVENT_OP
- ATTR_WIN_EVENT_TYPE
- ATTR_WIN_SERVICE_STATUS
- ATTR_XCRYPT_TYPE
- ATTR_XPORT
- ATTR_ZOS_FTP_DIRECTION
- ATTR_ZOS_TRIGGER_TYPE

values

Identifies an integer array of values to match.

operator

Specifies the operator to apply to this integer. Possible values are:

- EQUAL
- NOT_EQUAL

JobFilterInt throws the following exception if an operator other than EQUAL or NOT_EQUAL is used: java.lang.IllegalArgumentException

toString

The `toString` method in the `JobFilterInt` class has the following syntax:

```
java.lang.String toString()
```

getFilterId

The `getFilterId` method in the `JobFilterInt` class returns the filter ID. This method has the following syntax:

```
int getFilterId()
```

getOperator

The `getOperator` method in the `JobFilterInt` class returns the operator. This method has the following syntax:

```
int getOperator()
```

getValue

The `getValue` method in the `JobFilterInt` class returns the integer value for this filter. This method has the following syntax:

```
int getValue()
```

getValues

The `getValues` method in the `JobFilterInt` class returns the integer values set for this filter. This method has the following syntax:

```
int[] getValues()
```

JobFilterString

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobFilterString

`JobFilterString` is a filter leaf object which is used to match on job String properties.

JobFilterString

This JobFilterString constructor creates a JobFilterString that matches jobs that have the given String value using the defined operator. This constructor has the following syntax:

```
JobFilterString(int id,java.lang.String value,int operator)
```

id

Specifies the ID of the String property. Possible values are:

- FLT_APPLICATION
- FLT_BOX_NAME
- FLT_EXCLUDE_CALENDAR
- FLT_GROUP
- FLT_JOB_NAME
- FLT_MACHINE
- FLT_OWNER
- FLT_RUN_CALENDAR
- FLT_RUN_MACHINE
- FLT_TIMEZONE
- FLT_SVCDESK_IMPACT
- FLT_SVCDESK_SEVERITY
- FLT_SVCDESK_PRIORITY
- FLT_SVCDESK_ATTRIBUTE
- FLT_NOTIFY_ID
- ATTR_ABAP_LANG
- ATTR_ABAP_NAME
- ATTR_ADMINISTRATOR
- ATTR_AFTER_TIME
- ATTR_AGENT_NAME
- ATTR_ALERT
- ATTR_AMOUNT
- ATTR_APPLICATION
- ATTR_ARC_CONNECT
- ATTR_ARC_DATE
- ATTR_ARC_DOC_CLASS
- ATTR_ARC_DOC_TYPE

- ATTR_ARC_FORMAT
- ATTR_ARC_HOST_LINK
- ATTR_ARC_INFO
- ATTR_ARC_OBJ_NAME
- ATTR_ARC_OBJ_TYPE
- ATTR_ARC_OBJ_VARIANT
- ATTR_ARC_PATH
- ATTR_ARC_PRINTER
- ATTR_ARC_PROTOCOL
- ATTR_ARC_REPORT
- ATTR_ARC_SERVICE
- ATTR_ARC_STORAGE
- ATTR_ARC_TEXT
- ATTR_ARC_USERID
- ATTR_ARC_VERSION
- ATTR_ARGS
- ATTR_AUTHORIZATION
- ATTR_I5_CC_EXIT
- ATTR_I5_CURR_LIB
- ATTR_I5_JOB_DESC
- ATTR_I5_JOB_NAME
- ATTR_I5_JOB_QUEUE
- ATTR_I5_LDA
- ATTR_I5_LIB
- ATTR_I5_LIBRARY_LIST
- ATTR_I5_NAME
- ATTR_I5_OTHERS
- ATTR_I5_PARAMS
- ATTR_AUTH_STRING
- ATTR_BDC_SYSTEM
- ATTR_BEAN_NAME
- ATTR_BOX_NAME
- ATTR_BROADCAST_ADDRESS

- ATTR_CHARACTER_CODE
- ATTR_CHK_FILES
- ATTR_CLASS_NAME
- ATTR_COMMAND
- ATTR_COND_JOB_AUTOSERV
- ATTR_COND_JOB_NAME
- ATTR_COND_OPERATOR
- ATTR_COND_TEST_GLOBAL_VALUE
- ATTR_COND_TYPE
- ATTR_CONNECT_STRING
- ATTR_CONNECTION_FACTORY
- ATTR_COPY_GENERATION
- ATTR_COPY_JCL
- ATTR_CREATE_METHOD
- ATTR_CREATE_NAME
- ATTR_DATASET
- ATTR_DAYS_OF_WEEK
- ATTR_DEPT
- ATTR_DESCRIPTION
- ATTR_DEST
- ATTR_DESTINATION_FILE
- ATTR_DESTINATION_NAME
- ATTR_DISK_DRIVE
- ATTR_ENCODING
- ATTR_ENDPOINT_URL
- ATTR_EXCLUDE_CALENDAR
- ATTR_FIELD_NAME
- ATTR_FILTER
- ATTR_FINDER_NAME
- ATTR_FORMAT
- ATTR_FTP_LOCAL_NAME
- ATTR_FTP_LOCAL_USER
- ATTR_FTP_REMOTE_NAME

- ATTR_FTP_SERVER_NAME
- ATTR_GROUP
- ATTR_HIGH
- ATTR_INITIAL_CONTEXT_FACTORY
- ATTR_INSTANCE
- ATTR_IP_HOST
- ATTR_J2EE_AUTHENTICATION_ORDER
- ATTR_J2EE_CONN_DOMAIN
- ATTR_J2EE_CONN_ORIGIN
- ATTR_J2EE_PROXY_DOMAIN
- ATTR_J2EE_PROXY_HOST
- ATTR_J2EE_PROXY_ORIGIN_HOST
- ATTR_J2EE_PROXY_USER
- ATTR_J2EE_USER
- ATTR_JCL_LIBRARY
- ATTR_JCL_MEMBER
- ATTR_JMX_USER
- ATTR_JOB_NAME
- ATTR_KEY_TO_AGENT
- ATTR_LOW
- ATTR_LOWER_BOUNDARY
- ATTR_MAC_ADDRESS
- ATTR_MACHINE
- ATTR_MACHINE_NAME
- ATTR_MBEAN_ATTR
- ATTR_MBEAN_NAME
- ATTR_MBEAN_OPERATION
- ATTR_MESSAGE_CLASS
- ATTR_METHOD_NAME
- ATTR_MF_JCL_NAME
- ATTR_MF_SERVER
- ATTR_MF_USER
- ATTR_MODIFY_PARAMETER

- ATTR_MONITOR_COND
- ATTR_NAME
- ATTR_NODE_NAME
- ATTR_NOTIFICATION_ID
- ATTR_NOTIFICATION_MSG
- ATTR_OBJECT_NAME
- ATTR_ORACLE_APPL_NAME
- ATTR_ORACLE_ARGS
- ATTR_ORACLE_DESC
- ATTR_ORACLE_PRINT_STYLE
- ATTR_ORACLE_PRINTER
- ATTR_ORACLE_PROGRAM
- ATTR_ORACLE_REQ_SET
- ATTR_ORACLE_RESP
- ATTR_ORACLE_USER
- ATTR_OS_USER
- ATTR_OWNER
- ATTR_PERMISSION
- ATTR_PORT_NAME
- ATTR_PORTS
- ATTR_PRINT_STYLE
- ATTR_PRINTER
- ATTR_PROC_STEP
- ATTR_PROCESS_NAME
- ATTR_PROFILE
- ATTR_PROGRAM
- ATTR_PROVIDER_URL
- ATTR_PS_ARGS
- ATTR_PS_DETAIL_FOLDER
- ATTR_PS_DLST_ROLES
- ATTR_PS_DLST_USERS
- ATTR_PS_EMAIL_ADDRESS
- ATTR_PS_EMAIL_ADDRESS_EXPANDED

- ATTR_PS_EMAIL SUBJECT
- ATTR_PS_EMAIL TEXT
- ATTR_PS_OPERATOR_ID
- ATTR_PS_OUTPUT_DEST
- ATTR_PS_PROCESS_NAME
- ATTR_PS_PROCESS_TYPE
- ATTR_PS_RUN_CNTL_ID
- ATTR_PS_RUN_CONTROL_TABLE
- ATTR_PS_SERVER_NAME
- ATTR_PS_TIME_ZONE
- ATTR_QUEUE_NAME
- ATTR_RC
- ATTR_RECIPIENT
- ATTR_RECIPIENT_NAME
- ATTR_REMOTE_NAME
- ATTR_REQUEST_ID
- ATTR_REQ_TYPE
- ATTR_RES_NAME
- ATTR_RESOURCE_NAME
- ATTR_RESULT_TYPE
- ATTR_RETURN_CLASS_NAME
- ATTR_RETURN_NAMESPACE
- ATTR_RETURN_XML_NAME
- ATTR_RUN_CALENDAR
- ATTR_RUN_MACHINE
- ATTR_RUN_WINDOW
- ATTR_SAP_CHAIN_ID
- ATTR_SAP_CLIENT
- ATTR_SAP_EVENT_ID
- ATTR_SAP_EVENT_PARM
- ATTR_SAP_FAIL_MSG
- ATTR_SAP_INFO_PACK
- ATTR_SAP_JOB_NAME

- ATTR_SAP_LANG
- ATTR_SAP_PROC_USER
- ATTR_SAP_RFC_DEST
- ATTR_SAP_SUCCESS_MSG
- ATTR_SAP_TARGET_JOBNAME
- ATTR_SAP_TARGET_SYS
- ATTR_SCP_REMOTE_DIR
- ATTR_SCP_REMOTE_NAME
- ATTR_SCP_LOCAL_NAME
- ATTR_SCP_LOCAL_USER
- ATTR_SCP_SERVER_NAME
- ATTR_SEARCH_BW
- ATTR_SERVICE_NAME
- ATTR_SHELL
- ATTR_SNMP_COMM_STRING
- ATTR_SNMP_HOST
- ATTR_SNMP_MIB
- ATTR_SNMP_OID
- ATTR_SNMP_PRIVACY_USER
- ATTR_SP_NAME
- ATTR_SPOOL_NAME
- ATTR_SQL_COMMAND
- ATTR_STD_ERR_FILE
- ATTR_STD_IN_FILE
- ATTR_STD_OUT_FILE
- ATTR_STEP_NAME
- ATTR_STEP_USER
- ATTR_STRING_VALUES
- ATTR_SUCCESS_CRITERIA
- ATTR_SUCCESS_PATTERN
- ATTR_SVCDESK_ATTR
- ATTR_SVCDESK_DESC
- ATTR_TABLENAME

- ATTR_TARGET_NAMESPACE
- ATTR_TEXT_FILE_FILTER
- ATTR_TEXT_FILE_NAME
- ATTR_TIME_FORMAT
- ATTR_TIMEZONE
- ATTR_TITLE
- ATTR_TRIGGER_COND
- ATTR_UPPER_BOUNDARY
- ATTR_URL
- ATTR_USER
- ATTR_USER_ROLE
- ATTR_VALUE
- ATTR_VARIANT
- ATTR_VERSION
- ATTR_WAKE_PASSWORD
- ATTR_WATCH_FILE
- ATTR_WATCH_FILE_GROUPNAME
- ATTR_WATCH_FILE_OWNER
- ATTR_WATCH_FILE_WIN_USER
- ATTR_WEB_USER
- ATTR_WIN_EVENT_CATEGORY
- ATTR_WIN_EVENT_COMPUTER
- ATTR_WIN_EVENT_DESCRIPTION
- ATTR_WIN_EVENT_ID
- ATTR_WIN_EVENT_SOURCE
- ATTR_WIN_LOG_NAME
- ATTR_WIN_SERVICE_NAME
- ATTR_WSDL_OPERATION
- ATTR_WSDL_URL
- ATTR_XKEY_TO_MANAGER
- ATTR_XMACHINE
- ATTR_XMANAGER
- ATTR_ZOS_DATASET

- ATTR_ZOS_FTP_HOST
- ATTR_ZOS_FTP_USERID
- ATTR_ZOS_JOBNAME
- ATTR_ZOS_TRIGGER_BY

value

Defines the String value to match on. Wildcards are accepted.

operator

Specifies the operator to apply to this String. Possible values are:

- EQUAL
- GREATER
- LESS
- LIKE
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL
- NOT_LIKE

A `java.lang.IllegalArgumentException` is thrown if an invalid operator is given.

JobFilterString

This JobFilterString constructor creates a JobFilterString that matches job string fields that match the given String value. '%' can be used as a wildcard. This constructor has the following syntax:

```
JobFilterString(int id,java.lang.String value)
```

id

Specifies the ID of the String property. Possible values are:

- FLT_APPLICATION
- FLT_BOX_NAME
- FLT_EXCLUDE_CALENDAR
- FLT_GROUP
- FLT_JOB_NAME
- FLT_MACHINE
- FLT_OWNER
- FLT_RUN_CALENDAR
- FLT_RUN_MACHINE
- FLT_TIMEZONE
- FLT_SVCDESK_IMPACT
- FLT_SVCDESK_SEVERITY
- FLT_SVCDESK_PRIORITY
- FLT_SVCDESK_ATTRIBUTE
- FLT_NOTIFY_ID
- ATTR_ABAP_LANG
- ATTR_ABAP_NAME
- ATTR_ADMINISTRATOR
- ATTR_AFTER_TIME
- ATTR_AGENT_NAME
- ATTR_ALERT
- ATTR_AMOUNT
- ATTR_APPLICATION
- ATTR_ARC_CONNECT
- ATTR_ARC_DATE
- ATTR_ARC_DOC_CLASS

- ATTR_ARC_DOC_TYPE
- ATTR_ARC_FORMAT
- ATTR_ARC_HOST_LINK
- ATTR_ARC_INFO
- ATTR_ARC_OBJ_NAME
- ATTR_ARC_OBJ_TYPE
- ATTR_ARC_OBJ_VARIANT
- ATTR_ARC_PATH
- ATTR_ARC_PRINTER
- ATTR_ARC_PROTOCOL
- ATTR_ARC_REPORT
- ATTR_ARC_SERVICE
- ATTR_ARC_STORAGE
- ATTR_ARC_TEXT
- ATTR_ARC_USERID
- ATTR_ARC_VERSION
- ATTR_ARGS
- ATTR_AUTHORIZATION
- ATTR_I5_CC_EXIT
- ATTR_I5_CURR_LIB
- ATTR_I5_JOB_DESC
- ATTR_I5_JOB_NAME
- ATTR_I5_JOB_QUEUE
- ATTR_I5_LDA
- ATTR_I5_LIB
- ATTR_I5_LIBRARY_LIST
- ATTR_I5_NAME
- ATTR_I5_OTHERS
- ATTR_I5_PARAMS
- ATTR_AUTH_STRING
- ATTR_BDC_SYSTEM
- ATTR_BEAN_NAME
- ATTR_BOX_NAME

- ATTR_BROADCAST_ADDRESS
- ATTR_CHARACTER_CODE
- ATTR_CHK_FILES
- ATTR_CLASS_NAME
- ATTR_COMMAND
- ATTR_COND_JOB_AUTOSERV
- ATTR_COND_JOB_NAME
- ATTR_COND_OPERATOR
- ATTR_COND_TEST_GLOBAL_VALUE
- ATTR_COND_TYPE
- ATTR_CONNECT_STRING
- ATTR_CONNECTION_FACTORY
- ATTR_COPY_GENERATION
- ATTR_COPY_JCL
- ATTR_CREATE_METHOD
- ATTR_CREATE_NAME
- ATTR_DATASET
- ATTR_DAYS_OF_WEEK
- ATTR_DEPT
- ATTR_DESCRIPTION
- ATTR_DEST
- ATTR_DESTINATION_FILE
- ATTR_DESTINATION_NAME
- ATTR_DISK_DRIVE
- ATTR_ENCODING
- ATTR_ENDPOINT_URL
- ATTR_EXCLUDE_CALENDAR
- ATTR_FIELD_NAME
- ATTR_FILTER
- ATTR_FINDER_NAME
- ATTR_FORMAT
- ATTR_FTP_LOCAL_NAME
- ATTR_FTP_LOCAL_USER

- ATTR_FTP_REMOTE_NAME
- ATTR_FTP_SERVER_NAME
- ATTR_GROUP
- ATTR_HIGH
- ATTR_INITIAL_CONTEXT_FACTORY
- ATTR_INSTANCE
- ATTR_IP_HOST
- ATTR_J2EE_AUTHENTICATION_ORDER
- ATTR_J2EE_CONN_DOMAIN
- ATTR_J2EE_CONN_ORIGIN
- ATTR_J2EE_PROXY_DOMAIN
- ATTR_J2EE_PROXY_HOST
- ATTR_J2EE_PROXY_ORIGIN_HOST
- ATTR_J2EE_PROXY_USER
- ATTR_J2EE_USER
- ATTR_JCL_LIBRARY
- ATTR_JCL_MEMBER
- ATTR_JMX_USER
- ATTR_JOB_NAME
- ATTR_KEY_TO_AGENT
- ATTR_LOW
- ATTR_LOWER_BOUNDARY
- ATTR_MAC_ADDRESS
- ATTR_MACHINE
- ATTR_MACHINE_NAME
- ATTR_MBEAN_ATTR
- ATTR_MBEAN_NAME
- ATTR_MBEAN_OPERATION
- ATTR_MESSAGE_CLASS
- ATTR_METHOD_NAME
- ATTR_MF_JCL_NAME
- ATTR_MF_SERVER
- ATTR_MF_USER

- ATTR MODIFY_PARAMETER
- ATTR_MONITOR_COND
- ATTR_NAME
- ATTR_NODE_NAME
- ATTR_NOTIFICATION_ID
- ATTR_NOTIFICATION_MSG
- ATTR_OBJECT_NAME
- ATTR_ORACLE_APPL_NAME
- ATTR_ORACLE_ARGS
- ATTR_ORACLE_DESC
- ATTR_ORACLE_PRINT_STYLE
- ATTR_ORACLE_PRINTER
- ATTR_ORACLE_PROGRAM
- ATTR_ORACLE_REQ_SET
- ATTR_ORACLE_RESP
- ATTR_ORACLE_USER
- ATTR_OS_USER
- ATTR_OWNER
- ATTR_PERMISSION
- ATTR_PORT_NAME
- ATTR_PORTS
- ATTR_PRINT_STYLE
- ATTR_PRINTER
- ATTR_PROC_STEP
- ATTR_PROCESS_NAME
- ATTR_PROFILE
- ATTR_PROGRAM
- ATTR_PROVIDER_URL
- ATTR_PS_ARGS
- ATTR_PS_DETAIL_FOLDER
- ATTR_PS_DLST_ROLES
- ATTR_PS_DLST_USERS
- ATTR_PS_EMAIL_ADDRESS

- ATTR_PS_EMAIL_ADDRESS_EXPANDED
- ATTR_PS_EMAIL SUBJECT
- ATTR_PS_EMAIL_TEXT
- ATTR_PS_OPERATOR_ID
- ATTR_PS_OUTPUT_DEST
- ATTR_PS_PROCESS_NAME
- ATTR_PS_PROCESS_TYPE
- ATTR_PS_RUN_CNTL_ID
- ATTR_PS_RUN_CONTROL_TABLE
- ATTR_PS_SERVER_NAME
- ATTR_PS_TIME_ZONE
- ATTR_QUEUE_NAME
- ATTR_RC
- ATTR_RECIPIENT
- ATTR_RECIPIENT_NAME
- ATTR_REMOTE_NAME
- ATTR_REQUEST_ID
- ATTR_REQ_TYPE
- ATTR_RES_NAME
- ATTR_RESOURCE_NAME
- ATTR_RESULT_TYPE
- ATTR_RETURN_CLASS_NAME
- ATTR_RETURN_NAMESPACE
- ATTR_RETURN_XML_NAME
- ATTR_RUN_CALENDAR
- ATTR_RUN_MACHINE
- ATTR_RUN_WINDOW
- ATTR_SAP_CHAIN_ID
- ATTR_SAP_CLIENT
- ATTR_SAP_EVENT_ID
- ATTR_SAP_EVENT_PARM
- ATTR_SAP_FAIL_MSG
- ATTR_SAP_INFO_PACK

- ATTR_SAP_JOB_NAME
- ATTR_SAP_LANG
- ATTR_SAP_PROC_USER
- ATTR_SAP_RFC_DEST
- ATTR_SAP_SUCCESS_MSG
- ATTR_SAP_TARGET_JOBNAME
- ATTR_SAP_TARGET_SYS
- ATTR_SCP_REMOTE_DIR
- ATTR_SCP_REMOTE_NAME
- ATTR_SCP_LOCAL_NAME
- ATTR_SCP_LOCAL_USER
- ATTR_SCP_SERVER_NAME
- ATTR_SEARCH_BW
- ATTR_SERVICE_NAME
- ATTR_SHELL
- ATTR_SNMP_COMM_STRING
- ATTR_SNMP_HOST
- ATTR_SNMP_MIB
- ATTR_SNMP_OID
- ATTR_SNMP_PRIVACY_USER
- ATTR_SP_NAME
- ATTR_SPOOL_NAME
- ATTR_SQL_COMMAND
- ATTR_STD_ERR_FILE
- ATTR_STD_IN_FILE
- ATTR_STD_OUT_FILE
- ATTR_STEP_NAME
- ATTR_STEP_USER
- ATTR_STRING_VALUES
- ATTR_SUCCESS_CRITERIA
- ATTR_SUCCESS_PATTERN
- ATTR_SVCDESK_ATTR
- ATTR_SVCDESK_DESC

- ATTR_TABLENAME
- ATTR_TARGET_NAMESPACE
- ATTR_TEXT_FILE_FILTER
- ATTR_TEXT_FILE_NAME
- ATTR_TIME_FORMAT
- ATTR_TIMEZONE
- ATTR_TITLE
- ATTR_TRIGGER_COND
- ATTR_UPPER_BOUNDARY
- ATTR_URL
- ATTR_USER
- ATTR_USER_ROLE
- ATTR_VALUE
- ATTR_VARIANT
- ATTR_VERSION
- ATTR_WAKE_PASSWORD
- ATTR_WATCH_FILE
- ATTR_WATCH_FILE_GROUPNAME
- ATTR_WATCH_FILE_OWNER
- ATTR_WATCH_FILE_WIN_USER
- ATTR_WEB_USER
- ATTR_WIN_EVENT_CATEGORY
- ATTR_WIN_EVENT_COMPUTER
- ATTR_WIN_EVENT_DESCRIPTION
- ATTR_WIN_EVENT_ID
- ATTR_WIN_EVENT_SOURCE
- ATTR_WIN_LOG_NAME
- ATTR_WIN_SERVICE_NAME
- ATTR_WSDL_OPERATION
- ATTR_WSDL_URL
- ATTR_XKEY_TO_MANAGER
- ATTR_XMACHINE
- ATTR_XMANAGER

- ATTR_ZOS_DATASET
- ATTR_ZOS_FTP_HOST
- ATTR_ZOS_FTP_USERID
- ATTR_ZOS_JOBNAME
- ATTR_ZOS_TRIGGER_BY

value

Defines the String value on which to match. Wildcards are accepted.

JobFilterString

This JobFilterString constructor is a copy constructor. This constructor has the following syntax:

```
JobFilterString(IJobFilterString filter)
```

filter

Defines an object that implements the IJobFilterString interface.

toString

The `toString` method in the `JobFilterString` class has the following syntax:

```
java.lang.String toString()
```

getFilterId

The `getFilter` method in the `JobFilterString` class returns the filter ID. This method has the following syntax:

```
int getFilterId()
```

getOperator

The `getOperator` method in the `JobFilterString` class returns the operator. This method has the following syntax:

```
int getOperator()
```

getValue

The `getValue` method in the `JobFilterString` class returns the String value set for this filter. This method has the following syntax:

```
java.lang.String getValue()
```

JobFilterBool

extends

com.ca.autosys.services.request.filter.Filter

implements

IFilterLeaf, IJobFilterBool

JobFilterBool is used to match job fields that are Boolean in nature.

JobFilterBool

This JobFilterBool constructor has the following syntax:

```
JobFilterBool(int id, boolean value)
```

id

Specifies the ID of the Boolean property. Possible values are:

- FLT_ALARM_IF_FAIL
- FLT_AUTO_HOLD
- FLT_BOX_TERMINATOR
- FLT_JOB_TERMINATOR
- FLT_DATE_CONDITIONS
- FLT_SVCDESK_ENABLE
- ATTR_ALARM
- ATTR_ALARM_IF_FAIL
- ATTR_ALARM_VERIF
- ATTR_ALL_EVENTS
- ATTR_ALL_STATUS
- ATTR_AUTO_HOLD
- ATTR_BANNER_PAGE
- ATTR_BDC_EXT_LOG
- ATTR_BOX_TERMINATOR
- ATTR_CURRUN
- ATTR_DATE_CONDITIONS
- ATTR_EXPRESS
- ATTR_FAILURE
- ATTR_FOOTER
- ATTR_FTP_USE_SSL
- ATTR_J2EE_NO_GLOBAL_PROXY_DEFAULTS
- ATTR_JOB_TERMINATOR
- ATTR_NEW_SPOOL
- ATTR_NO_FORWARD
- ATTR_NO_PRINT
- ATTR_ORACLE_MON_CHILDREN

- ATTR_ORACLE_SAVE_OUTPUT
- ATTR_ORACLE_USE_ARG_DEF
- ATTR_PRINT_IMM
- ATTR_PROVISION
- ATTR_PS_EMAIL_LOG
- ATTR_PS_EMAIL_WEB_REPORT
- ATTR_PS_RESTARTS
- ATTR_PS_SKIP_PARM_UPDATES
- ATTR_RELEASE
- ATTR_REMOVE_REFERENCES
- ATTR_RESTART
- ATTR_RUNNING
- ATTR_SAP_MON_CHILD
- ATTR_SAP_OFFICE
- ATTR_SAVEOP
- ATTR_SERVICE_DESK
- ATTR_STARTING
- ATTR_SUCCESS
- ATTR_TERMINATED
- ATTR_TEXT_FILE_FILTER_EXISTS
- ATTR_USE_TOPIC
- ATTR_ZOS_DSN_RENAMED
- ATTR_ZOS_DSN_UPDATED
- ATTR_ZOS_EXPLICIT_DSN

value

Defines the Boolean value on which to match.

JobFilterBool

This JobFilterBool constructor is a copy constructor. This constructor has the following syntax:

`JobFilterBool(IJobFilterBool filter)`

filter

Defines an object that implements the IJobFilterBool interface.

toString

The `toString` method in the `JobFilterBool` class has the following syntax:

```
java.lang.String toString()
```

getFilterId

The `GetFilterId` method in the `JobFilterBool` class returns the filter ID. This method has the following syntax:

```
int getFilterId()
```

getValue

The `getValue` method in the `JobFilterBool` class returns the Boolean value. This method has the following syntax:

```
boolean getValue()
```

JobFilterChar

extends

com.ca.autosys.services.request.filter.FilterChar

implements

IFilterLeaf, IJobFilterChar

JobFilterChar is a JobFilter leaf that filters on Boolean values.

JobFilterChar

This constructor creates a JobFilterChar that matches character fields based on the given char value according to the supplied operator. This constructor has the following syntax:

`JobFilterChar(int id, char value, int operator)`

id

Specifies the ID of the character property. Possible values are:

- FLT_JOB_TYPE
- FLT_NOTIFY_CONDITION
- ATTR_BLOB_TYPE
- ATTR_JOB_CLASS
- ATTR_SAP_JOB_CLASS
- ATTR_SVCDESK_IMP
- ATTR_SVCDESK_PRI
- ATTR_SVCDESK_SEV
- ATTR_TYPE
- ATTR_XTYPE

value

Defines the value to filter against.

operator

Specifies the comparison operator. Possible values are:

- EQUAL
- NOT_EQUAL

JobFilterChar throws the following exception if an operator other than EQUAL or NOT_EQUAL is used: `java.lang.IllegalArgumentException`

JobFilterChar

This constructor creates a JobFilterChar that matches character fields based on the supplied set of char values. If the operator supplied in this constructor is EQUAL and the job field matches any character in the supplied array, it is considered a match. If the supplied operator is NOT_EQUAL and the job field does not match any character in the supplied array, it is considered a match. This constructor has the following syntax:

`JobFilterChar(int id, char[] values, int operator)`

id

Specifies the ID of the character property. Possible values are:

- FLT_JOB_TYPE
- FLT_NOTIFY_CONDITION
- ATTR_BLOB_TYPE
- ATTR_JOB_CLASS
- ATTR_SAP_JOB_CLASS
- ATTR_SVCDESK_IMP
- ATTR_SVCDESK_PRI
- ATTR_SVCDESK_SEV
- ATTR_TYPE
- ATTR_XTYPE

values

Defines the values to filter against.

operator

Specifies the comparison operator. Possible values are:

- EQUAL
- NOT_EQUAL

JobFilterChar throws the following exception if an operator other than EQUAL or NOT_EQUAL is used: `java.lang.IllegalArgumentException`

JobFilterChar

JobFilterChar is a convenience constructor that builds a filter based on the contents of the given filter. This constructor has the following syntax:

JobFilterChar(IJobFilterChar *filter*)

filter

Defines an object that implements the IJobFilterBool interface.

toString

The `toString` method in the JobFilterChar class has the following syntax:

```
java.lang.String toString()
```

Methods Inherited

This class inherits the following methods from the FilterChar class:

- `getFilterId`
- `getOperator`
- `getValue`
- `getValues`

GetJobsWithFilterRsp

extends

ApiResponse

implements

IGetJobsWithFilterRsp

GetJobsWithFilterRsp

The GetJobsWithFilterRsp constructor has the following syntax:

```
public GetJobsWithFilterRsp(com.ca.autosys.services.JFilterRsp rsp,  
GetJobsWithFilterRspSet rspSet)
```

rsp

Defines the raw response object.

A `java.lang.IllegalArgumentException` is thrown if a null response is given.

getAttributes

The `getAttributes` method in the `GetJobsWithFilterRsp` class returns the list of attributes, as defined in `IJobFilterAttributes`, which should be equal to the number of attributes requested. This method has the following syntax:

```
int[] getAttributes()
```

getAttributeName

The `getAttributeName` method in the `GetJobsWithFilterRsp` class returns the string name resembling the column name for this attribute. This method has the following syntax:

```
java.lang.String getAttributeName(int attribute)
```

attribute

Defines the integer representation of the attribute as specified in `IJobFilterAttributes`.

getAttributeType

The `getAttributeType` method in the `GetJobsWithFilterRsp` class returns the attribute type for the given attribute. The type returns the corresponding get function to call. This method has the following syntax:

```
int getAttributeType(int attribute)
```

attribute

Returns the attribute type for the given attribute. Possible values and the corresponding get functions are:

- `TYPE_INT`—`getInt()`
- `TYPE_STRING`—`getString()`
- `TYPE_BOOLEAN`—`getBoolean()`
- `TYPE_DATE`—`getDate()`
- `TYPE_INT_LIST`—`getInts()`
- `TYPE_TIMEOFDAY_LIST`—`getTimesOfDay()`
- `TYPE_CHKFILE_LIST`—`getChkFiles()`
- `TYPE_CHAR`—`getChar()`

getInt

The `getInt` method in the `GetJobsWithFilterRsp` class returns the int value for the given attribute. This method has the following syntax:

```
int getInt(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. Possible values are:

- `ATTR_JOID`
- `ATTR_APPL_NTRY`
- `ATTR_EXIT_CODE`
- `ATTR_HEARTBEAT_INTERVAL`
- `ATTR_JC_PID`
- `ATTR_JOB_LOAD`
- `ATTR_MAX_EXIT_SUCCESS`
- `ATTR_MAX_RUN_ALARM`
- `ATTR_MIN_RUN_ALARM`
- `ATTR_N_RETRY`
- `ATTR_NTRY`
- `ATTR_PID`
- `ATTR_PRIORITY`
- `ATTR_RUN_NUM`
- `ATTR_RUN_PRIORITY`
- `ATTR_STATUS`
- `ATTR_TERM_RUN_TIME`
- `ATTR_TOTAL_ITEMS`
- `ATTR_WATCH_FILE_MIN_SIZE`
- `ATTR_WATCH_INTERVAL`

getLong

The `getLong` method in the `GetJobsWithFilterRsp` class returns the long value of a given job attribute. This method has the following syntax:

```
public long getLong(int attribute)
```

attribute

Defines the job attribute. Valid attributes are:

- `WATCH_FILE_MIN_SIZE`
- `WATCH_FILE_CHANGE_VALUE`

getString

The `getString` method in the `GetJobsWithFilterRsp` class returns the string value for the given attribute. This method has the following syntax:

```
java.lang.String getString(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. Possible values are:

- `ATTR_APPLICATION`
- `ATTR_BOX_FAILURE`
- `ATTR_BOX_NAME`
- `ATTR_BOX_SUCCESS`
- `ATTR_CHK_FILES_STRING`
- `ATTR_COMMAND`
- `ATTR_CONDITION`
- `ATTR_DAYS_OF_WEEK_STRING`
- `ATTR_DESCRIPTION`
- `ATTR_EXCLUDE_CALENDAR`
- `ATTR_GROUP`
- `ATTR_JOB_NAME`
- `ATTR_MACHINE`
- `ATTR_OWNER`
- `ATTR_PERMISSION_STRING`
- `ATTR_PROFILE`
- `ATTR_RUN_MACHINE`
- `ATTR_RUN_WINDOW_STRING`
- `ATTR_START_MINUTES_STRING`
- `ATTR_START_TIMES_STRING`
- `ATTR_STD_ERR_FILE`
- `ATTR_STD_IN_FILE`
- `ATTR_STD_OUT_FILE`
- `ATTR_TIMEZONE`
- `ATTR_WATCH_FILE`

getBoolean

The `getBoolean` method in the `GetJobsWithFilterRsp` class returns the boolean value for the given job attribute. This method has the following syntax:

```
boolean getBoolean(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. Possible values are:

- `ATTR_ALARM_IF_FAIL`
- `ATTR_AUTO_DELETE`
- `ATTR_AUTO_HOLD`
- `ATTR_BOX_TERMINATOR`
- `ATTR_JOB_TERMINATOR`

getDate

The `getDate` method in the `GetJobsWithFilterRsp` class returns the date value of the given job attribute. This method has the following syntax:

```
java.util.Date getDate(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. Possible values are:

- `ATTR_LAST_END`
- `ATTR_LAST_HEARTBEAT`
- `ATTR_LAST_START`
- `ATTR_STATUS_TIME`

getInts

The `getInts` method in the `GetJobsWithFilterRsp` class returns the list of integers. This method has the following syntax:

```
int[] getInts(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. Possible values are:

- `ATTR_DAYS_OF_WEEK`
- `ATTR_START_MINS`

getTimesOfDay

The `getTimesOfDay` method in the `GetJobsWithFilterRsp` class returns the list of hours and minutes. This method has the following syntax:

```
com.ca.autosys.services.TimeOfDay[] getTimesOfDay(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. Possible values are:

- ATTR_RUN_WINDOW
- ATTR_START_TIMES

getChkFiles

The `getChkFiles` method in the `GetJobsWithFilterRsp` class returns the list of chk_files and the corresponding permissions. This method has the following syntax:

```
com.ca.autosys.services.ChkFile[] getChkFiles(int attribute)
```

attribute

Specifies the attribute whose value to retrieve. ATTR_CHK_FILES is the only valid attribute to pass in.

getChar

The `getChar` method in the `GetJobsWithFilterRsp` class returns the char value for the given attribute. This method has the following syntax:

```
char getChar(int attribute)
```

attribute

Specifies the job attribute. Possible values are:

- ATTR_JOB_TYPE
- ATTR_SEND_NOTIFICATION

getJobItem

The `getJobItem` method in the `GetJobsWithFilterRsp` class returns the `JobItem` object that is the repeating item's container. This item will have to be stored or cast into the appropriate container. This method has the following syntax:

```
public JobItem getJobItem(int attribute)
```

attribute

Specifies the job attribute to return.

ChkFile

extends
java.lang.Object

The ChkFile API is a container class that represents a chk_files object.

ChkFile

The ChkFile constructor has the following syntax:

```
public ChkFile(java.lang.String file, int size)
```

getFile

The getFile method in the ChkFile class returns the file name. This method has the following syntax:

```
java.lang.String getFile()
```

getSize

The getSize method in the ChkFile class returns the size. This method has the following syntax:

```
int getSize()
```

toString

The toString method in the ChkFile class overrides toString in java.lang.Object class. This method has the following syntax:

```
java.lang.String toString()
```

TimeOfDay

extends

`java.lang.Object`

This is a container class representing the time of day in hours and minutes.

TimeOfDay

The TimeOfDay constructor creates a TimeOfDay object. This constructor has the following syntax:

```
public TimeOfDay(int hour, int minute)
```

getHour

The getHour method in the TimeOfDay class returns the hour of the day. This method has the following syntax:

```
public int getHour()
```

getMinute

The getMinute method in the TimeOfDay class returns the minute of the hour. This method has the following syntax:

```
public int getMinute()
```

toString

The toString method in the TimeOfDay class has the following syntax:

```
public java.lang.String toString()
```

This method overrides toString in class `java.lang.Object`.

JobItem

extends

`java.lang.Object`

The JobItem class helps in defining repeating job items. This class is the superclass for all repeating fields in a job definition. When creating a repeating field in a job definition, the subclasses should be used directly.

Note: JobItems can be retrieved from API responses only once. When a JobItem has been retrieved, it is marked as processed and cannot be retrieved a second time. The client program should store the JobItems for future use.

JobItem

This constructor creates a JobItem object. This constructor has the following syntax:

```
public JobItem()
```

add

This method imbeds a JobItem inside this JobItem. This method throws the AsFieldValidationException if the add fails because of field validation. This constructor has the following syntax:

```
public void add(JobItem itemObj)  
    throws AsFieldValidationException
```

itemObj

Specifies the item to imbed.

getAttributes

This method gets the list of attributes that make up this JobItem. This method has the following syntax:

```
public int[] getAttributes()
```

getBool

This method gets the value of a keyword referencing a Boolean field. This method has the following syntax:

```
public boolean getBool(int keyword)
```

keyword

Specifies the keyword to get the value for.

getInt

This method gets the value of a keyword referencing an int field. This method has the following syntax:

```
public int getInt(int keyword)
```

keyword

Specifies the keyword to get the value for.

getItemType

This method gets the item type of the JobItem object. This method has the following syntax:

```
public int getItemType()
```

getJobItem

This method gets an imbedded JobItem from this JobItem. This method has the following syntax:

```
public JobItem getJobItem(int attribute)
```

attribute

Specifies the JobItem type to retrieve.

getLoggableItem

This method gets a string of all fields that were returned in the JobItem. This is useful in logging the JobItem. This method has the following syntax:

```
public java.lang.String getLoggableItem(boolean isSubItem)
```

isSubItem

Specifies True if this JobItem is imbedded in another JobItem.

getString

This method gets the value of a keyword referencing a string field. This method has the following syntax:

```
public java.lang.String getString(int keyword)
```

keyword

Specifies the keyword to get the value for.

hasSubItems

This method checks the JobItem for imbedded JobItems. This method has the following syntax:

```
public boolean hasSubItems()
```

set

This method sets a boolean JobItem value. This method has the following syntax:

```
public void set(int keyword, boolean value)
```

keyword

Specifies the keyword to be set.

value

Specifies the value for the keyword.

set

This method sets an integer JobItem value. This method has the following syntax:

```
public void set(int keyword, int value)
```

keyword

Specifies the keyword to be set.

value

Specifies the value for the keyword.

set

This method sets a boolean JobItem value. This method has the following syntax:

```
public void set(int keyword, boolean value)
```

keyword

Specifies the keyword to be set.

value

Specifies the value for the keyword.

toString

This method returns the string representation of the JobItem's contents and overrides `toString` in `java.lang.Object`. This method has the following syntax:

```
public java.lang.String toString()
```

GetJobTypesDetail

GetJobTypesDetail retrieves the job type information from the CA Workload Automation AE database.

Example: Sample Code for Executing the GetJobTypesDetail API

```
private void sample(AsApi api) {
    // Set up the response set and the request.
    IApiResponseSet types = null;
    GetJobTypesDetailReq req = new GetJobTypesDetailReq();

    // Request all job types.
    req.setRequest("ALL");

    // Execute the request, list the results, and check for errors.
    try {
        types = req.execute(api);

        while(types.hasNext()) {
            GetJobTypesDetailRsp rsp = (GetJobTypesDetailRsp) types.next();
            System.out.println("\nJob Type Detail:\n\n" +
                "Type:          " + rsp.getJobType() + "\n" +
                "Type Binary:   " + rsp.getBinary() + "\n" +
                "Type Description: " + rsp.getDescription());
        }
        System.out.println("\n");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsObjectDoesNotExistException noObjEx) {
        System.out.println("AsObjectDoesNotExistException: " +
noObjEx.getMessage());
    } catch (AsBadJobTypeException badJobEx) {
        System.out.println("AsBadJobTypeException: " + badJobEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetJobTypesDetailReq

extends

Cat2Request

implements

IGetJobTypesDetailReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsObjectDoesNotExistException - The requested type does not exist.
- AsBadJobTypeException - The requested type is invalid.

setRequest

The setRequest method in the GetJobTypesDetailReq class configures the request for execution. This method has the following syntax:

```
public void setRequest(java.lang.String jobType)
```

jobType

Specifies the job type to retrieve. Specify ALL to return all job types.

GetJobTypesDetailRsp

extends

ApiResponse

implements

IGetJobTypesDetailRsp

This is the GetJobTypesDetail response object.

GetJobTypesDetailRsp

The GetJobTypesDetailRsp constructor creates a GetJobTypesDetailRsp object. This constructor has the following syntax:

```
public GetJobTypesDetailRsp(com.ca.autosys.services.JResponse response)
```

getJobType

The getJobType method in the GetJobTypesDetailRsp class returns the job type. This method has the following syntax:

```
public char getJobType()
```

getBinary

The `getBinary` method in the `GetJobTypesDetailRsp` class returns the binary associated with the job type. This method has the following syntax:

```
public java.lang.String getBinary()
```

getDescription

The `getDescription` method in the `GetJobTypesDetailRsp` class returns the description of the job type. This method has the following syntax:

```
public java.lang.String getDescription()
```

GetMachDefs

`GetMachDefs` retrieves machine definitions for the requested machine. `GetMachDefs` is used by `autorep -M -q`.

`GetMachDefs` does not accept wildcards. The code must specify a specific machine name or `ALL` to retrieve all machine definitions.

If you have list access to the `AUTOREP` security resource, `GetMachDefs` does not verify individual machine security. If you do not have the appropriate list access and appropriate access the individual machine, `GetMachDefs` does not return a response.

The machines contained in a virtual machine are returned as separate responses. They are associated with the parent machine by the values in the returned response.

Example: Sample Code for Executing the GetMachDefs API

```
private void sample(AsApi appServer, String machineName) {  
    // Set up the request  
    IGetMachDefsReq req = new GetMachDefsReq();  
    req.setRequest(machineName);  
  
    // Execute the request, display results, and check for errors.  
    try {  
        IApiResponseSet machines = req.execute(appServer);  
  
        while(machines.hasNext()) {  
            GetMachDefsRsp rsp = (GetMachDefsRsp) machines.next();  
            System.out.println("\nMachine: " + rsp.getMachineName() + "\nType: "  
+ rsp.getMachineType());  
        }  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsObjectDoesNotExistException noObjEx) {  
        System.out.println("AsObjectDoesNotExistException: " +  
noObjEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

GetMachDefsReq

extends
Cat2Request
implements
IGetMachDefsReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsObjectDoesNotExistException - The specified machine does not exist.

setRequest

The setRequest method in the GetMachDefsReq class specifies the machine name to use in the request. This method has the following syntax:

```
void setRequest(java.lang.String mach_name)  
mach_name
```

Specifies the name of the machine for which to request the machine definition. Set this value to ALL to return all machine definitions. Wildcards are not valid.

GetMachDefsRsp

extends
ApiResponse
implements
IGetMachDefsRsp

The GetMachDefsRsp class is an individual response that is returned when iterating through a GetMachDefsRspSet.

GetMachJobs

GetMachJobs retrieves the jobs currently running on the requested machines. GetMachJobs is used by autorep -M -d.

GetMachJobs does not support wildcards. To retrieve load information for all machines, set the machine name to ALL. When the machine name is set to ALL, GetMachJobs verifies security for the AUTOREP list security resource. If permission is granted, GetMachJobs does not verify security for each job returned. If permission fails, GetMachJobs verifies job access individually, and returns empty responses for jobs to which you do not have read permission.

Example: Sample Code for Executing the GetMachJobs API

```
private void sample(AsApi appServer, String machineName) {

    // Set up the request object.
    IGetMachJobsReq req = new GetMachJobsReq();
    req.setRequest(machineName);

    // Execute the request, check for errors, display results.
    try {
        IApiResponseSet machines = req.execute(appServer);

        while(machines.hasNext()) {
            GetMachJobsRsp rsp = (GetMachJobsRsp) machines.next();
            System.out.println("\nMachine:      " + rsp.getDefinedMachine() +
                "\nLoad:          " + rsp.getDefinedLoad() +
                "\nJob Name:     " + rsp.getName() +
                "\nPriority:     " + rsp.getPriority() +
                "\nRun Machine:  " + rsp.getRunMachine() +
                "\nRun Priority: " + rsp.getRunPriority());
        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsObjectDoesNotExistException noObjEx) {
        System.out.println("AsObjectDoesNotExistException: " +
            noObjEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetMachJobsReq

extends
Cat2Request
implements
IGetMachJobsReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsObjectDoesNotExistException—The specified machine does not exist.

setRequest

The setRequest method in the GetMachJobsReq class defines the name of the machine to use in the request. This method has the following syntax:

```
void setRequest(java.lang.String mach_name)  
mach_name
```

Defines the name of the machine for which to retrieve load information. Wildcards are not supported. Set this value to ALL to retrieve load information for all defined machines.

GetMachJobsRsp

extends
ApiResponse
implements
IGetMachJobsRsp

The GetMachJobsRsp class is an individual response that is returned when iterating through a GetMachJobsRspSet.

getDefinedLoad

The getDefinedLoad method in the GetMachJobsRsp class returns the defined load for the job named by the getName method. This method has the following syntax:

```
int getDefinedLoad()
```

getDefinedMachine

The `getDefinedMachine` method in the `GetMachJobsRsp` class returns the defined machine for the job named by the `getName` method. This method has the following syntax:

```
java.lang.String getDefinedMachine()
```

getName

The `getName` method in the `GetMachJobsRsp` class returns the name of the job for which load information is being returned. This method has the following syntax:

```
java.lang.String getName()
```

getPriority

The `getPriority` method in the `GetMachJobsRsp` class returns the defined priority of the job named by the `getName` method. This method has the following syntax:

```
int getPriority()
```

getRunMachine

The `getRunMachine` method in the `GetMachJobsRsp` class returns the name of the machine on which the job named by the `getName` method ran. This method has the following syntax:

```
java.lang.String getRunMachine()
```

getRunPriority

The `getRunPriority` method in the `GetMachJobsRsp` class returns the priority with which the job named by the `getName` method ran. This method has the following syntax:

```
int getRunPriority()
```

getStatus

The `getStatus` method in the `GetMachJobsRsp` class returns the status of the job named by the `getName` method. This method has the following syntax:

```
int getStatus()
```

GetMachRuns

GetMachRuns retrieves the dynamic load information for the requested machines. GetMachRuns is used by autorep -M.

GetMachRuns does not support wildcards. To retrieve load information for all machines, set the machine name to ALL. When the machine name is set to ALL, GetMachRuns verifies security for the AUTOREP list security resource. If permission is granted, GetMachRuns does not verify security for each machine returned. If permission fails, GetMachRuns verifies machine access individually, and returns empty responses (zero-length machine names) for machines to which you do not have read permission.

Example: Sample Code for Executing the GetMachRuns API

```
private void sample(AsApi appServer, String machineName) {
    // Set up the request.
    IGetMachRunsReq req = new GetMachRunsReq();
    req.setRequest(machineName);

    // Execute the request, display the data, and check for errors.
    try {
        IApiResponseSet machines = req.execute(appServer);

        while(machines.hasNext()) {
            GetMachRunsRsp rsp = (GetMachRunsRsp) machines.next();
            System.out.println("\nMachine: " + rsp.getMachineName() + "\n" +
                "Type: " + rsp.getMachineType() + "\n" +
                "Load: " + rsp.getLoad());
        }
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsObjectDoesNotExistException noObjEx) {
        System.out.println("AsObjectDoesNotExistException: " +
noObjEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

GetMachRunsReq

extends
Cat2Request
implements
IGetMachRunsReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsObjectDoesNotExistException - The specified machine does not exist.

setRequest

The setRequest method in the GetMachRunsReq class defines the name of the machine to use in the request. This method has the following syntax:

```
void setRequest(java.lang.String mach_name)  
mach_name
```

Specifies the name of machine for which to retrieve dynamic load information. Wildcards are not accepted. Set this value to ALL to retrieve load information for all defined machines.

GetMachRunsRsp

extends
ApiResponse
implements
IGetMachRunsRsp

The GetMachRunsRsp class is an individual response that is returned when iterating through a GetMachRunsRspSet.

getLoad

The getLoad method in the GetMachRunsRsp class returns the current load on the machine named by the getMachineName method. This method has the following syntax:

```
int getLoad()
```

getMachineName

The `getMachineName` method in the `GetMachRunsRsp` class returns the name of the machine for which load information is being returned. This method has the following syntax:

```
java.lang.String getMachineName()
```

getMachineType

The `getMachineType` method in the `GetMachRunsRsp` class returns the type of machine named by the `getMachineName` method. This method has the following syntax:

```
char getMachineType()
```

getParentName

The `getParentName` method in the `GetMachRunsRsp` class returns the name of the virtual machine of which the machine named by the `getMachineName` method is a component. This method returns no value when the requested machine is not a component of a virtual machine. This method has the following syntax:

```
java.lang.String getParentName()
```

getMachineStatus

The `getMachineStatus` method in the `GetMachRunsRsp` class returns the current status of the machine. This method has the following syntax:

```
int getMachineStatus()
```

GetObjectNames

GetObjectNames returns the names of the objects that match the requested pattern for the specified type.

Example: Sample Code for Executing the GetObjectNames API

```
void sample(AsApi appServer) {  
    GetObjectNamesReq req = new GetObjectNamesReq();  
    req.setRequest("sample%", GetObjectNamesReq.JOB);  
  
    try {  
        ApiResponseSet rspSet = (ApiResponseSet)req.execute(appServer);  
  
        while(rspSet.hasNext()) {  
            GetObjectNamesRsp rsp = (GetObjectNamesRsp)rspSet.next();  
            System.out.println("----- Begin Response -----");  
            System.out.println("Name: " + rsp.getName());  
            System.out.println("----- End Response -----\\n");  
        }  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

GetObjectNamesReq

extends
Cat2Request
implements
IGetObjectNamesReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The setRequest method in the GetObjectNamesReq class sets the name of jobs, machines, calendars, and global variables to retrieve. This is a fast way for retrieving only the names of objects without retrieving all the attributes of those objects. This method has the following syntax:

```
void setRequest(java.lang.String name, int type)
```

name

Defines the name pattern to retrieve.

type

Defines the type of object to retrieve. Possible values are:

- JOB
- MACHINE
- CALENDAR
- GLOBAL_VARIABLE
- EXTENDED_CALENDAR
- CYCLE_CALENDAR
- GLOBAL_BLOB
- APPLICATION
- GROUP

GetObjectNamesRsp

extends
ApiResponse
implements
IGetObjectNamesRsp

The GetObjectNamesRsp class represents one object name.

getName

The getName method in the GetObjectNamesRsp class retrieves the object name. This method has the following syntax:

```
java.lang.String getName()
```

GetResourcesWithFilter

GetResourcesWithFilter lets you retrieve resource information from the CA EWA database based on supplied filtering criteria.

GetResourcesWithFilterReq

extends

Cat2Request

implements

IGetResourcesWithFilterReq

GetResourcesWithFilterReq

The GetResourcesWithFilterReq constructor creates a GetResourcesWithFilterReq object. This constructor has the following syntax:

```
public GetResourcesWithFilterReq()
```

SetRequest

The SetRequest method in the GetResourcesWithFilterReq class sets up the GetResourcesWithFilterReq request for processing. This method has the following syntax:

```
public void setRequest(IResourceFilter filter, int[] attributes)
```

GetResourcesWithFilterRsp

extends
ApiResponse
implements
IGetResourcesWithFilterRsp

This is an individual response that is returned when iterating through a GetResourcesWithFilterRspSet.

GetResourcesWithFilterRsp

The GetResourcesWithFilterRsp constructor creates a GetResourcesWithFilterRsp object. This constructor has the following syntax:

```
public GetResourcesWithFilterRsp(com.ca.autosys.services.JFilterRsp rsp,  
GetResourcesWithFilterRspSet rspSet)
```

rsp

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

getAttributes

The getAttributes method in the GetResourcesWithFilterRsp class returns the list of attributes present in this response. This list should correspond to the attributes set in the initial request. This method has the following syntax:

```
public int[] getAttributes()
```

getAttributeName

The getAttributeName method in the GetResourcesWithFilterRsp class returns the string name resembling the column name for this attribute. This method has the following syntax:

```
public java.lang.String getAttributeName(int attribute)
```

attribute

Defines the attribute.

getAttributeType

The `getAttributeType` method in the `GetResourcesWithFilterRsp` class returns the attribute type for the given attribute and the corresponding get function to call. This method has the following syntax:

```
public int getAttributeType(int attribute)
```

attribute

Defines the attribute.

Possible return values and their corresponding get functions are as follows:

- `TYPE_INT`—`getInt()`
- `TYPE_STRING`—`getString()`
- `TYPE_BOOLEAN`—`getBoolean()`
- `TYPE_DATE`—`getDate()`
- `TYPE_INT_LIST`—`getInts()`
- `TYPE_TIMEOFDAY_LIST`—`getTimesOfDay()`
- `TYPE_CHKFILE_LIST`—`getChkFiles()`
- `TYPE_CHAR`—`getChar()`

getInt

The `getInt` method in the `GetResourcesWithFilterRsp` class returns the int value for the given attribute. This method has the following syntax:

```
public int getInt(int attribute)
```

attribute

Defines the resource attribute.

getString

The `getString` method in the `GetResourcesWithFilterRsp` class returns the string value for the given resource attribute. This method has the following syntax:

```
public java.lang.String getString(int attribute)
```

attribute

Defines the resource attribute.

GetResourcesWithFilterRspSet

extends

FilterRspSet

This is a response set produced from a get resources with filter request. This set returns response objects of the type GetResourcesWithFilterRsp.

The following exceptions can be thrown when iterating over this set:

- AsApiGeneralException
- AsNoAttributesException—A request was submitted that defined no attributes to be returned.
- AsBadAttributesException—The request defined invalid attributes.
- AsBadFilterFieldsException—The request defined invalid filter fields.

GetResourcesWithFilterRspSet

The GetResourcesWithFilterRspSet constructor creates a GetResourcesWithFilterRspSet object. This constructor has the following syntax:

```
public GetResourcesWithFilterRspSet(com.ca.autosys.services.JResponseQ responseQ)
```

responseQ
Defines the raw ResponseQ object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

GetResourceUsage

GetResourceUsage retrieves the resource usage on a machine.

Example: Sample Code for Executing the GetResourceUsageReq API

```
private void sample(AsApi api, String RESOURCE_NAME, String MACHINE_NAME) {  
    // Set up the request.  
    IGetResourceUsageReq req = new GetResourceUsageReq();  
    int detailLvl = 1;  
    req.setRequest(RESOURCE_NAME, MACHINE_NAME, DETAIL_LEVEL);  
    req.setUser(TEST_USER, SERVER_NAME);  
  
    // Execute the request, check for errors.  
    try {  
        req.execute(api);  
        System.out.println("Resource " + RESOURCE_NAME + " has been deleted from  
Machine " + MACHINE_NAME);  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

GetResourceUsageReq

extends

Cat1Request

implements

IGetResourceUsageReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to insert the resource.

GetResourceUsageReq

The GetResourceUsageReq constructor creates a GetResourceUsageReq object. This constructor has the following syntax:

```
public GetResourceUsageReq()
```

SetRequest

The SetRequest method in the GetResourceUsageReq class sets up the GetResourceUsage request for processing. This method has the following syntax:

```
public void setRequest(java.lang.String resource_name, java.lang.String  
machine_name, int detailLvl)
```

GetResourceUsageRsp

extends
 ApiResponse
implements
 IGetResourceUsageRsp

This is a response to the get resource usage request.

GetResourceUsageRsp

The GetResourceUsageRsp constructor creates a GetResourceUsageRsp object. This constructor has the following syntax:

```
public GetResourceUsageRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

getSummaryCount

The getSummaryCount method in the GetResourceUsageRsp class is a getter method for the summary count. This method has the following syntax:

```
public int getSummaryCount()
```

getDetailCount

The getDetailCount method in the GetResourceUsageRsp class is a getter method for the detail count. This method has the following syntax:

```
public int getDetailCount()
```

getResName

The getResName method in the GetResourceUsageRsp class is a getter method for the resource name. This method has the following syntax:

```
public java.lang.String getResName(int idx)
```

getMachineName

The getMachineName method in the GetResourceUsageRsp class is a getter method for the machine name. This method has the following syntax:

```
public java.lang.String getMachineName(int idx)
```

getResType

The `getResType` method in the `GetResourceUsageRsp` class is a getter method for the resource type. This method has the following syntax:

```
public java.lang.String getResType(int idx)
```

getResDesc

The `getResDesc` method in the `GetResourceUsageRsp` class is a getter method for the resource description. This method has the following syntax:

```
public java.lang.String getResDesc(int idx)
```

getAmountDefined

The `getAmountDefined` method in the `GetResourceUsageRsp` class is a getter method for the amount defined. This method has the following syntax:

```
public int getAmountDefined(int idx)
```

getAmountAvailable

The `getAmountAvailable` method in the `GetResourceUsageRsp` class is a getter method for the amount available. This method has the following syntax:

```
public int getAmountAvailable(int idx)
```

getDetailResName

The `getDetailResName` method in the `GetResourceUsageRsp` class is a getter method for the resource name. This method has the following syntax:

```
public java.lang.String getDetailResName(int idx)
```

getDetailJobName

The `getDetailJobName` method in the `GetResourceUsageRsp` class is a getter method for the job name. This method has the following syntax:

```
public java.lang.String getDetailJobName(int idx)
```

getDetailJobStatus

The `getDetailJobStatus` method in the `GetResourceUsageRsp` class is a getter method for the job status. This method has the following syntax:

```
public java.lang.String getDetailJobStatus(int idx)
```

getDetailRunMachine

The `getDetailRunMachine` method in the `GetResourceUsageRsp` class is a getter method for the run machine. This method has the following syntax:

```
public java.lang.String getDetailRunMachine(int idx)
```

getDetailJobRunNum

The `getDetailJobRunNum` method in the `GetResourceUsageRsp` class is a getter method for the job run number. This method has the following syntax:

```
public int getDetailJobRunNum(int idx)
```

getDetailJobNtry

The `getDetailJobNtry` method in the `GetResourceUsageRsp` class is a getter method for the job Ntry. This method has the following syntax:

```
public int getDetailJobNtry(int idx)
```

getDetailAmountInUse

The `getDetailAmountInUse` method in the `GetResourceUsageRsp` class is a getter method for the amount in use. This method has the following syntax:

```
public int getDetailAmountInUse(int idx)
```

GetResourceUsageRspSet

extends

`ApiResponseSet`

This is a response set produced from a get machine jobs request. This set returns response objects of type `GetResourceUsageRsp`. Iterating over this set can throw an `AsApiGeneralException`.

GetResourceUsageRspSet

The `GetResourceUsageRspSet` constructor creates a `GetResourceUsageRspSet` object. This constructor has the following syntax:

```
public GetResourceUsageRspSet(com.ca.autosys.services.JResponseQ responseQ)  
responseQ
```

Defines the raw `responseQ` object.

A `java.lang.IllegalArgumentException` is thrown if a null response is given.

GetSchedulerLog

GetSchedulerLog requests the specified last number of lines of the Scheduler log with a possible keyword filter. Execution of this request produces a GetLogRsp.

Example: Sample Code for Executing the GetSchedulerLog API

```
void sample(AsApi appServer) {  
    GetSchedulerLogReq req = new GetSchedulerLogReq();  
    // Search the last 100 lines of the log for occurrences of "EVENT:".  
    req.setRequest("EVENT:", 100);  
  
    try {  
        GetLogRsp rsp = (GetLogRsp)req.execute(appServer);  
        String log = rsp.getLog();  
        System.out.println(log);  
    } catch (AsGeneralErrorException ge) {  
        System.out.println("AsGeneralErrorException: " + ge.getMessage());  
    } catch (AsSecurityException se) {  
        System.out.println("AsSecurityException: " + se.getMessage());  
    } catch (AsObjectDoesNotExistException noObjEx) {  
        System.out.println("AsObjectDoesNotExistException: " +  
noObjEx.getMessage());  
    } catch (AsException e) {  
        System.out.println("AsException: " + e.getMessage());  
    }  
}
```

GetSchedulerLogReq

extends
Cat1Request
implements
IGetSchedulerLogReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The setRequest method in the GetSchedulerLogReq class configures the request to retrieve the EP log. This method has the following syntax:

```
void setRequest(java.lang.String pattern, int numberOfLines)
```

pattern

Specifies the pattern on which to match each line of the EP log. This is used to filter the contents of the log.

numberOfLines

Specifies the number of lines from the end of the EP log to return.

GetLogRsp

extends
ApiResponse
implements
IGetLogRsp

GetLogRsp contains the response for the various get log requests.

getLog

The getLog method in the GetLogRsp class returns the requested log contents. This method has the following syntax:

```
public java.lang.String getLog()
```

GetStatistics

GetStatistics requests CA Workload Automation AE statistics.

Example: Sample Code for Executing the GetStatistics API

```
private void sample(AsApi api) {
    // Set up vars for a 90 day period ending now.
    long a_day = 86400000L;
    Date endDate = new Date();
    Date startDate = new Date((endDate.getTime() - (90 * a_day)));

    // Set up the attributes fields.
    int[] attributes = {
        IStatisticsAttributes.ATTR_JOB_FAILURES,
        IStatisticsAttributes.ATTR_ALARMS_JOB_FAILURE,
        IStatisticsAttributes.ATTR_ALARMS_START_JOB_FAILURE,
        IStatisticsAttributes.ATTR_JOBS_FAILURE,
        IStatisticsAttributes.ATTR_JOBS_TERMINATED
    };

    // Set up the request.
    IGetStatisticsReq req = new GetStatisticsReq();
    req.setRequest(IGetStatisticsReq.INTERVAL_DAILY,
                  startDate, endDate, attributes);

    // Execute the request, check for errors, show data.
    try {
        IApiResponseSet set = req.execute(api);

        while(set.hasNext()) {
            System.out.println("\n----- Begin Response -----");
            GetStatisticsRsp rsp = (GetStatisticsRsp) set.next();

            int[] rspAttributes = rsp.getAttributes();

            // Loop and display the returned data.
            for(int i=0; i<attributes.length; i++) {
                String attributeName = rsp.getAttributeName(rspAttributes[i]);
                int attributeType = rsp.getAttributeType(rspAttributes[i]);
                String attributeValue = "";

                switch(attributeType) {
                    case AsApiConstants.AtDouble:
                        attributeValue =
                            String.valueOf(rsp.getDouble(rspAttributes[i]));
                        break;

                    case AsApiConstants.AtInt:
```

```
        attributeName =
String.valueOf(rsp.getInt(rspAttributes[i]));
        break;
    }
    System.out.println(attributeName + " = " + attributeValue);
}
System.out.println("----- End Response -----\\n");
}
} catch (AsGeneralErrorException genEx) {
    System.out.println("AsGeneralErrorException: " + genEx.getMessage());
} catch (AsNoAttributesException noAttribEx) {
    System.out.println("AsNoAttributesException: " +
noAttribEx.getMessage());
} catch (AsBadAttributesException badAttribEx) {
    System.out.println("AsBadAttributesException: " +
badAttribEx.getMessage());
} catch (AsException asEx) {
    System.out.println("AsException: " + asEx.getMessage());
}
}
```

GetStatisticsReq

extends

Cat2Request

implements

IGetStatisticsReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsNoAttributesException - No attributes were defined.
- AsBadAttributesException - Bad attributes were requested.

setRequest

The `setRequest` method in the `GetStatisticsReq` class sets up the request for execution. This method has the following syntax:

```
public void setRequest(int dataType, java.util.Date startDate, java.util.Date endDate, int[] attributes)
```

dataType

Specifies the data type to retrieve. Possible values are:

- HOURLY
- DAILY
- WEEKLY
- MONTHLY
- YEARLY

startDate

Specifies the start date for retrieval.

endDate

Specifies the end date for retrieval.

attributes

Specifies the attributes on which to report. Possible values are:

- ATTR_JOBS_RUNNING
- ATTR_JOBS_STARTING
- ATTR_JOBS_SUCCESS
- ATTR_JOBS_FAILURE
- ATTR_JOBS_TERMINATED
- ATTR_JOBS_ON_HOLD
- ATTR_JOBS_ON_ICE
- ATTR_JOBS_INACTIVE
- ATTR_JOB_RUNS
- ATTR_JOB_FAILURES
- ATTR_JOB_FORCE_STARTS
- ATTR_JOB_RESTARTS
- ATTR_JOB_KILLS

- ATTR_JOB_OPEN_SVCDESK
- ATTR_ALARM_RESPONSE_TIME_AVG
- ATTR_ALARM_TOTAL
- ATTR_ALARMS_UNANSWERED
- ATTR_ALARMS_JOB_FAILURE
- ATTR_ALARMS_START_JOB_FAILURE.
- ATTR_ALARMS_MAX_RETRYS
- ATTR_ALARMS_MAX_RUNTIME
- ATTR_ALARMS_MIN_RUNTIME
- ATTR_ALARMS_DATABASE_ROLLOVER
- ATTR_ALARMS_SCHEDULER_ROLLOVER
- ATTR_ALARMS_SCHEDULER_SHUTDOWN
- ATTR_TOTAL_EVENTS
- ATTR_TOTAL_EVENT_LATENCY

GetStatisticsRsp

extends

ApiResponse

implements

IGetStatisticsRsp, IStatisticsAttributes

GetStatisticsRsp represents one response in the set of responses returned for a GetStatistics request.

getAttributeName

The getAttributeName method in the GetStatisticsRsp class returns the string name resembling the column name for this attribute. This method has the following syntax:

```
public java.lang.String getAttributeName(int attribute)
```

attribute

Specifies the attribute.

getAttributes

The `getAttributes` method in the `GetStatisticsRsp` class returns the list of attributes present in this response. This list should correspond to the attributes set in the initial request. This method has the following syntax:

```
public int[] getAttributes()
```

getAttributeType

The `getAttributeType` method in the `GetStatisticsRsp` class returns the attribute type for the given attribute. The returned type provides the corresponding get function to call. This method has the following syntax:

```
public int getAttributeType(int attribute)
```

attribute

Specifies the attribute on which to return the attribute type. Possible returns and the corresponding get functions are:

- `TYPE_INT` - `getInt()`
- `TYPE_STRING` - `getString()`
- `TYPE_BOOLEAN` - `getBoolean()`
- `TYPE_DATE` - `getDate()`
- `TYPE_INT_LIST` - `getInts()`
- `TYPE_TIMEOFDAY_LIST` - `getTimesOfDay()`
- `TYPE_CHKFILE_LIST` - `getChkFiles()`
- `TYPE_CHAR` - `getChar()`

getDataType

The `getDataType` method in the `GetStatisticsRsp` class returns the type of data retrieved. This method has the following syntax:

```
public int getDataType()
```

getDouble

The `getDouble` method in the `GetStatisticsRsp` class returns the double value of the specified attribute. This method has the following syntax:

```
public double getDouble(int attribute)
```

attribute

Specifies the statistics attribute, `ATTR_ALARM_RESPONSE_TIME_AVG`.

getInt

The getInt method in the GetStatisticsRsp class returns the integer value for the specified attribute. This method has the following syntax:

```
public int getInt(int attribute)
```

attribute

Specifies the statistics attribute. Valid attributes are:

- ATTR_JOBS_RUNNING
- ATTR_JOBS_STARTING
- ATTR_JOBS_SUCCESS
- ATTR_JOBS_FAILURE
- ATTR_JOBS_TERMINATED
- ATTR_JOBS_ON_HOLD
- ATTR_JOBS_ON_ICE
- ATTR_JOBS_INACTIVE
- ATTR_JOB_RUNS
- ATTR_JOB_FAILURES
- ATTR_JOB_FORCE_STARTS
- ATTR_JOB_RESTARTS
- ATTR_JOB_KILLS
- ATTR_JOB_OPEN_SVCDESK
- ATTR_ALARM_TOTAL
- ATTR_ALARMS_UNANSWERED
- ATTR_ALARMS_JOB_FAILURE
- ATTR_ALARMS_START_JOB_FAILURE
- ATTR_ALARMS_MAX_RETRYs
- ATTR_ALARMS_MAX_RUNTIME
- ATTR_ALARMS_MIN_RUNTIME
- ATTR_ALARMS_DATABASE_ROLLOVER
- ATTR_ALARMS_SCHEDULER_ROLLOVER
- ATTR_ALARMS_SCHEDULER_SHUTDOWN

getTimeStamp

The `getTimeStamp` method in the `GetStatisticsRsp` class returns the timestamp on the data. This method has the following syntax:

```
public java.util.Date getTimeStamp()
```

GetUniqueNames

`GetUniqueNames` is a cat2 API that requests the unique list of names for the specified type. No security checks are performed on the returned names.

Example: Sample Code for Executing the GetUniqueNames API

```
private void sample(AsApi api) {  
    GetUniqueNamesReq req = new GetUniqueNamesReq();  
    req.setRequest(IGetUniqueNamesReq.OWNER);  
  
    try {  
        ApiResponseSet rspSet = (ApiResponseSet)req.execute(api);  
  
        while(rspSet.hasNext()) {  
            GetUniqueNamesRsp rsp = (GetUniqueNamesRsp)rspSet.next();  
            System.out.println(rsp.getName());  
        }  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

GetUniqueNamesReq

extends
Cat2Request
implements
IGetUniqueNamesReq

GetUniqueNamesReq returns the unique list of names for the specified type. For example, it finds all owners specified for jobs and returns them. An owner of multiple jobs is only returned once.

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The setRequest method in the GetUniqueNamesReq class configures the request for the unique list of names. This method has the following syntax:

```
void setRequest(int type)  
type
```

Specifies the type of names to return. Possible values are:

OWNER

Returns all unique job owners.

MACHINE

Returns all unique machines defined in existing jobs.

RUN_MACHINE

Returns all unique run machines for existing jobs.

GetUniqueNamesRsp

extends
 ApiResponse
implements
 IGetUniqueNamesRsp

GetUniqueNamesRsp is the response object for the GetUniqueNames API.

getName

The getName method in the GetUniqueNamesRsp class returns the unique name for the specified type. This method has the following syntax:

```
java.lang.String getName()
```

I5OthersItem

extends

JobItem

The I5OthersItem is the container class for specifying i5 others arguments in job definitions. This JobItem subclass directly maps to the i5_others jil keyword.

Example: Sample Code for Using the I5OthersItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_I5);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_I5_ACTION,
        AsJobConstants.VALUE__I5_ACTION__COMMAND);
        insertReq.set(AsJobConstants.KEYWORD_I5_CC_EXIT,
        AsJobConstants.VALUE__I5_CC_EXIT__USER);
        insertReq.set(AsJobConstants.KEYWORD_I5_CURR_LIB, "myLibrary");
        insertReq.set(AsJobConstants.KEYWORD_I5_JOB_DESC, "TEST$I5$DESC");
        insertReq.set(AsJobConstants.KEYWORD_I5_NAME, "myCommand");

        // Set up the i5_others part of the job definition and
        // attach it to the base job definition.
        I5OthersItem i5oi = new I5OthersItem();
        i5oi.set(AsJobConstants.KEYWORD_I5_OTHERS, "other1=val1");
        i5oi.set(AsJobConstants.KEYWORD_I5_OTHERS, "other2=val2");
        i5oi.set(AsJobConstants.KEYWORD_I5_OTHERS, "other3=val3");
        insertReq.add(i5oi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_I5_ACTION,  
        AsJobConstants.KEYWORD_I5_CC_EXIT,  
        AsJobConstants.KEYWORD_I5_CURR_LIB,  
        AsJobConstants.KEYWORD_I5_JOB_DESC,  
        AsJobConstants.KEYWORD_I5_OTHERS  
    };  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();
```

```
// Process job level properties.  
System.out.println("----- Begin Response -----");  
for(int i=0; i<rspProperties.length; ++i) {  
    String keyword =  
        jobProperties.getKeywordFromConstant(rspProperties[i]);  
    switch(getRsp.getAttributeType(rspProperties[i])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
            System.out.println(keyword + ": " +  
                getRsp.getString(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
            System.out.println(keyword + ": " +  
                getRsp.getInt(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
            System.out.println(keyword + ": " +  
                getRsp.getBoolean(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_LONG:  
            System.out.println(keyword + ": " +  
                getRsp.getLong(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:  
            System.out.println("\n----- Begin " + keyword + " JobItem  
-----");  
            JobItem ji = getRsp.getJobItem(rspProperties[i]);  
            int itemProperties[] = ji.getAttributes();
```

```
// Process job item level properties.  
for(int j=0; j<itemProperties.length; ++j) {  
    System.out.print(keyword + ": ");  
  
    switch(jobProperties.getKeywordDataType(itemProperties[j])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
  
            System.out.println(ji.getString(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
  
            System.out.println(ji.getInt(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
  
            System.out.println(ji.getBool(itemProperties[j]));  
            break;  
    }  
    System.out.println("----- End " + keyword + " JobItem  
-----\n");  
    break;  
}  
}  
System.out.println("----- End Response -----");  
}  
} catch (AsInitializationException e) {  
    e.printStackTrace();  
} catch (AsException e) {  
    e.printStackTrace();  
}  
}  
}
```

Sample output

```
----- Begin Response -----
job_name: i5_others_sample
alarm_if_fail: true
machine: localhost
i5_action: 3
i5_cc_exit: 2
i5_curr_lib: myLibrary
i5_job_desc: TEST$I5$DESC

----- Begin i5_others JobItem -----
i5_others: other1=val1
i5_others: other2=val2
i5_others: other3=val3
----- End i5_others JobItem -----

----- End Response -----
```

I5OthersItem

This constructor has the following syntax:

```
public I5OthersItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

I5ParamsItem

extends

JobItem

The I5ParamsItem is the container class for specifying i5 params arguments in job definitions. This JobItem subclass directly maps to the i5_params jil keyword.

Example: Sample Code for Using the I5ParamsItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;
    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_I5);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_I5_ACTION,
        AsJobConstants.VALUE__I5_ACTION__COMMAND);
        insertReq.set(AsJobConstants.KEYWORD_I5_CC_EXIT,
        AsJobConstants.VALUE__I5_CC_EXIT__USER);
        insertReq.set(AsJobConstants.KEYWORD_I5_CURR_LIB, "myLibrary");
        insertReq.set(AsJobConstants.KEYWORD_I5_JOB_DESC, "TEST$I5$DESC");
        insertReq.set(AsJobConstants.KEYWORD_I5_NAME, "myCommand");

        // Set up the i5_params part of the job definition and
        // attach it to the base job definition.
        I5ParamsItem i5pi = new I5ParamsItem();
        i5pi.set(AsJobConstants.KEYWORD_I5_PARAMS, "param1=val1 ");
        i5pi.set(AsJobConstants.KEYWORD_I5_PARAMS, "param2=val2");
        i5pi.set(AsJobConstants.KEYWORD_I5_PARAMS, "param3=val3");
        insertReq.add(i5pi);
        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_I5_ACTION,  
        AsJobConstants.KEYWORD_I5_CC_EXIT,  
        AsJobConstants.KEYWORD_I5_CURR_LIB,  
        AsJobConstants.KEYWORD_I5_JOB_DESC,  
        AsJobConstants.KEYWORD_I5_PARAMS  
    };  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();
```

```
// Process job level properties.  
System.out.println("----- Begin Response -----");  
for(int i=0; i<rspProperties.length; ++i) {  
    String keyword =  
        jobProperties.getKeywordFromConstant(rspProperties[i]);  
    switch(getRsp.getAttributeType(rspProperties[i])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
            System.out.println(keyword + ": " +  
                getRsp.getString(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
            System.out.println(keyword + ": " +  
                getRsp.getInt(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
            System.out.println(keyword + ": " +  
                getRsp.getBoolean(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_LONG:  
            System.out.println(keyword + ": " +  
                getRsp.getLong(rspProperties[i]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:  
            System.out.println("\n----- Begin " + keyword + " JobItem  
-----");  
            JobItem ji = getRsp.getJobItem(rspProperties[i]);  
            int itemProperties[] = ji.getAttributes();
```

```
// Process job item level properties.  
for(int j=0; j<itemProperties.length; ++j) {  
    System.out.print(keyword + ": ");  
  
    switch(jobProperties.getKeywordDataType(itemProperties[j])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
  
            System.out.println(ji.getString(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
  
            System.out.println(ji.getInt(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
  
            System.out.println(ji.getBool(itemProperties[j]));  
            break;  
    }  
    System.out.println("----- End " + keyword + " JobItem  
-----\n");  
    break;  
}  
}  
System.out.println("----- End Response -----");  
}  
} catch (AsInitializationException e) {  
    e.printStackTrace();  
} catch (AsException e) {  
    e.printStackTrace();  
}  
}  
}
```

Sample output

```
----- Begin Response -----
job_name: i5_params_sample
alarm_if_fail: true
machine: localhost
i5_action: 3
i5_cc_exit: 2
i5_curr_lib: myLibrary
i5_job_desc:TEST$I5$DESC

----- Begin i5_params JobItem -----
i5_params: param1=val1
i5_params: param2=val2
i5_params: param3=val3
----- End i5_params JobItem -----

----- End Response -----
```

I5ParamsItem

This constructor has the following syntax:

```
public I5ParamsItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- `add`
- `getAttributes`
- `getBool`
- `getInt`
- `getItemType`
- `getJobItem`
- `getLoggableItem`
- `getString`
- `hasSubItems`
- `set(int keyword, boolean value)`
- `set(int keyword, int value)`
- `set(int keyword, java.lang.String value)`
- `toString`

IInsertJobReq

extends

ICat1Request

This interface exposes methods for InsertJobReq.

setRequest

This method sets up a request and gets it ready for execution. This method has the following syntax:

```
void setRequest(boolean verify)
```

verify

Validates the job condition as follows:

- True—Validates the job conditions and indicates the results in the response.
- False—Does not validate the job conditions.

set

This set method in the IInsertJobReq interface sets an integer job parameter. This method has the following syntax:

```
void set(int keyword, int value) throws AsFieldValidationException
```

keyword

Specifies the keyword constant to be set.

value

Specifies the integer value that keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the IInsertJobReq interface sets a string job parameter. This method has the following syntax:

```
void set(int keyword, java.lang.String value) throws AsFieldValidationException
```

keyword

Specifies the keyword constant to be set.

value

Specifies the string value that keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the IInsertJobReq interface sets a blob job parameter. This method has the following syntax:

```
void set(int keyword, byte[] value) throws AsFieldValidationException  
keyword
```

Specifies the keyword constant to be set.

value

Specifies the blob value that keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the IInsertJobReq interface sets a Boolean job parameter. This method has the following syntax:

```
void set(int keyword, boolean value) throws AsFieldValidationException  
keyword
```

Specifies the keyword constant to be set.

value

Specifies the Boolean value that keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the IInsertJobReq interface sets a double job parameter. This method has the following syntax:

```
void set(int keyword, double value) throws AsFieldValidationException  
keyword
```

Specifies the keyword constant to be set.

value

Specifies the double value that keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the IIInsertJobReq interface sets a long job parameter. This method has the following syntax:

```
void set(int keyword, long value) throws AsFieldValidationException
```

keyword

Specifies the keyword constant to be set.

value

Specifies the long value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

add

This method in the IIInsertJobReq interface adds repeating job parameters to the job. This method has the following syntax:

```
void add(JobItem itemObj) throws AsFieldValidationException
```

itemObj

Specifies the repeating job object to be added.

An AsFieldValidationException is thrown for an invalid field specification.

IInsertJobRsp

extends

IApiResponse

This interface exposes methods for InsertJobRsp.

hasUndefinedDependentJobs

This method verifies if the conditions defined in the request specify any undefined jobs.

This method has the following syntax:

```
boolean hasUndefinedDependentJobs()
```

This is TRUE only if the client specified "verify" in the request.

getUndefinedDependentJobs

This method gets undefined dependent jobs. This method has the following syntax:

```
java.lang.String[] getUndefinedDependentJobs()
```

This array will contain values only if hasUndefinedDependentJobs() returns TRUE.

IInsertUpdateJobReq

extends

IJobReq

This interface exposes methods for insert/update job request.

setRequest

This method configures the request. This method has the following syntax:

```
void setRequest(java.util.Properties jobInfo, boolean insert, boolean verify)  
jobInfo
```

Indicates a properties object that has the following keys defined with string objects as values.

ALARM_IF_FAIL

Indicates whether an alarm should be posted to the Event processor if the job fails or is terminated.

APPLICATION

Indicates the job's associated application.

AUTO_DELETE

Indicates whether the job should be automatically deleted after completion.

AUTO_HOLD

Indicates whether to change the job state to ON_HOLD when the box it is in begins RUNNING.

AVG_RUNTIME

Indicates the average runtime (in minutes) for newly submitted jobs into the database.

BOX_FAILURE

Specifies the conditions to be interpreted as a box failure, for example, failure(JobA) or failure(JobB) where JobA and JobB are in the box.

BOX_NAME

The name of the box to which this job belongs.

BOX_SUCCESS

Specifies the conditions to be interpreted as a box success, for example, success(JobA) or success(JobB) where JobA and JobB are in the box.

BOX_TERMINATOR

Indicates, with TRUE or FALSE, whether the box containing this job should be terminated if the job fails or terminates.

CHK_FILES

Specifies the minimum file space required on the designated file systems for the job to be started; for example, "/tmp 1024" indicates the /tmp file system and 1024 KB.

COMMAND

Indicates a job command, for example, "cd", "echo hello", and so on.

CONDITION

Indicates the conditions for the job to run (for example, "success(DB_BACKUP)", "exitcode (my_job=4)", "VALUE(TODAY)=Friday")

DATE_CONDITIONS

Indicates, with TRUE or FALSE, if there are date and time conditions to start this job.

DAYS_OF_WEEK

Indicates, with TRUE or FALSE, the days of the week when the job will be run.

DESCRIPTION

Specifies a description for the job; for documentation purpose only.

EXCLUDE_CALENDAR

Indicates the custom calendar to be used to determine the days of the week on which the job will not run.

GROUP

Indicates the job's associated group.

HEARTBEAT_INTERVAL

Specifies the frequency (in minutes) at which this job's command is expected to issue a heartbeat.

JOB_LOAD

Specifies the amount of processing power the job will consume.

JOB_NAME

Indicates the CA Workload Automation AE job name.

MACHINE

Indicates the machine on which the job is performed.

MAX_EXIT_SUCCESS

Specifies the maximum exit code with which the job can exit and still be considered a success.

MAX_RUN_ALARM

Specifies the maximum runtime (in minutes) that a job will require to finish normally.

MIN_RUN_ALARM

Specifies the minimum runtime (in minutes) that a job will require to finish normally.

N_RETRYs

Specifies the number of, if any, the job should be restarted after exiting with a FAILURE status.

NOTIFICATION_ID

Indicates the NSM Notification ID.

NOTIFICATION_MSG

Indicates the NSM Notification message.

OWNER

Indicates the CA Workload Automation AE job owner, for example, "testuser@TANT-A01"; "user1@machine1".

PERMISSION

Indicates the permission for this job.

PRIORITY

Specifies the queue priority of the job.

PROFILE

Specifies a job profile that defines the environment variables to be set before the specified command is executed.

RUN_CALENDAR

Indicates the custom calendar to be used to determine the days of the week on which a job will run.

RUN_WINDOW

Indicates the time span during which the job is allowed to start. If this attribute is specified, the job is eligible to run only during this time interval.

SEND_NOTIFICATION

Sends an NSM notification on job failure.

SERVICE_DESK

Creates a Service Desk issue on job failure.

START_MINS

Indicates the number of minutes past the hour, every hour, on the specified days or dates, when the job is started, for example, "20,40"; "20".

START_TIMES

Indicates the times of day, in 24-hour format, on the specified days or dates, when the job is started, for example, "12:00, 13:00, 14:15".

STD_ERR_FILE

Specifies the file to which the standard error file's output should be re-directed.

STD_IN_FILE

Specifies the file to which the standard input file for the job should be re-directed.

STD_OUT_FILE

Specifies the file to which the standard output file should be re-directed.

SVCDESK_ATTR

Indicates the Service Desk attribute.

SVCDESK_DESC

Indicates the Service Desk description.

SVCDESK_IMP

Indicates the Service Desk impact level as follows:

- 0—None
- 1—High
- 2—Medium-High
- 3—Medium
- 4—Medium-Low
- 5—Low

SVCDESK_PRI

Indicates the Service Desk priority as follows:

- 0—None
- 1—High
- 2—Medium-High
- 3—Medium
- 4—Medium-Low
- 5—Low

SVCDESK_SEV

Indicates the Service Desk severity as follows:

- 0—None
- 1—High
- 2—Medium-High
- 3—Medium
- 4—Medium-Low
- 5—Low

TERM_RUN_TIME

Specifies the maximum runtime (in minutes) that a job should require to finish normally.

TIMEZONE

Lets you schedule a job based on a time zone.

WATCH_FILE

Specifies the full path name of the file for which this file watcher job should watch.

WATCH_FILE_MIN_SIZE

Specifies the minimum size (in bytes) of the watch file to determine if enough data has been written to the file to consider it complete.

WATCH_FILE_INTERVAL

Specifies the interval (in seconds) at which the file watcher job will check for the existence and size of the file watched.

insert

Indicates as follows:

- True—Create a new job.
- False—Update the existing job.

verify

Indicates as follows:

- True—Validate the job conditions and reflect the results in the response
- False—Do not validate job conditions.

InsertCalendarDate

InsertCalendarDate creates a new standard calendar. InsertCalendarDate is used by autocal_asc.

InsertCalendarDate verifies create permissions. After the calendar is created, the API modifies audit information as appropriate.

Example: Sample Code for Executing the InsertCalendarDate API

```
private void sample(AsApi api, String calendarName) {  
    // Set up the date array for input.  
    long a_day = 86400000L;  
    Date now = new Date();  
    Date[] newDates = {  
        now,  
        new Date(now.getTime() + a_day),           // tomorrow  
        new Date(now.getTime() + (30 * a_day)), // 30 days from now  
        new Date(now.getTime() + (90 * a_day)) // 90 days from now  
    };  
  
    // Set up the request.  
    ICalendarDateReq req = new InsertCalendarDateReq();  
    req.setRequest(calendarName, newDates);  
  
    // Execute the request, check for errors.  
    try {  
        req.execute(api);  
        System.out.println("Dates successfully inserted.");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

InsertCalendarDateReq

extends

Cat1Request

implements

ICalendarDateReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—You do not have permission to insert or append dates to the named calendar.

InsertCalendarDateReq

The InsertCalendarDateReq constructor has the following syntax:

```
public InsertCalendarDateReq()
```

setRequest

The setRequest method in the InsertCalendarDateReq class defines the calendar name and the dates to add. This method has the following syntax:

```
void setRequest(java.lang.String calendarName, java.util.Date[] calendarDates)
```

calendarName

Defines the name of the calendar to create.

calendarDates

Defines the dates to add to or remove from the named calendar.

InsertCalendarDateRsp

extends

ApiResponse

The InsertCalendarDateRsp class comprises the response for the insert calendar date request. This response contains no special information.

InsertCalendarDateRsp

The InsertCalendarDateRsp constructor has the following syntax:

```
public InsertCalendarDateRsp(com.ca.autosys.services.JResponse rsp)
```

rsp Defines the raw response object.

The InsertCalendarDateRsp constructor throws java.lang.IllegalArgumentException if a null response is given.

InsertCycleCalendarDays

InsertCycleCalendarDays inserts cycle calendar days into the CA Workload Automation AE database.

Example: Sample Code for Executing the InsertCycleCalendarDays API

```
void sample(AsApi api, String calName) {
    InsertCycleCalendarDaysReq req = new InsertCycleCalendarDaysReq(calName);
    req.appendCycle(Tools.getDateFromString("08/08/2005 08:00"),
    Tools.getDateFromString("08/31/2005 17:00"));
    req.appendCycle(Tools.getDateFromString("12/01/2005 08:00"),
    Tools.getDateFromString("12/31/2005 17:00"));

    try {
        req.execute(api);
        System.out.println("Cycle calendar successfully inserted.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("General Error Encountered: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("Security Error Encountered: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("Autosys Error Encountered: " + asEx.getMessage());
    }
}
```

InsertCycleCalendarDaysReq

extends

Cat1Request

implements

IInsertCycleCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have authority to perform the insert.

InsertCycleCalendarDaysReq

The InsertCycleCalendarDaysReq constructor has the following syntax:

`InsertCycleCalendarDaysReq(java.lang.String calendarName)`

calendarName

Specifies the calendar name for this cycle calendar.

appendCycle

The appendCycle method in the InsertCycleCalendarDaysReq class appends date ranges to the cycle calendar. This method has the following syntax:

`void appendCycle(java.util.Date startDate, java.util.Date endDate)`

startDate

Specifies the date the cycle should start.

endDate

Specifies the date the cycle should end.

InsertCycleCalendarDaysRsp

extends

ApiResponse

implements

IInsertCycleCalendarDaysRsp

The InsertCycleCalendarDaysRsp class comprises the response for the insert cycle calendar days request. This response contains no special information. An exception occurs if an error is detected.

InsertCycleCalendarDaysRsp

The InsertCycleCalendarDaysRsp constructor has the following syntax:

```
public InsertCycleCalendarDaysRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

The InsertCycleCalendarDaysRsp constructor throws java.lang.IllegalArgumentException if the response is null.

InsertExtendedCalendarDays

InsertExtendedCalendarDays creates an extended calendar in the CA Workload Automation AE database.

Example: Sample Code for Executing the InsertExtendedCalendarDays API

```
void Sample(AsApi api, String server, int port, String calendar) {
    int[] workdays = {
        AsConstants.DAY_MONDAY,
        AsConstants.DAY_TUESDAY,
        AsConstants.DAY_WEDNESDAY,
        AsConstants.DAY_THURSDAY,
        AsConstants.DAY_FRIDAY
    };
    String holidayCalendar = "HolidaySample";
    String cycleCalendar = "CycleSample";
    int holidayAction = AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY;
    int weekendAction = AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY;
    int conditionAdjustment = -5;
    String conditionKeyword = "FOMWEEK";

    InsertExtendedCalendarDaysReq req = new InsertExtendedCalendarDaysReq();
    req.setRequest(calendar, workdays, holidayCalendar, cycleCalendar,
    holidayAction, weekendAction, conditionAdjustment, conditionKeyword);

    try {
        req.execute(api);
        System.out.println("Calendar " + calendar + " has been created.");
    } catch (AsSecurityException secEx) {
        System.out.println("\nSecurity Violation: " + secEx.getMessage().trim());
    } catch (AsGeneralErrorException genEx) {
        System.out.println("\nGeneral Error: " + genEx.getMessage().trim());
    } catch (AsException e) {
        System.out.println("\nAutoSys Error: " + e.getMessage().trim());
    }
}
```

[InsertExtendedCalendarDaysReq](#)

extends

Cat1Request

implements

IInsertExtendedCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have authority to insert the extended calendar.

[InsertExtendedCalendarDaysReq](#)

The InsertExtendedCalendarDaysReq constructor has the following syntax:

```
public InsertExtendedCalendarDaysReq()
```

setRequest

The setRequest method in the InsertExtendedCalendarDaysReq sets up the request to insert an extended calendar into the CA Workload Automation AE database. This method has the following syntax:

```
void setRequest(java.lang.String calendarName, int[] workDays, java.lang.String holidayCalendar, java.lang.String cycleCalendar, int holidayAction, int weekendAction, int conditionAdjustment, java.lang.String condition)
```

calendarName

Specifies the name of the calendar to be inserted.

workDays

Specifies the requested workdays. Possible values are:

- AsConstants.DAY_SUNDAY—Set Sunday as a workday.
- AsConstants.DAY_MONDAY—Set Monday as a workday.
- AsConstants.DAY_TUESDAY—Set Tuesday as a workday.
- AsConstants.DAY_WEDNESDAY—Set Wednesday as a workday.
- AsConstants.DAY_THURSDAY—Set Thursday as a workday.
- AsConstants.DAY_FRIDAY—Set Friday as a workday.
- AsConstants.DAY_SATURDAY—Set Saturday as a workday.

holidayCalendar

Specifies the name of the holiday calendar to use.

cycleCalendar

Specifies the name of the cycle calendar to use.

holidayAction

Specifies the action to take when the work falls on a holiday. Possible values are:

- AsConstants.CALENDAR_ACTION_DEFAULT—Do not schedule the job.
- AsConstants.CALENDAR_ACTION_ONLY—Schedule only on a holiday.
- AsConstants.CALENDAR_ACTION_SCHEDULE—Schedule always.
- AsConstants.CALENDAR_ACTION_NEXT_DAY—Schedule on the next day (day could be a holiday).
- AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY—Schedule on the next workday.
- AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY—Schedule on the previous workday.

weekendAction

Specifies the action to take when the work falls on a non-workday. Possible values are:

- AsConstants.CALENDAR_ACTION_DEFAULT—Schedule always.
- AsConstants.CALENDAR_ACTION_ONLY—Schedule only on a non-workday.
- AsConstants.CALENDAR_ACTION_NEXT_DAY—Schedule on the next day (day could be a non-workday).
- AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY—Schedule on the next workday.
- AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY—Schedule on the previous workday.

conditionAdjustment

Defines the days before or after the condition setting using a positive or negative number.

condition

Specifies the date keyword to be used as a scheduling condition.

Note: For more information about condition keywords, see the *Reference Guide*.

InsertExtendedCalendarDaysRsp

extends

ApiResponse

implements

IInsertExtendedCalendarDaysRsp

The InsertExtendedCalendarDaysRsp class comprises the response for the insert extended calendar days request. This response contains no special information. An exception occurs if an error is detected.

InsertExtendedCalendarDaysRsp

The InsertExtendedCalendarDaysRsp Constructor has the following syntax:

```
public InsertExtendedCalendarDaysRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the response object.

The InsertExtendedCalendarDaysRsp constructor throws java.lang.IllegalArgumentException if a null response is received.

InsertGbloc

InsertGbloc inserts global blobs into the CA Workload Automation AE database.

Example: Sample Code for Executing the InsertGbloc API

```
void sample(AsApi api, String globalBlobName) {  
    File blobFile = new File("C:\\\\Temp\\\\blobFile");  
    byte[] blobData = new byte[0];  
    InsertGblocReq req = new InsertGblocReq();  
    try {  
        // get the blob ready for passing to the api  
        int blobSize = (int) blobFile.length();  
        blobData = new byte[blobSize];  
        DataInputStream inStream = new DataInputStream(new  
FileInputStream(blobFile));  
        inStream.readFully(blobData);  
        inStream.close();  
  
        // get the request ready  
        req.setRequest(globalBlobName, blobData);  
  
        // execute the request and check for errors  
        req.execute(api);  
        System.out.println("\nBlob successfully inserted into the database.\n");  
    } catch (EOFException eofEx) {  
        System.out.println("EOF encountered while reading blob file.\n\n" +  
eofEx.getMessage());  
    } catch (IOException ioEx) {  
        System.out.println("Failed to read blob file.\n\n" + ioEx.getMessage());  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AutoSys General Error. " + genEx.getMessage());  
    } catch (AsNoAttributesException noAttEx) {  
        System.out.println("AutoSys No Attributes Error. " + noAttEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AutoSys Error. " + asEx.getMessage());  
    }  
}
```

InsertGblobReq

extends

Cat1Request

implements

IInsertGblobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsNoAttributesException—No blob data was provided for insertion.

InsertGblobReq

The InsertGblobReq constructor creates an InsertGblob object and has the following syntax:

```
public InsertGblobReq()
```

setRequest

The setRequest method in the InsertGblobReq class configures the request for execution. This method has the following syntax:

```
void setRequest(java.lang.String blobName, byte[] blobData)
```

blobName

Defines the name to assign to the global blob.

blobData

Defines the data that composes the global blob.

InsertGblobRsp

extends
 ApiResponse
implements
 IInsertGblobRsp

InsertGblobRsp is the response object for the InsertGblob API.

InsertGblobRsp

The InsertGblobRsp constructor creates an InsertGblobRsp object and has the following syntax:

```
public InsertGblobRsp(com.ca.autosys.services.JResponse response)
```

InsertJblob

InsertJblob ties a binary large object to an existing CA Workload Automation AE job.

Example: Sample Code for Executing the InsertJblob API

```
private void sample(AsApi api, String jobName) {
    File blobFile = new File("C:\\\\Temp\\\\blobFile");
    byte[] blobData = new byte[0];
    InsertJblobReq req = new InsertJblobReq();
    InsertJblobRsp rsp = null;

    try {
        // get the blob ready for passing to the api
        int blobSize = (int) blobFile.length();
        blobData = new byte[blobSize];
        DataInputStream inStream = new DataInputStream(new
FileInputStream(blobFile));
        inStream.readFully(blobData);
        inStream.close();

        // get the request ready
        req.setRequest(jobName, blobData);

        // execute the request and check for errors
        rsp = (InsertJblobRsp)req.execute(api);
        System.out.println("\nBlob version " + rsp.getBlobVersion() + " "
successfully inserted into job " + jobName + ".\n");
    } catch (EOFException eofEx) {
        System.out.println("EOF encountered while reading blob file.\n\n" +
eofEx.getMessage());
    } catch (IOException ioEx) {
        System.out.println("Failed to read blob file.\n\n" + ioEx.getMessage());
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AutoSys General Error. " + genEx.getMessage());
    } catch (AsNoAttributesException noAttEx) {
        System.out.println("AutoSys No Attributes Error. " + noAttEx.getMessage());
    } catch (AsBadAttributesException badAttEx) {
        System.out.println("AutoSys Bad Attributes Error. " +
badAttEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AutoSys Error. " + asEx.getMessage());
    }
}
```

InsertJblobReq

extends
Cat1Request
implements
IInsertJblobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsNoAttributesException—No BLOB data to insert.
- AsBadAttributesException—Invalid characters in jobname.
- AsObjectDoesNotExistException—Invalid jobname was passed.

InsertJblobReq

The InsertJblobReq constructor has the following syntax:

```
public InsertJblobReq()
```

setRequest

The setRequest method in the InsertJblobReq class configures the Insert JBLOB request for processing. This method has the following syntax:

```
void setRequest(java.lang.String jobName, byte[] blob)  
jobName
```

Specifies the job name to which the BLOB is to be tied.

blob

Specifies the Binary Large Object to tie to the job.

InsertJblobRsp

extends
 ApiResponse
implements
 IInsertJblobRsp

InsertJblobRsp is the response object for the InsertJblob API.

InsertJblobRsp

The InsertJblobRsp constructor has the following syntax:

```
public InsertJblobRsp(com.ca.autosys.services.JResponse response)
```

getBlobVersion

The getBlobVersion method in the InsertJblobRsp class returns the integer containing the version number of the blob. This method has the following syntax:

```
int getBlobVersion()
```

InsertJob

UpdateJob is the API for inserting jobs in the database. This is the preferred method of updating jobs at CA Workload Automation AE r11.3 and later.

Example: Sample Code for Executing the InsertJob API

```
private void sample(AsApi api) {
    InsertJobReq req = new InsertJobReq();
    InsertJobRsp rsp;

    try {
        // Set up the job definition.
        req.set(AsJobConstants.KEYWORD_JOB_NAME, "sample_job");
        req.set(AsJobConstants.KEYWORD_JOB_TYPE,
AsJobConstants.JOB_TYPE_COMMAND);
        req.set(AsJobConstants.KEYWORD_COMMAND, "myCommand");
        req.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        req.setRequest(false);

        // Insert the job.
        rsp = (InsertJobRsp)req.execute(api);

        // Check if there were undefined dependent jobs.
        if(rsp.hasUndefinedDependentJobs()) {
            String dependents[] = rsp.getUndefinedDependentJobs();
            System.out.println("Dependent Jobs: ");
            for(int i=0; i<dependents.length; ++i) {
                System.out.println(dependents[i]);
            }
        }

        // Check if there were warnings on the insert.
        if(rsp.hasWarnings()) {
            AsMessage msgs[] = rsp.getWarnings();
            System.out.println("Warnings: ");
            for(int i=0; i<msgs.length; ++i) {
                System.out.println(msgs[i]);
            }
        }
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

InsertJobReq

extends

Cat1Request

implements

IInsertJobReq

The execution of this request can raise the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to insert the named job.
- AsExternalValidationException—Failure in custom job validation.
- AsFieldValidationException—Field validation error.

InsertJobReq

The InsertJobReq constructor creates an InsertJobReq object. This constructor has the following syntax:

```
public InsertJobReq()
```

set

This set method in the InsertJobReq class configures an integer job parameter. This method has the following syntax:

```
public void set(int keyword, int value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the integer value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the InsertJobReq class configures a character job parameter. This method has the following syntax:

```
public void set(int keyword, char value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the character value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the InsertJobReq class configures a string job parameter. This method has the following syntax:

```
public void set(int keyword, java.lang.String value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the string value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the InsertJobReq class configures a BLOB job parameter. This method has the following syntax:

```
public void set(int keyword, byte[] value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the byte array BLOB value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the InsertJobReq class configures a Boolean job parameter. This method has the following syntax:

```
public void set(int keyword, boolean value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the Boolean value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the InsertJobReq class configures a double job parameter. This method has the following syntax:

```
public void set(int keyword, double value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the double value that keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

set

This set method in the InsertJobReq class configures a long job parameter. This method has the following syntax:

```
public void set(int keyword, long value)
```

keyword

Specifies the keyword constant to be set.

value

Specifies the long value that the keyword is to be set to.

An AsFieldValidationException is thrown for an invalid field specification.

setRequest

The setRequest method in the InsertJobReq class configures a request and gets it ready for execution. This method has the following syntax:

```
public void setRequest(boolean verify)
```

verify

Specifies the job validation as follows:

- True—Validates the job conditions and indicates the results in the response.
- False—Does not validate the job conditions.

add

The add method method in the InsertJobReq class adds repeating job parameters to the job. This method has the following syntax:

```
public void add(JobItem itemObj)
```

itemObj

Specifies the repeating job object to be added.

An AsFieldValidationException is thrown for an invalid field specification.

InsertJobRsp

extends

ApiResponse

implements

IInsertJobRsp

InsertJobRsp

The InsertJobRsp constructor creates an InsertJobRsp object. This constructor has the following syntax:

```
public InsertJobRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

getUndefinedDependentJobs

The method returns an array of job names specified in the job conditions that are undefined. This array will contain values only if hasUndefinedDependentJobs() returns TRUE. This method has the following syntax:

```
public java.lang.String[] getUndefinedDependentJobs()
```

hasUndefinedDependentJobs

This method returns whether the conditions defined in the request specify undefined jobs. This can be true only if the client specifies "verify" in the request. This method has the following syntax:

```
public boolean hasUndefinedDependentJobs()
```

InsertRealMachine

InsertRealMachine creates a real machine. This API is used by JIL to create real machines. The InsertRealMachine API verifies that the caller has create access to the named machine. It then inserts the machine and updates audit information.

Example: Sample code for executing the InsertRealMachine API

```
private void sample(AsApi api) {  
    // Set up the request.  
    String machineName = "sampleRealMachine";  
    IInsertRealMachineReq req = new InsertRealMachineReq();  
    req.setRequest(machineName, IInsertRealMachineReq.MR_UNIX, 1, 1);  
  
    // Execute the request, check for errors.  
    try {  
        req.execute(api);  
        System.out.println("Machine \\" + machineName + "\\ has been inserted.");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

InsertRealMachineReq

extends
Cat1Request
implements
IIInsertRealMachineReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to insert the named machine.

InsertRealMachineReq

The InsertRealMachineReq constructor has the following syntax:

```
public InsertRealMachineReq()
```

setRequest

This setRequest method in the InsertRealMachineReq class defines a new real machine. This method has the following syntax:

```
void setRequest(java.lang.String mach_name, int type, int max_load, double factor)
```

mach_name

Defines the name of the machine.

type

Defines the type of machine. Possible values are:

- MR_SYSTEM_AGENT—Defines a System Agent machine.
- MR_NT—Defines a Windows machine.
- MR_UNIX—Defines a UNIX machine.
- MR_UUJMA—Defines a Unicenter NSM or a Universal Job Management Agent machine.
- MR_CONNECT—Defines a CA Workload Automation AE Connect machine.
- MR_UNIX_VIRTUAL—Defines a virtual UNIX machine.
- MR_NT_LEGACY—Defines an NT legacy machine.
- MR_UNIX_LEGACY—Defines a UNIX legacy machine.

max_load

Defines the maximum load of the machine.

factor

Defines the factor for the machine. The factor is a multiplier used to weight the machine during load balancing.

[setRequest](#)

This `setRequest` method in the `InsertRealMachineReq` class configures the request. This method has the following syntax:

```
public void setRequest(java.lang.String mach_name)
```

mach_name

Defines the name of the machine.

[setType](#)

This `setType` method in the `InsertRealMachineReq` class configures the machine type. This method has the following syntax:

```
public void setType(int mach_type)
```

mach_type

Defines the type of machine. Possible values are:

- `MR_SYSTEM_AGENT`—System Agent machine.
- `MR_NT`—Windows machine.
- `MR_UNIX`—UNIX machine.
- `MR_UUJMA`—NSM or a Universal Job Management Agent machine.
- `MR_CONNECT`—CA Workload Automation AE Connect machine.
- `MR_UNIX_VIRTUAL`—Virtual UNIX machine.
- `MR_NT_LEGACY`—NT legacy machine.
- `MR_UNIX_LEGACY`—UNIX legacy machine.

[setMaxLoad](#)

This `setMaxLoad` method in the `InsertRealMachineReq` class defines the maximum load of the machine. This method has the following syntax:

```
public void setMaxLoad(int max_load)
```

max_load

Defines the maximum load of the machine.

setFactor

This setFactor method in the InsertRealMachineReq class defines the weight factor of the machine. This method has the following syntax:

```
public void setFactor(double factor)
```

factor

Defines the weight factor of the machine.

Limits: 0-1024

setDescription

This setDescription method in the InsertRealMachineReq class defines the description of the machine. This method has the following syntax:

```
public void setDescription(java.lang.String mach_desc)
```

mach_desc

Defines the description of the machine.

setnodeName

The setnodeName method in the InsertRealMachineReq class configures the node name when the agent is running. This method has the following syntax:

```
public void setnodeName(java.lang.String nodename)
```

nodename

Defines the node or host name where the agent runs.

setPort

The setPort method in the InsertRealMachineReq class configures the port number that the agent is listening on. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setPort(int port)
```

port

Defines the listening port of the agent.

setCharacterCode

The setCharacterCode method in the InsertRealMachineReq class configures the character encoding type. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setCharacterCode(int char_code)
```

char_code

Defines the type of encoding. Possible values are:

- CC_ASCII—ASCII encoding
- CC_EBCDIC—EBCDIC encoding

[setKeyToAgent](#)

The `setKeyToAgent` method in the `InsertRealMachineReq` class configures the encryption key used to send data to the agent. This method is valid only for `MR_SYSTEM_AGENT` type machines. This method has the following syntax:

```
public void setKeyToAgent(java.lang.String key)
```

key

Defines the encryption key.

[setEncryptionType](#)

The `setEncryptionType` method in the `InsertRealMachineReq` class configures the encryption algorithm type used by the agent. This method is valid only for `MR_SYSTEM_AGENT` type machines. This method has the following syntax:

```
public void setEncryptionType(int type)
```

type

Defines the encryption type. Possible values are:

- ENCRYPTION_TYPE_NONE—No encryption.
- ENCRYPTION_TYPE_DEFAULT—Use default encryption algorithm and key.
- ENCRYPTION_TYPE_AES—Use AES 128-bit encryption algorithm.

[setAgentName](#)

The `setAgentName` method in the `InsertRealMachineReq` class configures the internal name of the agent. This method is valid only for `MR_SYSTEM_AGENT` type machines. This method has the following syntax:

```
public void setAgentName(java.lang.String agent_name)
```

agent_name

Defines the internal name of the agent.

setHeartbeatFreq

The setHeartbeatFreq method in the InsertRealMachineReq class configures the heartbeat frequency. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setHeartbeatFreq(int freq)
```

freq

Defines the number of seconds between heartbeat polls.

setHeartbeatAttempts

The setHeartbeatAttempts method in the InsertRealMachineReq class configures the number of attempts to send a heartbeat. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setHeartbeatAttempts(int attempts)
```

attempts

Defines the number of attempts to send a heartbeat.

setProvision

The setProvision method in the InsertRealMachineReq class sets the flag to cause the agent to be provisioned automatically. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setProvision(boolean provision)
```

provision

Defines the flag to indicate whether the machine should be provisioned.

setAdministrator

The setAdministrator method in the InsertRealMachineReq class configures the administrator ID authorized to provision the agent. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setAdministrator(java.lang.String admin)
```

admin

Defines the authorized administrator ID.

setOpSys

The setOpSys method in the InsertRealMachineReq class configures the identify the operating system running the agent. This method is valid only for MR_SYSTEM_AGENT type machines. This method has the following syntax:

```
public void setOpSys(int opsys)  
opsys
```

Defines the operating system. Possible values are:

- OPSYS_AIX—AIX
- OPSYS_HPUX—HP-UX
- OPSYS_I5OS—i5/OS
- OPSYS_LINUX—Linux
- OPSYS_OPENVMS—OpenVMS
- OPSYS_SOLARIS—Solaris
- OPSYS_TANDEM—Tandem
- OPSYS_WINDOWS—Windows
- OPSYS_ZOS—z/OS

InsertRealMachineRsp

extends

ApiResponse

The InsertRealMachineRsp class comprises the response for the insert real machine request. This response contains no special information. An exception occurs if an error is detected.

InsertRealMachineRsp

The InsertRealMachineRsp constructor has the following syntax:

```
public InsertRealMachineRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

The InsertRealMachineRsp constructor throws java.lang.IllegalArgumentException if a null response is given.

InsertResource

InsertResource creates or updates a resource on a machine. The InsertResource API verifies user permissions. After the resource is inserted on a given machine, the API modifies audit information as appropriate.

Example: Sample code for executing the InsertResource API

```
private void sample(AsApi api) {
    // Set up the request.
    String RESOURCE_NAME = "sampleResource";
    String MACHINE_NAME = "sampleMachine";
    IInsertResourceReq req = new InsertResourceReq();
    req.setName(RESOURCE_NAME);
    req.setDescription("");
    req.setMachine(MACHINE_NAME);
    req.setAmount(5);
    req.setType(1);
    req.setRequest();
    req.setUser(TEST_USER, SERVER_NAME);

    // Execute the request, check for errors.
    try {
        req.execute(api);
        System.out.println("Resource " + RESOURCE_NAME + " has been inserted on
Machine " + MACHINE_NAME);
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

InsertResourceReq

extends
Cat1Request
implements
IInsertResourceReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to insert the resource.

InsertResourceReq

The InsertResourceReq constructor creates a InsertResourceReq object. This constructor has the following syntax:

```
public InsertResourceReq()
```

setRequest

The setRequest method in the InsertResourceReq class configures the insert resource request for processing. This method has the following syntax:

```
public void setRequest()
```

setName

The setName method in the InsertResourceReq class is a setter method for the resource name. This method has the following syntax:

```
public void setName(java.lang.String resource_name)
```

resource_name

Specifies the name of the resource.

setDescription

The setDescription method in the InsertResourceReq class is a setter method for the resource description. This method has the following syntax:

```
public void setDescription(java.lang.String resource_description)
```

resource_description

Specifies the description of the resource.

setMachine

The setMachine method in the InsertResourceReq class is a setter method for the machine name on which the resource should be inserted. This method has the following syntax:

```
public void setMachine(java.lang.String resource_machine)
```

resource_machine

Specifies the name of the machine.

setType

The setType method in the InsertResourceReq class is a setter method for the resource type. This method has the following syntax:

```
public void setType(int resource_type)
```

resource_type

Specifies the type of the resource.

setAmount

The setAmount method in the InsertResourceReq class is a setter method for resource amount. This method has the following syntax:

```
public void setAmount(int resource_amount)
```

resource_amount

Specifies the amount of the resource.

InsertResourceRsp

extends

ApiResponse

implements

IInsertResourceRsp

The InsertResourceRsp constructor creates a response object for the InsertResource API. This constructor has the following syntax:

```
public InsertResourceRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

InsertUpdateDeleteJobType

InsertUpdateDeleteJobType inserts, updates, and deletes job types in the CA Workload Automation AE database.

Example: Sample Code for Executing the InsertUpdateDeleteJobType API

```
void sample( AsApi api ) {
    char jobType = '0';
    String jobTypeCommand1 = "notepad.exe";
    String jobTypeCommand2 = "wordpad.exe";
    String jobTypeDescription = "Text Editor";
    IInsertUpdateDeleteJobTypeReq req = null;

    try {
        // setup the request for an insert and execute it
        req = new InsertUpdateDeleteJobTypeReq();
        req.setInsertRequest(jobType, jobTypeCommand1, jobTypeDescription );
        req.execute( api );
        System.out.println( "Job Type " + jobType +
            " inserted into the database with job command of \\" + 
            jobTypeCommand1 + "\"" + " and a description of \\" + 
            jobTypeDescription + "\"." );

        // redo the request as an update
        req = new InsertUpdateDeleteJobTypeReq();
        req.setUpdateRequest( jobType, jobTypeCommand2, jobTypeDescription );
        req.execute( api );
        System.out.println( "Job Type " + jobType +
            " updated in the database with job command of \\" + 
            jobTypeCommand2 + "\"." );

        // now, delete the job type
        req = new InsertUpdateDeleteJobTypeReq();
        req.setDeleteRequest( jobType );
        req.execute( api );
        System.out.println( "Job Type " + jobType +
            " deleted from the database." );

    } catch ( AsGeneralErrorException genEx ) {
        System.out.println( "AsGeneralErrorException: " +
            genEx.getMessage() );
    } catch ( AsException asEx ) {
        System.out.println( "AsException: " + asEx.getMessage() );
    }
}
```

InsertUpdateDeleteJobTypeReq

extends

Cat1Request

implements

IInsertUpdateDeleteJobTypeReq

Execution of this request can result in AsGeneralErrorException being thrown.

setInsertRequest

The setInsertRequest method in the InsertUpdateDeleteJobTypeReq class configures the request for an insert operation. This method has the following syntax:

```
void setInsertRequest(char jobType, java.lang.String jobTypeCommand,  
java.lang.String jobTypeDescription)
```

jobType

Specifies the job type to insert. Characters a through z are reserved for CA Workload Automation AE usage.

Limits: 0 - 9

jobTypeCommand

Specifies the command associated with the job type.

jobTypeDescription

Specifies the description associated with the job type.

setUpdateRequest

The setUpdateRequest method in the InsertUpdateDeleteJobTypeReq class configures the request for an update operation. This method has the following syntax:

```
public void setUpdateRequest(char jobType, java.lang.String jobTypeCommand,  
java.lang.String jobTypeDescription)
```

jobType

Specifies the job type to update. Characters a through z are reserved for CA Workload Automation AE usage.

Limits: 0 - 9

jobTypeCommand

Specifies the command associated with the job type.

jobTypeDescription

Specifies the description associated with the job type.

setDeleteRequest

The `setDeleteRequest` method in the `InsertUpdateDeleteJobTypeReq` class configures the request for a delete operation. This method has the following syntax:

```
void setDeleteRequest(char jobType)
```

jobType

Specifies the job type to delete. Characters a through z are reserved for CA Workload Automation AE usage.

Limits: 0 - 9

[InsertUpdateDeleteJobTypeRsp](#)

extends

`ApiResponse`

implements

`IInsertUpdateDeleteJobTypeRsp`

`InsertUpdateDeleteJobTypeRsp` is the response object for the `InsertUpdateDeleteJobType` API. This response contains no special information. An exception occurs if an error is detected.

InsertUpdateExternalInstance

InsertUpdateExternalInstance inserts or updates external instance entries in the CA Workload Automation AE database.

Example: Sample Code for Executing the InsertUpdateExternalInstance API

```
void sample(AsApi api, String instanceName) {
    InsertUpdateExternalInstanceReq req =
        new InsertUpdateExternalInstanceReq();
    String instanceType = AsConstants.EXTERNAL_INSTANCE_TYPE_APPSERVER;
    String[] instanceMachines = {
        "machineA:9000",
        "machineB:9001",
        "machineC:9002"
    };
    int requestType = AsConstants.EXTERNAL_INSTANCE_INSERT;

    req.setRequest( instanceName, instanceType, instanceMachines,
                    requestType );

    try {
        req.execute( api );
        System.out.println( "Instance successfully inserted." );
    } catch ( AsGeneralErrorException genEx ) {
        System.out.println( "AsGeneralErrorException Caught: " +
                           genEx.getMessage() );
    } catch ( AsException asEx ) {
        System.out.println( "AsException Caught: " + asEx.getMessage() );
    }
}
```

InsertUpdateExternalInstanceReq

extends

Cat1Request

implements

IInsertUpdateExternalInstanceReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The `setRequest` method in the `InsertUpdateExternalInstanceReq` class configures the request for execution. This method has the following syntax:

```
void setRequest(java.lang.String instanceName, java.lang.String instanceType,  
java.lang.String[] instanceMachines, int requestType)
```

instanceName

Specifies the external instance name to be updated or inserted. For example, ACE | CA7 | JMO.

instanceType

Specifies the instance type to be updated or inserted. Possible values are:

AsConstants.EXTERNAL_INSTANCE_TYPE_APPSERVER

Identifies an Application Server instance.

AsConstants.EXTERNAL_INSTANCE_TYPE_CONNECT

Identifies a CA Workload Automation AE Connect instance.

AsConstants.EXTERNAL_INSTANCE_TYPE_WORKLOAD

Identifies a Unicenter Workload instance.

instanceMachines

Specifies the strings containing xmachine attribute entries. For example:

Application Server instance

- machineA:9000
- machineB:9001

CA Workload Automation AE Connect or Unicenter Workload instance

- machineA
- machineB

requestType

Specifies the request type. Possible values are:

AsConstants.EXTERNAL_INSTANCE_INSERT

Specifies an insert request.

AsConstants.EXTERNAL_INSTANCE_UPDATE

Specifies an update request.

InsertUpdateExternalInstanceRsp

extends

IApiResponse

implements

IInsertUpdateExternalInstanceRsp

InsertUpdateExternalInstanceRsp exposes methods for the InsertUpdateExternalInstance response. This response contains no special information. An exception occurs if an error is detected.

InsertUpdateJob

InsertUpdateJob creates or updates a job in the database. InsertUpdateJob is used by JIL for insert_job and update_job.

InsertUpdateJob performs external job validation (if in place), then verifies security permissions. If InsertUpdateJob is being used to update a job, it verifies write permissions for the job. If InsertUpdateJob is being used to insert a new job, it verifies create permissions. You must also have execute permissions on the specified machine, the specified owner, the run calendar, and the exclude calendar.

InsertUpdateJob checks that the keywords specified for the job being created or updated are valid. For example, if you specify that the job is a command job, then you must have a command field. InsertUpdateJob then inserts the job, issues future STARTJOB events if required, and updates audit information.

Example: Sample Code for Executing the InsertUpdateJob API

```
private void sample(AsApi api) {
    // Set up the job parameters.
    Properties jobInfo = new Properties();
    jobInfo.put(IInsertUpdateJobReq.JOB_NAME, "test_job");
    jobInfo.put(IInsertUpdateJobReq.JOB_TYPE, "c");
    jobInfo.put(IInsertUpdateJobReq.COMMAND, "notepad.exe");
    jobInfo.put(IInsertUpdateJobReq.MACHINE, "test_machine");

    // Set up the request.
    IInsertUpdateJobReq req = new InsertUpdateJobReq();
    req.setRequest(jobInfo, true, true);

    // Execute the request, check for errors.
    try {
        req.execute(api);

        System.out.println("Job \""
        jobInfo.getProperty(IInsertUpdateJobReq.JOB_NAME) + "\" successfully inserted.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsExternalValidationException extValidEx) {
        System.out.println("AsExternalValidationException: " +
extValidEx.getMessage());
    } catch (AsFieldValidationException fldValidEx) {
        System.out.println("AsFieldValidationException: " +
fldValidEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

InsertUpdateJobReq

```
extends  
    JobReq  
implements  
    IInsertUpdateJobReq
```

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - You do not have permission to insert or update the named job.
- AsExternalValidationException - Failure in custom job validation.
- AsFieldValidationException - Field validation error.

setRequest

The setRequest method in the InsertUpdateJobReq class specifies the conditions for the request to insert or update a job. This method has the following syntax:

```
void setRequest(java.util.Properties JobInfo, boolean insert, boolean verify)
```

JobInfo

Specifies a Properties object that has the following keys defined with String objects as values:

alarm_if_fail

Specifies whether an alarm should be posted to the event processor if the job fails or is terminated. (true or false)

application

Specifies the application associated with the job.

auto_delete

Specifies whether the job should be automatically deleted after completion. (true or false.)

auto_hold

Specifies whether to change the job state to ON_HOLD when the box it is in begins RUNNING. (true or false)

avg_runtime

Specifies an average run time (in minutes) for a job that is newly submitted to the database.

box_failure

Specifies the conditions to be interpreted as a box failure. (for example, failure(JobA) OR failure(JobB) where JobA and JobB are in the box.)

box_name

Specifies the name of the box to which this job belongs.

box_success

Specifies the conditions to be interpreted as a box success. For example, success(JobA) OR success(JobB) where JobA and JobB are in the box.

box_terminator

Specifies whether the box containing this job should be terminated if the job fails or terminates. (true or false)

chk_files

Specifies the minimum amount of file space that must be available on designated file systems for the job to be started. For example, /tmp 1024 specifies the /tmp file system and 1024 KB.

command

Specifies the name and arguments of any command, executable, UNIX script or batch file to associate with the job.

condition

Defines conditions for a job to run. For example, success(DB_BACKUP), exitcode (my_job=4), VALUE(TODAY)=Friday.

date_conditions

Specifies whether or not there are date or time conditions for starting this job. (true or false)

days_of_week

Specifies the days of the week on which to run the job. For example, mo,tu,we,fr.

description

Specifies a description for the job; for documentation purposes only.

exclude_calendar

Specifies the name of the custom calendar to be used for determining the days of the week on which to not run this job.

group

Specifies the group the associated with the job.

heartbeat_interval

Specifies the frequency (in minutes) at which this command is expected to issue a heartbeat.

job_load

Specifies the relative amount of processing power the job consumes.

job_name

Specifies the name of the job to insert or update.

machine

Specifies the Client machine on which to run the job.

max_exit_success

Specifies the maximum exit code with which the job can exit and still be considered a success.

max_run_alarm

Specifies the maximum run time (in minutes) that a job should require to finish normally.

min_run_alarm

Specifies the minimum run time (in minutes) that a job should require to finish normally.

n_retrys

Specifies how many times, if any, to restart the job after exiting with a FAILURE status.

notification_id

Specifies the NSM Notification ID.

notification_msg

Specifies the NSM Notification message.

owner

Specifies the owner of the job.

permission

Specifies the permission for this job.

priority

Specifies the queue priority of the job.

profile

Specifies a job profile that defines environment variables to be set before the specified command is executed.

run_calendar

Specifies the name of the custom calendar to use when determining the days of the week on which to run a job.

run_window

Specifies the time span during which the job can start. If this attribute is specified, then when the job is eligible to run.

send_notification

Sends NSM notification on job failure.

service_desk

Creates a Service Desk issue on job failure.

start_mins

Specifies the number of minutes after the hour, every hour, to start the job on the specified days. For example, 20,40; 20.

start_time

Specifies the times of day, in 24-hour format, on the specified days or dates, to start the job. For example, 12:00, 13:00, 14:15.

std_err_file

Specifies the file to which the standard error file output should be re-directed.

std_in_file

Specifies the file to which the standard input file for the job should be re-directed.

std_out_file

Specifies the file to which the standard output file should be re-directed.

svcdesk_attr

Specifies the Service Desk attribute.

svcdesk_imp

Specifies the Service Desk impact level using one of the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

svcdesk_pri

Specifies the Service Desk priority using one of the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

svcdesk_sev

Specifies the Service Desk severity using one of the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

term_run_time

Specifies the maximum run time (in minutes) that a job should require to finish normally.

timezone

Specifies the time zone on which to schedule a job.

watch_file

Specifies the full path name of the file for which this file watcher job should watch.

watch_file_min_size

Specifies the watch file minimum size (in bytes), which determines when enough data has been written to the file to consider it complete.

watch_file_interval

Specifies the interval (in seconds) at which the file watcher job checks for the existence and size of the watched-for file.

insert

Specifies whether the API is intended to update or insert a job. Possible values are:

true

Creates a new job.

false

Updates the existing job.

verify

Specifies whether the API should validate the job conditions. Possible values are:

true

Validates the job conditions and reflects results in the response.

false

Does not validate job conditions.

[InsertUpdateJobRsp](#)

extends

ApiResponse

implements

IInsertUpdateJobRsp

[getUndefinedDependentJobs](#)

The `getUndefinedDependentJobs` method in the `InsertUpdateJobRsp` class returns an array of the names of undefined jobs specified in the conditions defined in the `JobInfo` parameter of the `setRequest` method in the `InsertUpdateJobReq` class. This method has the following syntax:

```
java.lang.String[] getUndefinedDependentJobs()
```

hasUndefinedDependentJobs

The hasUndefinedDependentJobs method in the InsertUpdateJobRsp class reports whether the conditions defined in the JobInfo parameter of the setRequest method in the InsertUpdateJobReq class specify undefined jobs. This can only be true if the Client specified to verify in the request. This method has the following syntax:

```
boolean hasUndefinedDependentJobs()
```

This method returns one of the following values:

true

Indicates that the job conditions specify undefined jobs.

false

Indicates that the job conditions do not specify undefined jobs.

InsertVirtualMachine

InsertVirtualMachine creates a virtual machine. InsertVirtualMachine is used by JIL to insert virtual machines.

InsertVirtualMachine verifies create access to the named machine. InsertVirtualMachine then inserts the machine and updates audit information.

Example: Sample Code for Executing the InsertVirtualMachine API

```
private void sample(AsApi api) {
    // Set up the component machines.
    ComponentMachine mach1 = new ComponentMachine("sampleCompMach1",1,1);
    ComponentMachine mach2 = new ComponentMachine("sampleCompMach2",1,1);
    ComponentMachine mach3 = new ComponentMachine("sampleCompMach3",1,1);
    ComponentMachine[] compMachines = {
        mach1, mach2, mach3
    };

    // Set up the request.
    String sampleMachine = "sampleVirtualMachine";
    InsertVirtualMachineReq req = new InsertVirtualMachineReq();
    req.setRequest(sampleMachine, IInsertVirtualMachineReq.MR_UNIX,
compMachines);

    // Execute the request, check for errors.
    try {
        req.execute(api);
        System.out.println("Virtual machine \"" + sampleMachine + "\" has been
inserted.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

InsertVirtualMachineReq

extends
Cat1Request
implements
IInsertVirtualMachineReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have permission to insert the named machine.

setRequest

The setRequest method in the InsertVirtualMachineReq class specifies a virtual machine. This method has the following syntax:

```
void setRequest(java.lang.String machineName, int type, ComponentMachine[] machines)  
machineName
```

Specifies the name of the virtual machine to create.

type

Specifies the type of the machine. Possible values are:

MR_NT

Defines a Windows virtual machine.

MR_UNIX

Defines a UNIX virtual machine.

machines

Specifies an array of machines that comprise the virtual machine.

InsertVirtualMachineRsp

extends
ApiResponse

The InsertVirtualMachineRsp class comprises the response for the insert virtual machine request. This response contains no special information. An exception occurs if an error is detected.

ComponentMachine

extends

`java.lang.Object`

The ComponentMachine API is a simple utility class used to define virtual machines for the InsertVirtualMachine API.

getMachineName

The getMachineName method in the ComponentMachine class returns the name of the real component machine. This method has the following syntax:

```
java.lang.String getMachineName()
```

IntCode

extends
java.lang.Object

The IntCode class represents an intcode.

IntCode

This constructor constructs an IntCode object with known values. This constructor has the following syntax:

```
public IntCode(int code, java.lang.String type, java.lang.String text)
```

code

Specifies the intcode.

type

Specifies the intcode type.

text

Specifies the intcode text.

getIntCode

This method gets the code for this object. This method has the following syntax:

```
public int getIntCode()
```

This method returns the IntCode stored in this object.

getCodeType

This method gets the code type for this object. This method has the following syntax:

```
public java.lang.String getCodeType()
```

This method returns the CodeType stored in this object.

getCodeText

This method gets the text for the object. This method has the following syntax:

```
public java.lang.String getCodeText()
```

This method returns the CodeText stored in this object.

IOverrideJobReq

extends

IJobReq

This interface exposes methods for the override job request.

setRequest

This method configures a request. This method has the following syntax:

```
void setRequest (java.util.Properties OverrideInfo)
```

OverrideInfo

Indicates a Properties object that has the following keys defined with String objects as values:

ALARM_IF_FAIL

Indicates, with TRUE or FALSE, whether an alarm should be posted to the Event processor if the job fails or is terminated.

APPLICATION

Indicates the job's associated application.

AUTO_DELETE

Indicates whether the job should be automatically deleted after completion. (e.g. true or false.)

AUTO_HOLD

Indicates whether to change the job state to ON_HOLD when the box it is in begins RUNNING. (e.g. true or false)

AVG_RUNTIME

Indicates an average runtime (in minutes) for a job that is newly submitted to the database.

BOX_FAILURE

Specifies the conditions to be interpreted as a box failure. (e.g. failure(JobA) OR failure(JobB) where JobA and JobB are in the box.

BOX_NAME

The name of the box to which this job belongs.

BOX_SUCCESS

Specifies the conditions to be interpreted as a box success. (e.g. success(JobA) OR success(JobB) where JobA and JobB are in the box.

BOX_TERMINATOR

Indicates whether the box containing this job should be terminated if the job fails or terminates. (e.g. true or false)

CHK_FILES

This resource check specifies the minimum amount of file space that must be available on designated file systems for the job to be started. (e.g. "/tmp 1024" this specifies the /tmp file system and 1024 KB)

COMMAND

Job command (e.g., "cd", "echo hello").

CONDITION

Conditions for job to run (e.g., "success(DB_BACKUP)", "exitcode (my_job=4)", "VALUE(TODAY)=Friday")

DATE_CONDITIONS

Indicates whether or not there are date or time conditions for starting this job. (e.g. true or false)

DAYS_OF_WEEK

Days of the week where the job will be run.(e.g., "mo,tu,we,fr")

DESCRIPTION

Specifies a description for the job; for documentation purposes only.

EXCLUDE_CALENDAR

Indicates the name of the custom calendar to be used for determining the days of the week on which this job will not run.

GROUP

Indicates the job's associated group.

HEARTBEAT_INTERVAL

Specifies the frequency (in minutes) at which this job's command is expected to issue a heart beat.

JOB_LOAD

Specifies the relative amount of processing power the job will consume.

MACHINE

Machine where job is performed.

MAX_EXIT_SUCCESS

Specifies the maximum exit code with which the job can exit and still be considered a success.

MAX_RUN_ALARM

Specifies the maximum runtime (in minutes) that a job should require to finish normally.

MIN_RUN_ALARM

Specifies the minimum runtime (in minutes) that a job should require to finish normally.

N_RETRY

Specifies how many times, if any, the job should be restarted after exiting with a FAILURE status.

NOTIFICATION_ID

NSM Notification ID

NOTIFICATION_MSG

NSM Notification message.

OWNER

CA Workload Automation AE job owner (e.g. "testuser@TANT-A01"; "user1@machine1").

PERMISSION

The permission for this job.

PRIORITY

Specifies the queue priority of the job.

PROFILE

Specifies a job profile that defines environment variables to be set before the specified command is executed.

RUN_CALENDAR

Indicates the name of the custom calendar to be used when determining the days of the week on which a job will run.

RUN_WINDOW

Indicates the time span during which the job is allowed to start. If this attribute is specified, the job is eligible to run only during this time interval.

SEND_NOTIFICATION

Send NSM notification on job failure.

SERVICE_DESK

Create a Service Desk issue on job failure.

START_MINS

The number of minutes past the hour, every hour, on the specified days or dates, when the job will be started (e.g., "20,40"; "20").

START_TIMES

Indicates the times of day, in 24-hour format, on the specified days or dates, when the job will be started. (e.g. "12:00, 13:00, 14:15")

STD_ERR_FILE

Specifies the file to which the standard error file's output should be re-directed.

STD_IN_FILE

Specifies the file to which the standard input file for the job should be re-directed.

STD_OUT_FILE

Specifies the file to which the standard output file should be re-directed.

SVCDESK_ATTR

Service Desk attribute

SVCDESK_IMP

Service Desk impact level. Can have the following values:

0: None

1: High

2: Medium-High

3: Medium

4: Medium-Low

5: Low

SVCDESK_PRI

Service Desk priority. Can have the following values:

0: None

1: High

2: Medium-High

3: Medium

4: Medium-Low

5: Low

SVCDESK_SEV

Service Desk severity: Can have the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

TERM_RUN_TIME

Specifies the maximum runtime (in minutes) that a job should require to finish normally.

TIMEZONE

Allows you to schedule a job based on a chosen time zone.

WATCH_FILE

Specifies the full path name of the file for which this file watcher job should watch

WATCH_FILE_MIN_SIZE

Specifies the watch file minimum size (in bytes), which determines when enough data has been written to the file to consider it complete.

WATCH_FILE_INTERVAL

Specifies the interval (in seconds) at which the file watcher job will check for the existence and size of the watched-for file.

J2eeParameterItem

extends

JobItem

J2eeParameterItem is the container class for specifying J2EE parameter arguments in job definitions. This JobItem subclass directly maps to the j2ee_parameter jil keyword.

Example: Sample Code for Using the J2eeParameterItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_J2EE_JMS_PUBLISH);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_CONNECTION_FACTORY, "myFactory");
        insertReq.set(AsJobConstants.KEYWORD_DESTINATION_NAME, "myDestination");
        insertReq.set(AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,
        "myContext");
        insertReq.set(AsJobConstants.KEYWORD_MESSAGE_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_PROVIDER_URL, "myURL");
        insertReq.set(AsJobConstants.KEYWORD_USE_TOPIC, true);

        // Set up the j2ee_parameter part of the job definition and
        // attach it to the base job definition.
        J2eeParameterItem j2parm = new J2eeParameterItem();
        j2parm.set(AsJobConstants.KEYWORD_J2EE_PARAMETER, "parm1=val1");
        j2parm.set(AsJobConstants.KEYWORD_J2EE_PARAMETER, "parm2=val2");
        j2parm.set(AsJobConstants.KEYWORD_J2EE_PARAMETER, "parm3=val3");
        insertReq.add(j2parm);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_CONNECTION_FACTORY,  
        AsJobConstants.KEYWORD_DESTINATION_NAME,  
        AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,  
        AsJobConstants.KEYWORD_MESSAGE_CLASS,  
        AsJobConstants.KEYWORD_PROVIDER_URL,  
        AsJobConstants.KEYWORD_USE_TOPIC,  
        AsJobConstants.KEYWORD_J2EE_PARAMETER  
    };  
  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();  
  
        // Process job level properties.  
        System.out.println("----- Begin Response -----");  
        for(int i=0; i<rspProperties.length; ++i) {  
            String keyword =  
                jobProperties.getKeywordFromConstant(rspProperties[i]);  
            switch(getRsp.getAttributeType(rspProperties[i])) {  
                case AsConstants.ATTRIBUTE_TYPE_STRING:  
                    System.out.println(keyword + ": " +  
                        getRsp.getString(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_INT:  
                    System.out.println(keyword + ": " +  
                        getRsp.getInt(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                    System.out.println(keyword + ": " +  
                        getRsp.getBoolean(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_LONG:
```

```

        System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
        System.out.println("\n----- Begin " + keyword + " JobItem
-----");
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {
            System.out.print(keyword + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
    case AsConstants.ATTRIBUTE_TYPE_STRING:

        System.out.println(ji.getString(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_INT:

        System.out.println(ji.getInt(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:

        System.out.println(ji.getBool(itemProperties[j]));
        break;
    }
}
        System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
}
        System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
}

```

Sample output

```

----- Begin Response -----
job_name: j2ee_parameter_sample
alarm_if_fail: true
machine: localhost
connection_factory: myFactory
destination_name: myDestination

```

```
initial_context_factory: myContext
message_class: myClass
provider_url: myURL
use_topic: true

----- Begin j2ee_parameter JobItem -----
j2ee_parameter: parm1=val1
j2ee_parameter: parm2=val2
j2ee_parameter: parm3=val3
----- End j2ee_parameter JobItem -----

----- End Response -----
```

[J2eeParameterItem](#)

This constructor has the following syntax:

```
public J2eeParameterItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- `add`
- `getAttributes`
- `getBool`
- `getInt`
- `getItemType`
- `getJobItem`
- `getLoggableItem`
- `getString`
- `hasSubItems`
- `set(int keyword, boolean value)`
- `set(int keyword, int value)`
- `set(int keyword, java.lang.String value)`
- `toString`

JmxParameterItem

extends

JobItem

Container class for specifying JMX parameter arguments in job definitions. This JobItem subclass directly maps to the jmx_parameter jil keyword.

Example: Sample Code for Using the JmxParameterItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_J2EE_JMX_MBEAN_CREATE);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_CLASS_NAME, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_MBEAN_NAME, "myBeanName");
        insertReq.set(AsJobConstants.KEYWORD_URL, "myURL");
        // Set up the jmx_parameter part of the job definition and
        // attach it to the base job definition.
        JmxParameterItem jpi = new JmxParameterItem();
        jpi.set(AsJobConstants.KEYWORD_JMX_PARAMETER, "parm1=val1");
        jpi.set(AsJobConstants.KEYWORD_JMX_PARAMETER, "parm2=val2");
        jpi.set(AsJobConstants.KEYWORD_JMX_PARAMETER, "parm3=val3");
        insertReq.add(jpi);
        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
        JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);
```

```
// Set up what attributes are to be returned.  
int attributes[] = {  
    AsJobConstants.KEYWORD_JOB_NAME,  
    AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
    AsJobConstants.KEYWORD_MACHINE,  
    AsJobConstants.KEYWORD_CLASS_NAME,  
    AsJobConstants.KEYWORD_MBEAN_NAME,  
    AsJobConstants.KEYWORD_URL,  
    AsJobConstants.KEYWORD_JMX_PARAMETER  
};  
// Ready the request and execute it.  
getReq.setRequest(filter, attributes);  
set = getReq.execute(appServer);  
  
// Process the returned responses.  
while(set.hasNext()) {  
    getRsp = (GetJobsWithFilterRsp)set.next();  
    int rspProperties[] = getRsp.getAttributes();
```

```

// Process job level properties.
System.out.println("----- Begin Response -----");
for(int i=0; i<rspProperties.length; ++i) {
    String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
    switch(getRsp.getAttributeType(rspProperties[i])) {
        case AsConstants.ATTRIBUTE_TYPE_STRING:
            System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_INT:
            System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
            System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_LONG:
            System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
            System.out.println("\n----- Begin " + keyword + " JobItem
-----");
            JobItem ji = getRsp.getJobItem(rspProperties[i]);
            int itemProperties[] = ji.getAttributes();

            // Process job item level properties.
            for(int j=0; j<itemProperties.length; ++j) {
                System.out.print(keyword + ": ");

                switch(jobProperties.getKeywordDataType(itemProperties[j])) {
                    case AsConstants.ATTRIBUTE_TYPE_STRING:
                        System.out.println(ji.getString(itemProperties[j]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_INT:
                        System.out.println(ji.getInt(itemProperties[j]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                        System.out.println(ji.getBool(itemProperties[j]));
                        break;
                }
            }
            System.out.println("----- End " + keyword + " JobItem
-----\n");
    }
}

```

```
        break;
    }
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: jmx_parameter_sample
alarm_if_fail: true
machine: localhost
class_name: myClass
mbean_name: myBeanName
url: myURL

----- Begin jmx_parameter JobItem -----
jmx_parameter: parm1=val1
jmx_parameter: parm2=val2
jmx_parameter: parm3=val3
----- End jmx_parameter JobItem -----

----- End Response -----
```

JmxParameterItem

This constructor has the following syntax:

```
public JmxParameterItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

JobFilterSort

extends

FilterSort

implements

IJobFilterSort

The JobFilterSort object allows sort criteria to be added to filtering API requests.

JobFilterSort

This JobFilterSort constructor creates a JobFilterSort object. This constructor has the following syntax:

```
public JobFilterSort()
```

JobFilterSort

This JobFilterSort constructor creates a JobFilterSort object with a known sort order and sort attribute. This constructor has the following syntax:

```
public JobFilterSort(int sortOrder, int sortAttribute)
```

sortOrder

Defines the sort order. Possible values are:

- ORDER_ASCENDING
- ORDER_DESCENDING

sortAttribute

Defines the attribute to sort by.

getSortKeys

The getSortKeys method in the JobFilterSort class gets all of the sort keys that were defined. This method has the following syntax:

```
public int[] getSortKeys()
```

JobProperty

JobProperty is a wrapper class that details the JIL name of the job property, the value, and its SQL type.

extends

java.lang.Object

implements

IJobProperty

JobProperty

The JobProperty constructor has the following syntax:

```
public JobProperty(java.lang.String name, java.lang.String value, int type)
```

getName

The getName method in the JobProperty class requests the name of the job property. This will be a valid JIL keyword. This method has the following syntax:

```
java.lang.String getName()
```

value

The value method in the JobProperty class requests the value of the job property. This method has the following syntax:

```
java.lang.String value()
```

getType

The getType method in the JobProperty class requests the SQL type for this property. This method has the following syntax:

```
int getType()
```

JobReq

extends

Cat1Request

implements

IJobReq

JobReq is an abstract class representing common functionality for job-related requests.

JobRunsFilterAnd

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobRunsFilterComposite, IAndComposite

The JobRunsFilterAnd class connects all contained filter objects (composite or leaf) with a logical 'and'.

JobRunsFilterAnd

The JobRunsFilterAnd constructor has the following syntax:

```
public JobRunsFilterAnd()
```

add

The add method in the JobRunsFilterAnd class has the following syntax:

```
public void add(IJobRunsFilter filter)
```

toString

The toString method in the JobRunsFilterAnd class overrides toString in java.lang.Object class. This method has the following syntax:

```
public java.lang.String toString()
```

getComponents

The getComponents method in the JobRunsFilterAnd class returns the list of cached leaf filter components. This method has the following syntax:

```
public java.util.ArrayList getComponents()
```

JobRunsFilterChar

extends

FilterChar

implements

IFilterLeaf, IJobRunsFilterChar

The JobRunsFilterChar class is a JobRunsFilter leaf that filters on character values.

JobRunsFilterChar

This JobRunsFilterChar constructor has the following syntax:

```
public JobRunsFilterChar(int id, char value, int operator)
```

id

Defines the ID of the character property. Only FLT_JOB_TYPE is supported.

value

Specifies the value to filter against.

operator

Specifies the comparison operator. Only EQUAL and NOT_EQUAL are accepted.

This JobRunsFilterChar constructor throws java.lang.IllegalArgumentException if an operator other than EQUAL or NOT_EQUAL is given.

JobRunsFilterChar

This JobRunsFilterChar constructor has the following syntax:

```
public JobRunsFilterChar(int id, char[] values, int operator)
```

id

Defines the ID of the character property. Only FLT_JOB_TYPE is supported.

values

Specifies the values to filter against.

operator

Defines the comparison operator. Only EQUAL and NOT_EQUAL are accepted.

This JobRunsFilterChar constructor throws java.lang.IllegalArgumentException if an operator other than EQUAL or NOT_EQUAL is given.

JobRunsFilterChar

This JobRunsFilterChar constructor is a convenience constructor that builds a filter based on the contents of the given filter. This constructor has the following syntax:

```
public JobRunsFilterChar(IJobRunsFilterChar filter)  
filter
```

Defines an object that implements the IJobRunsFilterChar interface.

toString

The `toString` method in the `JobRunsFilterChar` class gets the filter into string format. This method has the following syntax:

```
public java.lang.String toString()
```

JobRunsFilterDate

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobRunsFilterDate

The JobRunsFilterDate class is a JobRunsFilter leaf that filters on dates.

JobRunsFilterDate

This JobRunsFilterDate constructor creates a JobRunsFilterDate that matches the specified job field according to the specified operator. For example, to match jobs whose start time is more recent than April 4, 2004, create the object as follows:

```
Date fdate = new Date(104, 3, 4);
JobRunsFilterDate jrfd = new JobRunsFilterDate(JobRunsFilterDateFLT_START_TIME,
JobRunsFilterDate.GREATER);
```

This constructor has the following syntax:

```
public JobRunsFilterDate(int id, java.util.Date date, int operator)
```

id

Defines one of the following IDs of the date property:

- FLT_START_TIME
- FLT_END_TIME
- FLT_EVENT_TIME

date

Specifies the date to filter against.

operator

Specifies one of the following operators to apply to this date:

- EQUAL
- GREATER
- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

This JobRunsFilterDate constructor throws java.lang.IllegalArgumentException if an invalid operator is given.

JobRunsFilterDate

This JobRunsFilterDate constructor is a convenience constructor that builds a filter based on the contents of the given filter.

This constructor has the following syntax:

```
public JobRunsFilterDate(IJobRunsFilterDate filter)  
filter
```

Defines an object that implements the IJobRunsFilterDate interface

toString

The `toString` method in the JobRunsFilterDate class returns the string representation of the filter. This method has the following syntax:

```
public java.lang.String toString()
```

getFilterId

The `getFilterId` method in the JobRunsFilterDate class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getDate

The `getDate` method in the JobRunsFilterDate class returns the date. This method has the following syntax:

```
public java.util.Date getDate()
```

getOperator

The `getOperator` method in the JobRunsFilterDate class returns the operator. This method has the following syntax:

```
public int getOperator()
```

JobRunsFilterInt

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobRunsFilterInt

The JobRunsFilterInt is a JobRunsFilter leaf that filters on integers.

JobRunsFilterInt

This JobRunsFilterInt constructor creates a JobRunsFilterInt that will filter integer properties according to the given operator. This constructor has the following syntax:

`public JobRunsFilterInt(int id, int value, int operator)`

id

Defines one of the following IDs of the int property:

- FLT_EVENT
- FLT_JOID
- FLT_RUN_NUM
- FLT_NTRY
- FLT_STATUS
- FLT_EXIT_CODE
- FLT_SVC_DESK_STATUS
- FLT_BOX_JOID
- FLT_QUE_STATUS
- FLT_EVENT_STATUS

value

Specifies the integer to filter against.

operator

Specifies one of the following operators to apply to this integer:

- EQUAL
- GREATER

- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

This JobRunsFilterInt constructor throws java.lang.IllegalArgumentException if an invalid operator is given.

JobRunsFilterInt

This JobRunsFilterInt constructor creates a JobRunsFilterInt that will match integer properties for any integer in the given integer array.

This constructor has the following syntax:

```
public JobRunsFilterInt(int id, int[] values, int operator)
```

id

Defines one of the following IDs of the int property:

- FLT_EVENT
- FLT_JOID
- FLT_RUN_NUM
- FLT_NTRY
- FLT_STATUS
- FLT_EXIT_CODE
- FLT_SVC_DESK_STATUS
- FLT_BOX_JOID
- FLT_QUE_STATUS
- FLT_EVENT_STATUS

values

Specifies an integer array of values to match.

operator

Specifies one of the following operators to apply to this integer:

- EQUAL
- NOT_EQUAL

This JobRunsFilterInt constructor throws java.lang.IllegalArgumentException if an operator other than EQUAL or NOT_EQUAL is given.

JobRunsFilterInt

This JobRunsFilterInt constructor is a convenience constructor that builds a filter based on the contents of the given filter.

This constructor has the following syntax:

```
public JobRunsFilterInt(IJobRunsFilterInt filter)  
filter
```

Specifies an object that implements the IJobRunsFilterInt interface.

toString

The `toString` method in the JobRunsFilterInt class returns the string representation of the filter. This method has the following syntax:

```
public java.lang.String toString()
```

This method overrides `toString` in the class `java.lang.Object`.

getFilterId

The `getFilterId` method in the JobRunsFilterInt class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getOperator

The `getOperator` method in the JobRunsFilterInt class returns the operator. This method has the following syntax:

```
public int getOperator()
```

getValue

The `getValue` method in the JobRunsFilterInt class returns the value. This method has the following syntax:

```
public int getValue()
```

getValues

The `getValues` method in the JobRunsFilterInt class returns the set of values. This method has the following syntax:

```
public int[] getValues()
```

JobRunsFilterOr

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobRunsFilterComposite, IOrComposite

The JobRunsFilterOr class connects all contained filter objects (composite or leaf) with a logical 'or'.

JobRunsFilterOr

The JobRunsFilterOr constructor creates a JobRunsFilterOr object. This constructor has the following syntax:

```
public JobRunsFilterOr()
```

add

The add method in the JobRunsFilterOr class has the following syntax:

```
public void add(IJobRunsFilter filter)
```

toString

The toString method in the JobRunsFilterOr class overrides toString in java.lang.Object class. This method has the following syntax:

```
public java.lang.String toString()
```

getComponents

The getComponents method in the JobRunsFilterOr class returns the list of cached leaf filter components. This method has the following syntax:

```
public java.util.ArrayList getComponents()
```

JobRunsFilterString

extends

com.ca.autosys.services.request.filter.Filter

implements

IJobRunsFilterString

The JobRunsFilterString is a JobRunsFilter leaf that filters on Strings.

JobRunsFilterString

This JobRunsFilterString constructor creates a JobRunsFilterString that will match jobs that have the given String value using the defined operator. This constructor has the following syntax:

```
public JobRunsFilterString(int id, java.lang.String value, int operator)
```

id

Defines one of the following IDs of the String property:

- FLT_EVENT_TEXT
- FLT_JOB_NAME
- FLT_GROUP
- FLT_APPLICATION
- FLT_TIME_ZONE
- FLT_TARGET_MACHINE
- FLT_RUN_MACHINE
- FLT_SVC_DESK_HANDLE
- FLT_OVERRIDE_NUM
- FLT_BOX_NAME
- FLT_OWNER

value

Specifies the String value to match on. This parameter can contain wild cards.

operator

Specifies one of the following operators to apply to this String:

- LIKE
- NOT_LIKE

This JobRunsFilterString constructor throws java.lang.IllegalArgumentException if an invalid operator is given.

JobRunsFilterString

This JobRunsFilterString convenience constructor creates a JobRunsFilterString that will match jobs that have the given String value.

This constructor has the following syntax:

```
public JobRunsFilterString(int id, java.lang.String value)
```

id

Defines one of the following IDs of the String property:

- FLT_EVENT_TEXT
- FLT_JOB_NAME
- FLT_GROUP
- FLT_APPLICATION
- FLT_TIME_ZONE
- FLT_TARGET_MACHINE
- FLT_RUN_MACHINE
- FLT_SVC_DESK_HANDLE
- FLT_OVERRIDE_NUM
- FLT_BOX_NAME
- FLT_OWNER

value

Specifies the String value to match on. This parameter can contain wild cards.

JobRunsFilterString

This JobRunsFilterString constructor is a convenience constructor that builds a filter based on the contents of the given filter.

This constructor has the following syntax:

```
public JobRunsFilterString(IJobRunsFilterString filter)
```

filter

Specifies an object that implements the IJobRunsFilterString interface.

toString

The `toString` method in the `JobRunsFilterString` class returns the string representation of the filter. This method has the following syntax:

```
public java.lang.String toString()
```

This method overrides `toString` in the class `java.lang.Object`.

getFilterId

The `getFilterId` method in the `JobRunsFilterString` class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getOperator

The `getOperator` method in the `JobRunsFilterString` class returns the operator. This method has the following syntax:

```
public int getOperator()
```

getValue

The `getValue` method in the `JobRunsFilterString` class returns the value. This method has the following syntax:

```
public java.lang.String getValue()
```

ModifyCalendarDate

ModifyCalendarDate modifies an existing calendar by appending dates.

Example: Sample Code for Executing the ModifyCalendarDate API

```
void Sample(String serverName, int serverPort, String calendarName) {  
    // create the api  
    AsApi api = new AsApi(serverName, serverPort);  
  
    // create the request  
    ModifyCalendarDateReq req = new ModifyCalendarDateReq();  
    req.setRequest(calendarName, new Date[] {new Date()});  
  
    // run it, and check output  
    try {  
        req.execute(api);  
        System.out.println("\nModification Success: Current date and time added to  
" + calendarName + ".");  
    } catch (AsSecurityException secEx) {  
        System.out.println("\nSecurity Exception: " + secEx.getMessage().trim());  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("\nGeneral Exception: " + genEx.getMessage().trim());  
    } catch (AsException e) {  
        System.out.println("\nAutoSys Exception: " + e.getMessage().trim());  
    }  
}
```

ModifyCalendarDateReq

extends

Cat1Request

implements

ICalendarDateReq

ModifyCalendarDateReq modifies an existing calendar by appending dates.

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The user does not have authority to modify the calendar. No modification will be performed.

Example: Sample Code for Executing the ModifyCalendarDateReq API

```
void Sample(String serverName, int serverPort, String calendarName) {  
    // create the api  
    AsApi api = new AsApi(serverName, serverPort);  
  
    // create the request  
    ModifyCalendarDateReq req = new ModifyCalendarDateReq();  
    req.setRequest(calendarName, new Date[] {new Date()});  
  
    // run it, and check output  
    try {  
        req.execute(api);  
        System.out.println("\nModification Success: Current date and time added to  
" + calendarName + ".");  
    } catch (AsSecurityException secEx) {  
        System.out.println("\nSecurity Exception: " + secEx.getMessage().trim());  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("\nGeneral Exception: " + genEx.getMessage().trim());  
    } catch (AsException e) {  
        System.out.println("\nAutoSys Exception: " + e.getMessage().trim());  
    }  
}
```

ModifyCalendarDateReq

The ModifyCalendarDateReq constructor creates a ModifyCalendarDateReq object. This constructor has the following syntax:

```
public ModifyCalendarDateReq()
```

setRequest

The `setRequest` method in the `ModifyCalendarDateReq` class creates the request to modify calendar dates. This method has the following syntax:

```
void setRequest(java.lang.String calendarName, java.util.Date[] calendarDates)
```

calendarName

Defines the name of the calendar to modify.

calendarDates

Defines the dates to append to the calendar.

ModifyCalendarDateRsp

extends

`ApiResponse`

implements

`IModifyCalendarDateRsp`

The `ModifyCalendarDateRsp` class comprises the response for the modify calendar date request. This response contains no special information.

ModifyCalendarDateRsp

This constructor has the following syntax:

```
public ModifyCalendarDateRsp(com.ca.autosys.services.JResponse rsp)
```

rsp

Defines the raw response object.

A `java.lang.IllegalArgumentException` is thrown if a null response is given.

ModifyCycleCalendarDays

ModifyCycleCalendarDays modifies cycle calendar days in the CA Workload Automation AE database.

Example: Sample Code for Executing the ModifyCycleCalendarDays API

```
private void sample(AsApi api, String calName) {
    ModifyCycleCalendarDaysReq req = new ModifyCycleCalendarDaysReq(calName);
    req.appendCycle(Tools.getDateFromString("08/08/2006 08:00"),
    Tools.getDateFromString("08/31/2006 17:00"));
    req.appendCycle(Tools.getDateFromString("12/01/2006 08:00"),
    Tools.getDateFromString("12/31/2006 17:00"));

    try {
        req.execute(api);
        System.out.println("Cycle calendar successfully modified.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("General Error Encountered: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("Security Error Encountered: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AutoSys Error Encountered: " + asEx.getMessage());
    }
}
```

ModifyCycleCalendarDaysReq

extends

Cat1Request

implements

IModifyCycleCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have authority to modify the cycle calendar.

ModifyCycleCalendarDaysReq

This ModifyCycleCalendarDaysReq constructor creates a ModifyCycleCalendarDaysReq object. This constructor has the following syntax:

```
public ModifyCycleCalendarDaysReq(java.lang.String calendarName)  
calendarName
```

Specifies the calendar name to be modified.

appendCycle

The appendCycle method in the ModifyCycleCalendarDaysReq class appends date ranges to the cycle calendar. This method has the following syntax:

```
public void appendCycle(java.util.Date startDate, java.util.Date endDate)
```

startDate

Specifies the start date for the cycle.

endDate

Specifies the end date for the cycle.

ModifyCycleCalendarDaysRsp

extends

ApiResponse

implements

IModifyCycleCalendarDaysRsp

The ModifyCycleCalendarDaysRsp class comprises the response for the modify cycle calendar days request. This response contains no special information.

ModifyCycleCalendarDaysRsp

This ModifyCycleCalendarDaysRsp constructor creates the ModifyCycleCalendarDays response. This constructor has the following syntax:

```
public ModifyCycleCalendarDaysRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

ModifyExtendedCalendarDays

ModifyExtendedCalendarDays modifies an extended calendar in the CA Workload Automation AE database.

Example: Sample Code for Executing the ModifyExtendedCalendarDays API

```
private void sample(AsApi api, String server, int port, String calendar) {  
    int[] workdays = {  
        AsConstants.DAY_MONDAY,  
        AsConstants.DAY_WEDNESDAY,  
        AsConstants.DAY_FRIDAY  
    };  
    String holidayCalendar = "HolidaySample";  
    String cycleCalendar = "CycleSample";  
    int holidayAction = AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY;  
    int weekendAction = AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY;  
    int conditionAdjustment = -5;  
    String conditionKeyword = "FOMWEEK";  
  
    ModifyExtendedCalendarDaysReq req = new ModifyExtendedCalendarDaysReq();  
    req.setRequest(calendar, workdays, holidayCalendar, cycleCalendar,  
    holidayAction, weekendAction, conditionAdjustment, conditionKeyword);  
  
    try {  
        req.execute(api);  
        System.out.println("Calendar " + calendar + " has been modified.");  
    } catch (AsSecurityException secEx) {  
        System.out.println("\nSecurity Violation: " + secEx.getMessage().trim());  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("\nGeneral Error: " + genEx.getMessage().trim());  
    } catch (AsException e) {  
        System.out.println("\nAutoSys Error: " + e.getMessage().trim());  
    }  
}
```

ModifyExtendedCalendarDaysReq

extends

Cat1Request

implements

IModifyExtendedCalendarDaysReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have authority to modify the extended calendar.

ModifyExtendedCalendarDaysReq

The ModifyExtendedCalendarDaysReq constructor creates a ModifyExtendedCalendarDaysReq that modifies an extended calendar in the CA Workload Automation AE database. This constructor has the following syntax:

```
public ModifyExtendedCalendarDaysReq()
```

setRequest

The setRequest method in the ModifyExtendedCalendarDaysReq class sets up the request to modify an extended calendar in the CA Workload Automation AE database. This method has the following syntax:

```
public void setRequest(java.lang.String calendarName, int[] workDays,  
java.lang.String holidayCalendar, java.lang.String cycleCalendar, int  
holidayAction, int weekendAction, int conditionAdjustment, java.lang.String  
conditionKeyword)
```

calendarName

Specifies the name of the calendar to modify.

workDays

Specifies the requested workdays. Valid workdays are:

- AsConstants.DAY_SUNDAY—Set Sunday as a workday.
- AsConstants.DAY_MONDAY—Set Monday as a workday.
- AsConstants.DAY_TUESDAY—Set Tuesday as a workday.
- AsConstants.DAY_WEDNESDAY—Set Wednesday as a workday.
- AsConstants.DAY_THURSDAY—Set Thursday as a workday.
- AsConstants.DAY_FRIDAY—Set Friday as a workday.
- AsConstants.DAY_SATURDAY—Set Saturday as a workday.

holidayCalendar

Specifies the name of the holiday calendar to use.

cycleCalendar

Specifies the name of the cycle calendar to use.

holidayAction

Specifies the action to take when the work falls on a holiday. Valid holiday actions are:

- AsConstants.CALENDAR_ACTION_DEFAULT—Do not schedule the job.
- AsConstants.CALENDAR_ACTION_ONLY—Schedule only on a holiday.
- AsConstants.CALENDAR_ACTION_SCHEDULE—Schedule always.
- AsConstants.CALENDAR_ACTION_NEXT_DAY—Schedule on the next day (day could be a holiday)
- AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY—Schedule on the next workday.
- AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY—Schedule on the previous workday.

weekendAction

Specifies the action to take when the work falls on a non-workday. Valid weekend actions are:

- AsConstants.CALENDAR_ACTION_DEFAULT—Schedule always.
- AsConstants.CALENDAR_ACTION_ONLY—Schedule only on a non-workday.
- AsConstants.CALENDAR_ACTION_NEXT_DAY—Schedule on the next day (day could be a non-workday)
- AsConstants.CALENDAR_ACTION_NEXT_WORK_DAY—Schedule on the next workday.
- AsConstants.CALENDAR_ACTION_PREVIOUS_WORK_DAY—Schedule on the previous workday.

conditionAdjustment

Specifies the negative or positive number defining the days before or after the condition setting.

conditionKeyword

Specifies the date keyword to use as a scheduling condition.

Note: For more information about condition keywords, see the *Reference Guide*.

ModifyExtendedCalendarDaysRsp

extends

ApiResponse

implements

IModifyExtendedCalendarDaysRsp

The ModifyExtendedCalendarDaysRsp class comprises the response for the modify extended calendar days request. This response contains no special information.

ModifyExtendedCalendarDaysRsp

The ModifyExtendedCalendarDaysRsp constructor creates a ModifyExtendedCalendarDaysRsp object. This constructor has the following syntax:

```
public ModifyExtendedCalendarDaysRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

ModifyParameterItem

extends

JobItem

ModifyParameterItem is the container class for specifying modify parameter arguments in job definitions. This JobItem subclass directly maps to the modify_parameter jil keyword.

Example: Sample Code for Using the ModifyParameterItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_J2EE_JMS_ENTITY_BEAN);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_BEAN_NAME, "myBeanName");
        insertReq.set(AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,
        "myContext");
        insertReq.set(AsJobConstants.KEYWORD_PROVIDER_URL, "myURL");
        insertReq.set(AsJobConstants.KEYWORD_OPERATION_TYPE,
        AsJobConstants.VALUE__OPERATION_TYPE__CREATE);

        // Set up the modify_parameter part of the job definition and
        // attach it to the base job definition.
        ModifyParameterItem mpi = new ModifyParameterItem();
        mpi.set(AsJobConstants.KEYWORD MODIFY_PARAMETER, "parm1=val1");
        mpi.set(AsJobConstants.KEYWORD MODIFY_PARAMETER, "parm2=val2");
        mpi.set(AsJobConstants.KEYWORD MODIFY_PARAMETER, "parm3=val3");
        insertReq.add(mpi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_BEAN_NAME,  
        AsJobConstants.KEYWORD_INITIAL_CONTEXT_FACTORY,  
        AsJobConstants.KEYWORD_PROVIDER_URL,  
        AsJobConstants.KEYWORD_OPERATION_TYPE,  
        AsJobConstants.KEYWORD MODIFY_PARAMETER  
    };  
  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();  
  
        // Process job level properties.  
        System.out.println("----- Begin Response -----");  
        for(int i=0; i<rspProperties.length; ++i) {  
            String keyword =  
                jobProperties.getKeywordFromConstant(rspProperties[i]);  
            switch(getRsp.getAttributeType(rspProperties[i])) {  
                case AsConstants.ATTRIBUTE_TYPE_STRING:  
                    System.out.println(keyword + ": " +  
                        getRsp.getString(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_INT:  
                    System.out.println(keyword + ": " +  
                        getRsp.getInt(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                    System.out.println(keyword + ": " +  
                        getRsp.getBoolean(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_LONG:  
                    System.out.println(keyword + ": " +  
                        getRsp.getLong(rspProperties[i]));  
                    break;  
            }  
        }  
    }  
}
```

```

        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
            System.out.println("\n----- Begin " + keyword + " JobItem
-----");
            JobItem ji = getRsp.getJobItem(rspProperties[i]);
            int itemProperties[] = ji.getAttributes();

            // Process job item level properties.
            for(int j=0; j<itemProperties.length; ++j) {
                System.out.print(keyword + ": ");

                switch(jobProperties.getKeywordDataType(itemProperties[j])) {
                    case AsConstants.ATTRIBUTE_TYPE_STRING:
                        System.out.println(ji.getString(itemProperties[j]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_INT:
                        System.out.println(ji.getInt(itemProperties[j]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                        System.out.println(ji.getBool(itemProperties[j]));
                        break;
                }
            }
            System.out.println("----- End " + keyword + " JobItem
-----\n");
            break;
        }
    }
    System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}

```

Sample output

```

----- Begin Response -----
job_name: modify_parameter_sample
alarm_if_fail: true
machine: localhost
bean_name: myBeanName
initial_context_factory: myContext
provider_url: myURL
operation_type: 1

```

```
----- Begin modify_parameter JobItem -----
modify_parameter: parm1=val1
modify_parameter: parm2=val2
modify_parameter: parm3=val3
----- End modify_parameter JobItem -----

----- End Response -----
```

[ModifyParameterItem](#)

The `ModifyParameterItem` constructor creates a `ModifyParameterItem` object. This constructor has the following syntax:

```
public ModifyParameterItem()
```

Methods Inherited

This class inherits the following methods from the `JobItem` class:

- `add`
- `getAttributes`
- `getBool`
- `getInt`
- `getItemType`
- `getJobItem`
- `getLoggableItem`
- `getString`
- `hasSubItems`
- `set(int keyword, boolean value)`
- `set(int keyword, int value)`
- `set(int keyword, java.lang.String value)`
- `toString`

OracleProgramDataItem

extends

JobItem

OracleProgramDataItem is the container class for specifying Oracle program data arguments in job definitions. This JobItem subclass directly maps to the oracle_programdata jil keyword.

Example: Sample Code for Using the OracleProgramDataItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_ORACLE_REQUEST_SET);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_ORACLE_APPL_NAME, "myApplication");
        insertReq.set(AsJobConstants.KEYWORD_ORACLE_REQ_SET, "mySet");

        // Set up the oracle_programdata part of the job definition and
        // attach it to the base job definition.
        OracleProgramDataItem opdi = new OracleProgramDataItem();
        opdi.set(AsJobConstants.KEYWORD_ARGS, "myArgs");
        opdi.set(AsJobConstants.KEYWORD_INDEX, 5);
        opdi.set(AsJobConstants.KEYWORD_PRINT_COPIES, 2);
        opdi.set(AsJobConstants.KEYWORD_PRINT_STYLE, "myPrintStyle");
        opdi.set(AsJobConstants.KEYWORD_PRINTER, "myPrinter");
        opdi.set(AsJobConstants.KEYWORD_SAVEOP, false);
        insertReq.add(opdi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
```

```
try {
    // Set up the job name mask.
    JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

    // Set up what attributes are to be returned.
    int attributes[] = {
        AsJobConstants.KEYWORD_JOB_NAME,
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,
        AsJobConstants.KEYWORD_MACHINE,
        AsJobConstants.KEYWORD_ORACLE_APPL_NAME,
        AsJobConstants.KEYWORD_ORACLE_REQ_SET,
        AsJobConstants.KEYWORD_ORACLE_PROGRAMDATA
    };

    // Ready the request and execute it.
    getReq.setRequest(filter, attributes);
    set = getReq.execute(appServer);

    // Process the returned responses.
    while(set.hasNext()) {
        getRsp = (GetJobsWithFilterRsp)set.next();
        int rspProperties[] = getRsp.getAttributes();

        // Process job level properties.
        System.out.println("----- Begin Response -----");
        for(int i=0; i<rspProperties.length; ++i) {
            String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
            switch(getRsp.getAttributeType(rspProperties[i])) {
                case AsConstants.ATTRIBUTE_TYPE_STRING:
                    System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_INT:
                    System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                    System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_LONG:
                    System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
                    System.out.println("\n----- Begin " + keyword + " JobItem
-----");
                    break;
            }
        }
    }
}
```

```
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {

System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:

System.out.println(ji.getString(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_INT:

System.out.println(ji.getInt(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:

System.out.println(ji.getBool(itemProperties[j]));
            break;
        }

System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
    System.out.println("----- End Response -----");
}

} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: oracle_programdata_sample
alarm_if_fail: true
machine: localhost
oracle_appl_name: myApplication
oracle_req_set: mySet

----- Begin oracle_programdata JobItem -----
index: 5
args: myArgs
saveop: false
```

```
printer: myPrinter  
print_style: myPrintStyle  
print_copies: 2  
----- End oracle_programdata JobItem -----  
  
----- End Response -----
```

[OracleProgramDataItem](#)

The OracleProgramDataItem constructor creates an OracleProgramDataItem object. This constructor has the following syntax:

```
public OracleProgramDataItem()
```

[Methods Inherited](#)

This class inherits the following methods from the JobItem class:

- [add](#)
- [getAttributes](#)
- [getBool](#)
- [getInt](#)
- [getItemType](#)
- [getJobItem](#)
- [getLoggableItem](#)
- [getString](#)
- [hasSubItems](#)
- [set\(int keyword, boolean value\)](#)
- [set\(int keyword, int value\)](#)
- [set\(int keyword, java.lang.String value\)](#)
- [toString](#)

OverrideJob

OverrideJob specifies attributes of a job to be used on the next run of the job. This API is used by JIL for the override_job: feature. Set the job name and any fields you want to override.

Example: Sample Code for Executing the OverrideJob API

```
private void sample(AsApi api) {  
    // Set up the override properties.  
    Properties overrideInfo = new Properties();  
    overrideInfo.put(IOverrideJobReq.JOB_NAME, "test_job");  
    overrideInfo.put(IOverrideJobReq.MACHINE, "test_machine");  
  
    // Set up the request.  
    IOverrideJobReq req = new OverrideJobReq();  
    req.setRequest(overrideInfo);  
  
    // Execute the request and check for errors.  
    try {  
        req.execute(api);  
        System.out.println("\nOverride successful.");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsExternalValidationException extValidEx) {  
        System.out.println("AsExternalValidationException: " +  
extValidEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

OverrideJobReq

extends

JobReq

implements

IOverrideJobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—You do not have permission to override the named job.
- AsFieldValidationException—Properties specified in the override failed validation.

OverrideJobReq

The OverrideJobReq constructor creates an OverrideJobReq request for override job. This constructor has the following syntax:

```
public OverrideJobReq()
```

setRequest

The setRequest method in the OverrideJobReq class sets the job name and any fields to override. This method has the following syntax:

```
void setRequest (java.util.Properties overrideInfo)
```

overrideInfo

Specifies a Properties object that has the following keys defined with String objects as values:

alarm_if_fail

Indicates whether an alarm should be posted to the Event processor if the job fails or is terminated (true or false).

application

Indicates the associated application for the job.

auto_delete

Indicates whether the job should be automatically deleted after completion (true or false).

auto_hold

Indicates whether to change the job state to ON_HOLD when the box it is in begins RUNNING (true or false).

avg_runtime

Indicates an average run time (in minutes) for a job that is newly submitted to the database.

box_failure

Specifies the conditions to be interpreted as a box failure. For example, failure(JobA) OR failure(JobB) where JobA and JobB are in the box.

box_name

Specifies the name of the box to which this job belongs.

box_success

Specifies the conditions to be interpreted as a box success. For example, success(JobA) OR success(JobB) where JobA and JobB are in the box.

box_terminator

Indicates whether the box containing this job should be terminated if the job fails or terminates (true or false).

chk_files

Specifies the minimum amount of file space that must be available on designated file systems for the job to be started. For example, /tmp 1024 specifies the /tmp file system and 1024 KB.

command

Specifies the name and arguments of any command, executable, UNIX script or batch file to associate with the job.

condition

Defines conditions for a job to run For example, success(DB_BACKUP), exitcode (my_job=4), VALUE(TODAY)=Friday.

date_conditions

Indicates whether or not there are date or time conditions for starting this job (true or false).

days_of_week

Specifies the days of the week on which to run the job. For example, mo,tu,we,fr.

description

Specifies a description for the job; for documentation purposes only.

exclude_calendar

Indicates the name of the custom calendar to be used for determining the days of the week on which to not run this job.

group

Indicates the job's associated group.

heartbeat_interval

Specifies the frequency (in minutes) at which this job's command is expected to issue a heartbeat.

job_load

Specifies the relative amount of processing power the job consumes.

machine

Defines the Client machine on which to run the job.

max_exit_success

Specifies the maximum exit code with which the job can exit and still be considered a success.

max_run_alarm

Specifies the maximum run time (in minutes) that a job should require to finish normally.

min_run_alarm

Specifies the minimum run time (in minutes) that a job should require to finish normally.

n_retrys

Specifies how many times, if any, the job should be restarted after exiting with a FAILURE status.

notification_id

Specifies the NSM Notification ID.

notification_msg

Specifies the NSM Notification message.

owner

Defines the owner of the job.

permission

Specifies the permission for this job.

priority

Specifies the queue priority of the job.

profile

Specifies a job profile that defines environment variables to be set before the specified command is executed.

run_calendar

Indicates the name of the custom calendar to be used when determining the days of the week on which to run a job.

run_window

Indicates the time span during which the job will be allowed to start. If this attribute is specified, then when the job is eligible to run.

send_notification

Sends NSM notification on job failure.

service_desk

Creates a Service Desk issue on job failure.

start_mins

Defines the number of minutes after the hour, every hour, to start the job on the specified days. For example, "20,40"; "20".

start_times

Indicates the times of day, in 24-hour format, on the specified days or dates, to start the job. For example, "12:00, 13:00, 14:15".

std_err_file

Specifies the file to which the standard error file output should be redirected.

std_in_file

Specifies the file to which the standard input file for the job should be redirected.

std_out_file

Specifies the file to which the standard output file should be redirected.

svcdesk_attr

Specifies the Service Desk attribute.

svcdesk_imp

Specifies the Service Desk impact level using one of the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

svcdesk_pri

Specifies the Service Desk priority using one of the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

svcdesk_sev

Specifies the Service Desk severity using one of the following values:

- 0: None
- 1: High
- 2: Medium-High
- 3: Medium
- 4: Medium-Low
- 5: Low

term_run_time

Specifies the maximum run time (in minutes) that a job should require to finish normally.

timezone

Allows you to schedule a job based on a chosen time zone.

watch_file

Specifies the full path name of the file for which this file watcher job should watch.

watch_file_min_size

Specifies the watch file minimum size (in bytes), which determines when enough data has been written to the file to consider it complete.

watch_file_interval

Specifies the interval (in seconds) at which the file watcher job checks for the existence and size of the watched-for file.

OverrideJobRsp

```
extends  
    ApiResponse  
implements  
    IOverrideJobRsp
```

This response contains no special information. An exception is thrown if an error is detected.

OverrideJobRsp

The OverrideJobRsp constructor creates an OverrideJobRsp response for the override job request. This constructor has the following syntax:

```
public OverrideJobRsp(com.ca.autosys.services.JResponse rsp)
```

getOverrideNumber

The getOverrideNumber method in the OverrideJobRsp class returns the override number assigned to the override just created. This method has the following syntax:

```
int getOverrideNumber()
```

PingApi

PingApi is a convenient way to see if communications can be established with the Application Server. Since all response objects include the version of the Application Server (ApiResponse.getServerVersion()), this is also a convenient way to determine the version of an Application Server up front before continuing communications with it.

PingApi pings the CA Workload Automation AE server.

Example: Sample Code for Executing the PingApi API

```
private void sample(AsApi appServer) {  
    PingApiReq req = new PingApiReq();  
    PingApiRsp rsp = null;  
  
    try {  
        rsp = (PingApiRsp)req.execute(appServer);  
        System.out.println("Server version: " + rsp.getServerVersion());  
    } catch (AsException e) {  
        System.out.println("AsException: " + e.getMessage());  
    }  
}
```

PingApiReq

extends
Cat1Request
implements
IPingApiReq

This class pings the CA Workload Automation AE server to retrieve the server version.

Example: Sample Code for Using the PingApiReq Object

```
private void sample(AsApi appServer) {  
    PingApiReq req = new PingApiReq();  
    PingApiRsp rsp = null;  
  
    try {  
        rsp = (PingApiRsp)req.execute(appServer);  
        System.out.println("Server version: " + rsp.getServerVersion());  
    } catch (AsException e) {  
        System.out.println("AsException: " + e.getMessage());  
    }  
}
```

PingApiReq

The PingApiReq constructor in the PingApiReq class configures the PingApi request. This constructor has the following syntax:

```
Public PingApiReq()
```

PingApiRsp

extends
ApiResponse
implements
IPingApiRsp

PingApiRsp is the response object for the PingApi API.

PingApiRsp

The PingApiRsp constructor in the PingApiRsp class constructs the PingApi response. This constructor has the following syntax:

```
PingApiRsp(com.ca.autosys.services.JResponse response)
```

PsRunCntrlArgsItem

extends

JobItem

PsRunCntrlArgsItem is the container class for specifying peoplesoft run control arguments in job definitions. This JobItem subclass directly maps to the ps_run_cntrl_args jil keyword.

Example: Sample Code for Using the PsRunCntrlArgsItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_PEOPLESOFT);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_PS_PROCESS_NAME, "myProcess");

        // Set up the ps_run_cntrl_args part of the job definition and
        // attach it to the base job definition.
        PsRunCntrlArgsItem prcai = new PsRunCntrlArgsItem();
        prcai.set(AsJobConstants.KEYWORD_PS_RUN_CNTRL_ARGS, "Arg1");
        prcai.set(AsJobConstants.KEYWORD_PS_RUN_CNTRL_ARGS, "Arg2");
        prcai.set(AsJobConstants.KEYWORD_PS_RUN_CNTRL_ARGS, "Arg with spaces");
        prcai.set(AsJobConstants.KEYWORD_PS_RUN_CNTRL_ARGS, "Arg4");
        insertReq.add(prcai);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
```

```
        JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

        // Set up what attributes are to be returned.
        int attributes[] = {
            AsJobConstants.KEYWORD_JOB_NAME,
            AsJobConstants.KEYWORD_ALARM_IF_FAIL,
            AsJobConstants.KEYWORD_MACHINE,
            AsJobConstants.KEYWORD_PS_PROCESS_NAME,
            AsJobConstants.KEYWORD_PS_RUN_CNTRL_ARGS
};

        // Ready the request and execute it.
getReq.setRequest(filter, attributes);
set = getReq.execute(appServer);

        // Process the returned responses.
while(set.hasNext()) {
    getRsp = (GetJobsWithFilterRsp)set.next();
    int rspProperties[] = getRsp.getAttributes();

        // Process job level properties.
System.out.println("----- Begin Response -----");
for(int i=0; i<rspProperties.length; ++i) {
    String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
    switch(getRsp.getAttributeType(rspProperties[i])) {
        case AsConstants.ATTRIBUTE_TYPE_STRING:
            System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_INT:
            System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
            System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_LONG:
            System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
            System.out.println("\n----- Begin " + keyword + " JobItem
-----");
            JobItem ji = getRsp.getJobItem(rspProperties[i]);
            int itemProperties[] = ji.getAttributes();
    }
}
}
```

```
// Process job item level properties.  
for(int j=0; j<itemProperties.length; ++j) {  
  
System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");  
  
switch(jobProperties.getKeywordDataType(itemProperties[j])) {  
    case AsConstants.ATTRIBUTE_TYPE_STRING:  
  
System.out.println(ji.getString(itemProperties[j]));  
        break;  
    case AsConstants.ATTRIBUTE_TYPE_INT:  
  
System.out.println(ji.getInt(itemProperties[j]));  
        break;  
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
  
System.out.println(ji.getBool(itemProperties[j]));  
        break;  
    }  
}  
System.out.println("----- End " + keyword + " JobItem  
-----\n");  
        break;  
}  
}  
System.out.println("----- End Response -----");  
}  
}  
} catch (AsInitializationException e) {  
    e.printStackTrace();  
} catch (AsException e) {  
    e.printStackTrace();  
}  
}  
}
```

Sample output

```
----- Begin Response -----
job_name: ps_run_cntrl_args_sample
alarm_if_fail: true
machine: localhost
ps_process_name: myProcess

----- Begin ps_run_cntrl_args JobItem -----
ps_run_cntrl_args: Arg1
ps_run_cntrl_args: Arg2
ps_run_cntrl_args: Arg with spaces
ps_run_cntrl_args: Arg4
----- End ps_run_cntrl_args JobItem -----

----- End Response -----
```

PsRunCntrlArgsItem

This constructor has the following syntax:

```
public PsRunCntrlArgsItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)

toString

RegenerateExtendedCalendar

RegenerateExtendedCalendar regenerates an extended calendar in the CA Workload Automation AE database.

Example: Sample Code for Executing the RegenerateExtendedCalendar API

```
private void sample( AsApi api, String calendarName ) {  
    RegenerateExtendedCalendarReq req = new RegenerateExtendedCalendarReq();  
    req.setRequest( calendarName );  
  
    try {  
        req.execute( api );  
        System.out.println( "Calendar " + calendarName + " successfully  
    regenerated." );  
    } catch ( AsGeneralErrorException genEx ) {  
        System.out.println( "General Exception Caught: " + genEx.getMessage() );  
    } catch ( AsSecurityException secEx ) {
```

```

        System.out.println( "Security Exception Caught: " + secEx.getMessage() );
    } catch ( AsObjectDoesNotExistException noObjEx ) {
        System.out.println( "AsObjectDoesNotExistException: " +
noObjEx.getMessage() );
    } catch ( AsException asEx ) {
        System.out.println( "AutoSys Exception Caught: " + asEx.getMessage() );
    } }
}

```

RegenerateExtendedCalendarReq

extends

Cat1Request

implements

IRegenerateExtendedCalendarReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have the authority to regenerate the extended calendar.
- AsObjectDoesNotExistException - The requested calendar does not exist.

setRequest

The setRequest method in the RegenerateExtendedCalendarReq class configures the request to regenerate the extended calendar. This method has the following syntax:

`void setRequest(java.lang.String calendarName)`

calendarName

Specifies the extended calendar to regenerate.

RegenerateExtendedCalendarRsp

The RegenerateExtendedCalendarRsp class contains the RegenerateExtendedCalendarRsp object.

extends

ApiResponse

implements

IRegenerateExtendedCalendarRsp

RemoveOverride

RemoveOverride deletes an override that was previously set for the specified jobs. This API is used by JIL to execute `override_job: job_name delete`. RemoveOverride does not support wildcards.

Given a job name, RemoveOverride verifies write access to the job. It then removes the override and sends or deletes future events if the date or time fields are affected.

Example: Sample Code for Executing the RemoveOverride API

```
private void sample(AsApi api, String jobName) {  
    // Set up the request.  
    IRemoveOverrideReq req = new RemoveOverrideReq();  
    req.setRequest(jobName);  
  
    // Execute the request and check for errors.  
    try {  
        req.execute(api);  
        System.out.println("\noverride successfully removed from \\" + jobName +  
"\\".");  
    } catch (AsGeneralErrorException genEx) {  
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());  
    } catch (AsSecurityException secEx) {  
        System.out.println("AsSecurityException: " + secEx.getMessage());  
    } catch (AsException asEx) {  
        System.out.println("AsException: " + asEx.getMessage());  
    }  
}
```

RemoveOverrideReq

extends

Cat1Request

implements

IRemoveOverrideReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - You do not have permission to remove the override on the named job.

setRequest

The setRequest method in the RemoveOverrideReq class defines the name of the job for which to delete an override. This method has the following syntax:

```
void setRequest(java.lang.String jobname)  
jobname
```

Defines the name of the job for which to remove the override.

RemoveOverrideRsp

extends

```
ApiResponse
```

The RemoveOverrideRsp class comprises the response for the remove override request. This response contains no special information. An exception occurs if an error is detected.

RemoveOverrideRsp

The RemoveOverrideRsp constructor creates a RemoveOverrideRsp object. This constructor has the following syntax:

```
public RemoveOverrideRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

ResourceFilterAnd

extends

com.ca.autosys.services.request.filter.Filter

implements

IResourceFilterComposite, IAndComposite

The ResourceFilterAnd class is a container that connects the added filters with a logical AND.

ResourceFilterAnd

The ResourceFilterAnd constructor creates a ResourceFilterAnd object. This constructor has the following syntax:

```
public ResourceFilterAnd()
```

add

The add method in the ResourceFilterAnd class logically connects the given filter to this filter with AND. This method has the following syntax:

```
public void add(IResourceFilter filter)
```

filter

Defines the filter to AND.

toString

The toString method in the ResourceFilterAnd class overrides toString in java.lang.Object class. This method has the following syntax:

```
public java.lang.String toString()
```

getComponents

The getComponents method in the ResourceFilterAnd class returns the list of cached leaf filter components. This method has the following syntax:

```
public java.util.ArrayList getComponents()
```

ResourceFilterInt

extends

com.ca.autosys.services.request.filter.Filter

implements

IResourceFilterInt

The ResourceFilterInt class is a ResourceFilter leaf that filters on integers.

ResourceFilterInt

This ResourceFilterInt constructor creates a ResourceFilterInt object that will filter integer properties according to the given operator. This constructor has the following syntax:

```
public ResourceFilterInt(int id, int value, int operator)
```

id

Defines the ID of the int property. Possible values are:

- FLT_VR_TYPE
- FLT_VR_AMT_DEF
- FLT_VR_AMT_IN_USE

value

Defines the integer to filter against.

operator

Defines the operator to apply to this integer. Possible values are:

- EQUAL
- GREATER
- LESS
- NOT_EQUAL
- GREATER_OR_EQUAL
- LESS_OR_EQUAL

A java.lang.IllegalArgumentException is thrown if an invalid operator is given.

ResourceFilterInt

This ResourceFilterInt constructor creates a ResourceFilterInt object that will match integer properties that will match for any integer in the given integer array. This constructor has the following syntax:

```
public ResourceFilterInt(int id, int[] values, int operator)
```

id

Defines the ID of the int property.

values

Defines an integer array of values to match.

operator

Defines the operator to apply to this integer. Possible values are:

- EQUAL
- NOT_EQUAL

A java.lang.IllegalArgumentException is thrown if an operator other than EQUAL or NOT_EQUAL is given.

ResourceFilterInt

This ResourceFilterInt is a convenience constructor that builds a filter based on the contents of the given filter. This constructor has the following syntax:

```
public ResourceFilterInt(IResourceFilterInt filter)
```

filter

Defines an object that implements the IResourceFilterInt interface.

toString

The `toString` method in the ResourceFilterInt class overrides `toString` in `java.lang.Object` class. This method has the following syntax:

```
public java.lang.String toString()
```

getFilterId

The `getFilterId` method in the ResourceFilterInt class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

getOperator

The `getOperator` method in the `ResourceFilterInt` class returns the operator. This method has the following syntax:

```
public int getOperator()
```

getValue

The `getValue` method in the `ResourceFilterInt` class returns the value. This method has the following syntax:

```
public int getValue()
```

getValues

The `getValues` method in the `ResourceFilterInt` class returns the set of values. This method has the following syntax:

```
public int[] getValues()
```

ResourceFilterOr

extends

com.ca.autosys.services.request.filter.Filter

implements

IResourceFilterComposite, IOrComposite

The ResourceFilterOr class is a container that connects the added filters with a logical OR.

ResourceFilterOr

The ResourceFilterOr constructor creates a ResourceFilterOr object. This constructor has the following syntax:

```
public ResourceFilterOr()
```

add

The add method in the ResourceFilterOr class logically connects the given filter to this filter with OR. This method has the following syntax:

```
public void add(IResourceFilter filter)
```

filter

Defines the filter to OR.

toString

The toString method in the ResourceFilterOr class overrides toString in java.lang.Object class. This method has the following syntax:

```
public java.lang.String toString()
```

getComponents

The getComponents method in the ResourceFilterOr class returns the list of cached leaf filter components. This method has the following syntax:

```
public java.util.ArrayList getComponents()
```

ResourceFilterString

extends

com.ca.autosys.services.request.filter.Filter

implements

IResourceFilterString

The ResourceFilterString class is a ResourceFilter leaf that filters on String.

ResourceFilterString

This ResourceFilterString constructor creates a ResourceFilterString that will match resources that have the given String value. This constructor has the following syntax:

`public ResourceFilterString(int id, java.lang.String value, int operator)`

id

Defines the ID of the String property. Possible values are:

- FLT_VR_NAME
- FLT_VR_MACHINE
- FLT_VR_DESC

value

Defines the String value to match on. This parameter can contain wild cards.

operator

Defines the operator to apply to this integer. Possible values are:

- LIKE
- NOT_LIKE

ResourceFilterString

This ResourceFilterString is a convenience constructor that creates a ResourceFilterString that will match resources that have the given String value. This constructor has the following syntax:

`public ResourceFilterString(int id, java.lang.String value)`

id

Defines The ID of the String property.

value

Defines String value to match on. This parameter can contain wild cards.

[ResourceFilterString](#)

This ResourceFilterString is a convenience constructor that builds a filter based on the contents of the given filter. This constructor has the following syntax:

```
public ResourceFilterString(IResourceFilterString filter)
```

filter

Defines an object that implements the IResourceFilterString interface.

[toString](#)

The `toString` method in the ResourceFilterString class overrides `toString` in `java.lang.Object` class. This method has the following syntax:

```
public java.lang.String toString()
```

[getFilterId](#)

The `getFilterId` method in the ResourceFilterString class returns the filter ID. This method has the following syntax:

```
public int getFilterId()
```

[getOperator](#)

The `getOperator` method in the ResourceFilterString class returns the operator. This method has the following syntax:

```
public int getOperator()
```

[getValue](#)

The `getValue` method in the ResourceFilterString class returns the value. This method has the following syntax:

```
public java.lang.String getValue()
```

SapArcParmItem

extends

JobItem

SapArcParmItem is the container class for specifying SAP archive parameters in job definitions. This JobItem subclass directly maps to the arc_parms jil keyword.

Example: Sample Code for Using the SapArcParmItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_SAP_DATA_ARCHIVING);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_ARC_OBJ_VARIANT, "myVariant");

        // Set up the arc_parms part of the job definition and
        // attach it to the base job definition.
        SapArcParmItem sapi = new SapArcParmItem();
        sapi.set(AsJobConstants.KEYWORD_ARC_FORMAT, "myFormat");
        sapi.set(AsJobConstants.KEYWORD_ARC_PRINTER, "PRTR");
        sapi.set(AsJobConstants.KEYWORD_ARC_CLIENT, 235);
        sapi.set(AsJobConstants.KEYWORD_ARC_CONNECT, "myConnect");
        sapi.set(AsJobConstants.KEYWORD_ARC_DOC_CLASS, "myDocClass");
        sapi.set(AsJobConstants.KEYWORD_ARC_DOC_TYPE, "myDocType");
        sapi.set(AsJobConstants.KEYWORD_ARC_HOST_LINK, "myHostLink");
        sapi.set(AsJobConstants.KEYWORD_ARC_INFO, "INF");
        sapi.set(AsJobConstants.KEYWORD_ARC_OBJ_TYPE, "myObjType");
        sapi.set(AsJobConstants.KEYWORD_ARC_PATH, "myPath");
        sapi.set(AsJobConstants.KEYWORD_ARC_PROTOCOL, "myProto");
        sapi.set(AsJobConstants.KEYWORD_ARC_REPORT, "myReport");
        sapi.set(AsJobConstants.KEYWORD_ARC_SERVICE, "myService");
        sapi.set(AsJobConstants.KEYWORD_ARC_STORAGE, "my");
        sapi.set(AsJobConstants.KEYWORD_ARC_TEXT, "myText");
        sapi.set(AsJobConstants.KEYWORD_ARC_VERSION, "Vers");
        sapi.set(AsJobConstants.KEYWORD_ARC_USERID, "myUserId");
        sapi.set(AsJobConstants.KEYWORD_ARC_DATE, "myDate");
        insertReq.add(sapi);
    }
```

```
        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

        // Set up what attributes are to be returned.
        int attributes[] = {
            AsJobConstants.KEYWORD_JOB_NAME,
            AsJobConstants.KEYWORD_ALARM_IF_FAIL,
            AsJobConstants.KEYWORD_MACHINE,
            AsJobConstants.KEYWORD_ARC_OBJ_VARIANT,
            AsJobConstants.KEYWORD_ARC_PARMS
        };

        // Ready the request and execute it.
        getReq.setRequest(filter, attributes);
        set = getReq.execute(appServer);

        // Process the returned responses.
        while(set.hasNext()) {
            getRsp = (GetJobsWithFilterRsp)set.next();
            int rspProperties[] = getRsp.getAttributes();

            // Process job level properties.
            System.out.println("----- Begin Response -----");
            for(int i=0; i<rspProperties.length; ++i) {
                String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
                switch(getRsp.getAttributeType(rspProperties[i])) {
                    case AsConstants.ATTRIBUTE_TYPE_STRING:
                        System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_INT:
                        System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
```

```
        System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_LONG:
        System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
        System.out.println("\n----- Begin " + keyword + " JobItem
-----");
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {

System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
    case AsConstants.ATTRIBUTE_TYPE_STRING:
        System.out.println(ji.getString(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_INT:
        System.out.println(ji.getInt(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
        System.out.println(ji.getBool(itemProperties[j]));
        break;
    }
}
System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
}
System.out.println("----- End Response -----");
}

} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: arc_parms_sample
alarm_if_fail: true
machine: localhost
arc_obj_variant: myVariant

----- Begin arc_parms JobItem -----
arc_client: 235
arc_connect: myConnect
arc_date: myDate
arc_doc_class: myDocClass
arc_doc_type: myDocType
arc_format: myFormat
arc_host_link: myHostLink
arc_info: INF
arc_obj_type: myObjType
arc_path: myPath
arc_printer: PRTR
arc_protocol: myProto
arc_report: myReport
arc_service: myService
arc_storage: my
arc_text: myText
arc_userid: myUserId
arc_version: Vers
----- End arc_parms JobItem -----

----- End Response -----
```

SapArcParmItem

This constructor has the following syntax:

```
public SapArcParmItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem

- `getLoggableItem`
- `getString`
- `hasSubItems`
- `set(int keyword, boolean value)`
- `set(int keyword, int value)`
- `set(int keyword, java.lang.String value)`
- `toString`

SapExtTableItem

extends

JobItem

SapExtTableItem is the container class for specifying SAP ext_table parameters in job definitions. This JobItem subclass directly maps to the sap_ext_table jil keyword.

Example: Sample Code for Using the SapExtTableItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_SAP_INFOPACKAGE);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_SAP_CLIENT, "235");

        // Set up the sap_ext_table part of the job definition and
        // attach it to the base job definition.
        SapExtTableItem seti = new SapExtTableItem();
        seti.set(AsJobConstants.KEYWORD_FIELD_NAME, "myField");
        seti.set(AsJobConstants.KEYWORD_HIGH, "high");
        seti.set(AsJobConstants.KEYWORD_LOW, "low");
        seti.set(AsJobConstants.KEYWORD_OBJECT_NAME, "myObjectName");
        seti.set(AsJobConstants.KEYWORD_OPERATION,
        AsJobConstants.VALUE__SAP_OPERATION__EQUAL);
        seti.set(AsJobConstants.KEYWORD_SIGN,
        AsJobConstants.VALUE__SAP_SIGN__INCLUDE);
        insertReq.add(seti);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

```
// Try and retrieve the job.  
try {  
    // Set up the job name mask.  
    JobFilterString filter = new  
    JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);  
  
    // Set up what attributes are to be returned.  
    int attributes[] = {  
        AsJobConstants.KEYWORD_JOB_NAME,  
        AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
        AsJobConstants.KEYWORD_MACHINE,  
        AsJobConstants.KEYWORD_JOB_CLASS,  
        AsJobConstants.KEYWORD_SAP_CLIENT,  
        AsJobConstants.KEYWORD_SAP_EXT_TABLE  
    };  
  
    // Ready the request and execute it.  
    getReq.setRequest(filter, attributes);  
    set = getReq.execute(appServer);  
  
    // Process the returned responses.  
    while(set.hasNext()) {  
        getRsp = (GetJobsWithFilterRsp)set.next();  
        int rspProperties[] = getRsp.getAttributes();  
  
        // Process job level properties.  
        System.out.println("----- Begin Response -----");  
        for(int i=0; i<rspProperties.length; ++i) {  
            String keyword =  
                jobProperties.getKeywordFromConstant(rspProperties[i]);  
            switch(getRsp.getAttributeType(rspProperties[i])) {  
                case AsConstants.ATTRIBUTE_TYPE_STRING:  
                    System.out.println(keyword + ": " +  
                        getRsp.getString(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_INT:  
                    System.out.println(keyword + ": " +  
                        getRsp.getInt(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                    System.out.println(keyword + ": " +  
                        getRsp.getBoolean(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_LONG:  
                    System.out.println(keyword + ": " +  
                        getRsp.getLong(rspProperties[i]));  
                    break;  
                case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
```

```
System.out.println("\n----- Begin " + keyword + " JobItem\n-----");
JobItem ji = getRsp.getJobItem(rspProperties[i]);
int itemProperties[] = ji.getAttributes();

// Process job item level properties.
for(int j=0; j<itemProperties.length; ++j) {

    System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

    switch(jobProperties.getKeywordDataType(itemProperties[j])) {
        case AsConstants.ATTRIBUTE_TYPE_STRING:
            System.out.println(ji.getString(itemProperties[j]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_INT:
            System.out.println(ji.getInt(itemProperties[j]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
            System.out.println(ji.getBool(itemProperties[j]));
            break;
    }
}

System.out.println("----- End " + keyword + " JobItem\n-----\n");
break;
}
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: sap_ext_table_sample
alarm_if_fail: true
machine: localhost
job_class: myClass
sap_client: 235

----- Begin sap_ext_table JobItem -----
field_name: myField
```

```
high: high
low: low
object_name: myObjectName
sign: 1
operation: 2
----- End sap_ext_table JobItem -----

----- End Response -----
```

SapExtTableItem

This constructor has the following syntax:

```
public SapExtTableItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

SapMailListItem

extends

JobItem

SapMailListItem is the container class for specifying SAP mail list parameters in job definitions. This JobItem subclass directly maps to the sap_maillist jil keyword.

Example: Sample Code for Using the SapMailListItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_SAP);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_SAP_CLIENT, "235");
        insertReq.set(AsJobConstants.KEYWORD_SAP_EVENT_ID, "myEventId");
        insertReq.set(AsJobConstants.KEYWORD_SAP_JOB_NAME, "mySapJob");

        // Set up the sap_maillist part of the job definition and
        // attach it to the base job definition.
        SapMailListItem smli = new SapMailListItem();
        smli.set(AsJobConstants.KEYWORD_SAP_MAILLIST, "myAddress@ca.com");
        smli.set(AsJobConstants.KEYWORD_SAP_MAILLIST, "yourAddress@ca.com");
        insertReq.add(smli);

        // Set up the sap_step_parms part of the job definition and attach
        // it to the base job definition.
        SapStepParmItem sspi = new SapStepParmItem();
        sspi.set(AsJobConstants.KEYWORD_ABAP_LANG, "EN");
        sspi.set(AsJobConstants.KEYWORD_ABAP_NAME, "RSLG0000");
        insertReq.add(sspi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
```

```
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

        // Set up what attributes are to be returned.
        int attributes[] = {
            AsJobConstants.KEYWORD_JOB_NAME,
            AsJobConstants.KEYWORD_ALARM_IF_FAIL,
            AsJobConstants.KEYWORD_MACHINE,
            AsJobConstants.KEYWORD_JOB_CLASS,
            AsJobConstants.KEYWORD_SAP_CLIENT,
            AsJobConstants.KEYWORD_SAP_EVENT_ID,
            AsJobConstants.KEYWORD_SAP_JOB_NAME,
            AsJobConstants.KEYWORD_SAP_STEP_PARMS,
            AsJobConstants.KEYWORD_SAP_MAILLIST
        };

        // Ready the request and execute it.
        getReq.setRequest(filter, attributes);
        set = getReq.execute(appServer);

        // Process the returned responses.
        while(set.hasNext()) {
            getRsp = (GetJobsWithFilterRsp)set.next();
            int rspProperties[] = getRsp.getAttributes();

            // Process job level properties.
            System.out.println("---- Begin Response ----");
            for(int i=0; i<rspProperties.length; ++i) {
                String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
                switch(getRsp.getAttributeType(rspProperties[i])) {
                    case AsConstants.ATTRIBUTE_TYPE_STRING:
                        System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_INT:
                        System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                        System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
                        break;
                }
            }
        }
    }
}
```

```
        case AsConstants.ATTRIBUTE_TYPE_LONG:
            System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
            System.out.println("\n----- Begin " + keyword + " JobItem
-----");
            JobItem ji = getRsp.getJobItem(rspProperties[i]);
            int itemProperties[] = ji.getAttributes();

            // Process job item level properties.
            for(int j=0; j<itemProperties.length; ++j) {

                System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

                switch(jobProperties.getKeywordDataType(itemProperties[j])) {
                    case AsConstants.ATTRIBUTE_TYPE_STRING:
                        System.out.println(ji.getString(itemProperties[j]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_INT:
                        System.out.println(ji.getInt(itemProperties[j]));
                        break;
                    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                        System.out.println(ji.getBool(itemProperties[j]));
                        break;
                }
            }
            System.out.println("----- End " + keyword + " JobItem
-----\n");
            break;
        }
    }
    System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: sap_maillist_sample
alarm_if_fail: true
machine: localhost
```

```
job_class: myClass
sap_client: 235
sap_event_id: myEventId
sap_job_name: mySapJob

----- Begin sap_step_parms JobItem -----
sap_fail_msg:
sap_success_msg:
abap_name: RSLG0000
prt_arc_mode: -2147483647
sap_banner: -2147483647
banner_page: false
expiration: -2147483647
num_columns: -2147483647
copies: -2147483647
dept:
dest:
dataset:
footer: false
format:
host_page: -2147483647
print_imm: false
num_lines: -2147483647
new_spool: false
print_priority: -2147483647
release: false
req_type:
show_passwd:
title:
spool_name:
abap_lang: EN
step_user:
variant:
arc_client: -2147483647
arc_connect:
arc_date:
arc_doc_class:
arc_doc_type:
arc_format:
arc_host_link:
arc_info:
arc_obj_type:
arc_path:
arc_printer:
arc_protocol:
arc_report:
arc_service:
arc_storage:
arc_text:
```

```
arc_userid:  
arc_version:  
web_posting: false  
recipient_name:  
----- End sap_step_parms JobItem -----  
----- Begin sap_maillist JobItem -----  
sap_maillist: myAddress@ca.com  
sap_maillist: yourAddress@ca.com  
----- End sap_maillist JobItem -----  
----- End Response -----
```

Note: If numeric fields are null in the database, a constant value is returned to the caller. This value is mapped to NULL_INT_VALUE or NULL_LONG_VALUE as defined in the AsJobConstants class.

SapMailListItem

This constructor has the following syntax:

```
public SapMailListItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

SapPrintParmItem

extends

JobItem

SapPrintParmItem is the container class for specifying SAP print parameters in job definitions. This JobItem subclass directly maps to the sap_print_parms jil keyword.

Example: Sample Code for Using the SapPrintParmItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_SAP_DATA_ARCHIVING);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_SAP_CLIENT, "235");
        insertReq.set(AsJobConstants.KEYWORD_ARC_OBJ_VARIANT, "myVariant");

        // Set up the sap_print_parms part of the job definition and
        // attach it to the base job definition.
        SapPrintParmItem sppi = new SapPrintParmItem();
        sppi.set(AsJobConstants.KEYWORD_BANNER_PAGE, true);
        sppi.set(AsJobConstants.KEYWORD_COPIES, 2);
        sppi.set(AsJobConstants.KEYWORD_DATASET, "myData");
        sppi.set(AsJobConstants.KEYWORD_DEPT, "myDepartment");
        sppi.set(AsJobConstants.KEYWORD_DEST, "LP01");
        sppi.set(AsJobConstants.KEYWORD_EXPIRATION, 9);
        sppi.set(AsJobConstants.KEYWORD_FOOTER, false);
        sppi.set(AsJobConstants.KEYWORD_FORMAT, "X_65_132");
        sppi.set(AsJobConstants.KEYWORD_HOST_PAGE,
        AsJobConstants.VALUE__COMMON__YES);
        sppi.set(AsJobConstants.KEYWORD_NEW_SPOOL, true);
        sppi.set(AsJobConstants.KEYWORD_NUM_COLUMNS, 132);
        sppi.set(AsJobConstants.KEYWORD_NUM_LINES, 65);
        sppi.set(AsJobConstants.KEYWORD_PRT_ARC_MODE,
        AsJobConstants.VALUE__SAP_ARC_MODE__BOTH);
        sppi.set(AsJobConstants.KEYWORD_PRINT_IMM, true);
        sppi.set(AsJobConstants.KEYWORD_PRINT_PRIORITY, 1);
        sppi.set(AsJobConstants.KEYWORD_RECIPIENT_NAME, "myRecipient");
    }
}
```

```
sppi.set(AsJobConstants.KEYWORD_RELEASE, true);
sppi.set(AsJobConstants.KEYWORD_REQ_TYPE, "myRequestType");
sppi.set(AsJobConstants.KEYWORD_SAP_BANNER,
AsJobConstants.VALUE__COMMON__NO);
sppi.set(AsJobConstants.KEYWORD_SPOOL_NAME, "mySpoolName");
sppi.set(AsJobConstants.KEYWORD_TITLE, "myTitle");
insertReq.add(sppi);

// Ready the request and execute it.
insertReq.setRequest(false);
insertReq.execute(appServer);
} catch (AsFieldValidationException e) {
e.printStackTrace();
} catch (AsException e) {
e.printStackTrace();
}

// Try and retrieve the job.
try {
// Set up the job name mask.
JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

// Set up what attributes are to be returned.
int attributes[] = {
AsJobConstants.KEYWORD_JOB_NAME,
AsJobConstants.KEYWORD_ALARM_IF_FAIL,
AsJobConstants.KEYWORD_MACHINE,
AsJobConstants.KEYWORD_JOB_CLASS,
AsJobConstants.KEYWORD_SAP_CLIENT,
AsJobConstants.KEYWORD_SAP_PRINT_PARMS
};

// Ready the request and execute it.
getReq.setRequest(filter, attributes);
set = getReq.execute(appServer);

// Process the returned responses.
while(set.hasNext()) {
getRsp = (GetJobsWithFilterRsp)set.next();
int rspProperties[] = getRsp.getAttributes();

// Process job level properties.
System.out.println("---- Begin Response ----");
for(int i=0; i<rspProperties.length; ++i) {
String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
switch(getRsp.getAttributeType(rspProperties[i])) {
case AsConstants.ATTRIBUTE_TYPE_STRING:
```

```

        System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_INT:
        System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
        System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_LONG:
        System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
        System.out.println("\n----- Begin " + keyword + " JobItem
-----");
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {

System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:

System.out.println(ji.getString(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_INT:

System.out.println(ji.getInt(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:

System.out.println(ji.getBool(itemProperties[j]));
            break;
        }
        System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {

```

```
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

Sample output

```
----- Begin Response -----
job_name: sap_print_parms_sample
alarm_if_fail: true
machine: localhost
job_class: myClass
sap_client: 235

----- Begin sap_print_parms JobItem -----
prt_arc_mode: 3
sap_banner: -2147483647
banner_page: true
expiration: 9
num_columns: 132
copies: 2
dept: myDepartment
dest: LP01
dataset: myData
footer: false
format: X_65_132
host_page: 1
print_imm: true
num_lines: 65
new_spool: true
print_priority: 1
release: true
req_type: myRequestType
title: myTitle
spool_name: mySpoolName
recipient_name: myRecipient
----- End sap_print_parms JobItem -----
----- End Response -----
```

Note: If numeric fields are null in the database, a constant value is returned to the caller. This value is mapped to NULL_INT_VALUE or NULL_LONG_VALUE as defined in the AsJobConstants class.

SapPrintParmItem

This constructor has the following syntax:

```
public SapPrintParmItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

SapRecipientItem

extends

JobItem

SapRecipientItem is the container class for specifying SAP recipient parameters in job definitions. This JobItem subclass directly maps to the sap_recipients jil keyword.

Example: Sample Code for Using the SapRecipientItem Object

```
private static void sampleUsage(AsApi appServer) {  
    AsJobProperties jobProperties = new AsJobProperties();  
    InsertJobReq insertReq = new InsertJobReq();  
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();  
    GetJobsWithFilterRsp getRsp;  
    IApiResponseSet set;  
  
    // Try and insert the job.  
    try {  
        // Set up the base job definition.  
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);  
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,  
AsJobConstants.JOB_TYPE_SAP);  
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");  
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");  
        insertReq.set(AsJobConstants.KEYWORD_SAP_CLIENT, "235");  
        insertReq.set(AsJobConstants.KEYWORD_SAP_EVENT_ID, "myEventId");  
        insertReq.set(AsJobConstants.KEYWORD_SAP_JOB_NAME, "mySapJob");  
  
        // Set up the sap_recipient part of the job definition and  
        // attach it to the base job definition.  
        SapRecipientItem sri = new SapRecipientItem();  
        sri.set(AsJobConstants.KEYWORD_RCP_TYPE,  
AsJobConstants.VALUE__RCP_TYPE__INTERNETMAIL);  
        sri.set(AsJobConstants.KEYWORD_RECIPIENT, "myAddress@ca.com");  
        insertReq.add(sri);  
  
        // Set up the sap_step_parms of the job definition and  
        // attach it to the base job definition.  
        SapStepParmItem sspi = new SapStepParmItem();  
        sspi.set(AsJobConstants.KEYWORD_ABAP_LANG, "EN");  
        sspi.set(AsJobConstants.KEYWORD_ABAP_NAME, "RSLG0000");  
        insertReq.add(sspi);  
  
        // Ready the request and execute it.  
        insertReq.setRequest(false);  
        insertReq.execute(appServer);  
    } catch (AsFieldValidationException e) {  
        e.printStackTrace();  
    }  
}
```

```
        } catch (AsException e) {
            e.printStackTrace();
        }

        // Try and retrieve the job.
        try {
            // Set up the job name mask.
            JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

            // Set up what attributes are to be returned.
            int attributes[] = {
                AsJobConstants.KEYWORD_JOB_NAME,
                AsJobConstants.KEYWORD_ALARM_IF_FAIL,
                AsJobConstants.KEYWORD_MACHINE,
                AsJobConstants.KEYWORD_JOB_CLASS,
                AsJobConstants.KEYWORD_SAP_CLIENT,
                AsJobConstants.KEYWORD_SAP_EVENT_ID,
                AsJobConstants.KEYWORD_SAP_JOB_NAME,
                AsJobConstants.KEYWORD_SAP_STEP_PARMS,
                AsJobConstants.KEYWORD_SAP_RECIPIENTS
            };

            // Ready the request and execute it.
            getReq.setRequest(filter, attributes);
            set = getReq.execute(appServer);

            // Process the returned responses.
            while(set.hasNext()) {
                getRsp = (GetJobsWithFilterRsp)set.next();
                int rspProperties[] = getRsp.getAttributes();

                // Process job level properties.
                System.out.println("----- Begin Response -----");
                for(int i=0; i<rspProperties.length; ++i) {
                    String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
                    switch(getRsp.getAttributeType(rspProperties[i])) {
                        case AsConstants.ATTRIBUTE_TYPE_STRING:
                            System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                            break;
                        case AsConstants.ATTRIBUTE_TYPE_INT:
                            System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                            break;
                        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                            System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
                    }
                }
            }
        }
    }
}
```

```
        break;
    case AsConstants.ATTRIBUTE_TYPE_LONG:
        System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
        System.out.println("\n----- Begin " + keyword + " JobItem
-----");
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {

            System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

            switch(jobProperties.getKeywordDataType(itemProperties[j])) {
                case AsConstants.ATTRIBUTE_TYPE_STRING:
                    System.out.println(ji.getString(itemProperties[j]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_INT:
                    System.out.println(ji.getInt(itemProperties[j]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                    System.out.println(ji.getBool(itemProperties[j]));
                    break;
            }
        }
        System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: sap_recipient_sample
alarm_if_fail: true
```

```
machine: localhost
job_class: myClass
sap_client: 235
sap_event_id: myEventId
sap_job_name: mySapJob

----- Begin sap_step_parms JobItem -----
sap_fail_msg:
sap_success_msg:
abap_name: RSLG0000
prt_arc_mode: -2147483647
sap_banner: -2147483647
banner_page: false
expiration: -2147483647
num_columns: -2147483647
copies: -2147483647
dept:
dest:
dataset:
footer: false
format:
host_page: -2147483647
print_imm: false
num_lines: -2147483647
new_spool: false
print_priority: -2147483647
release: false
req_type:
title:
spool_name:
abap_lang: EN
step_user:
variant:
arc_client: -2147483647
arc_connect:
arc_date:
arc_doc_class:
arc_doc_type:
arc_format:
arc_host_link:
arc_info:
arc_obj_type:
arc_path:
arc_printer:
arc_protocol:
arc_report:
arc_service:
arc_storage:
arc_text:
```

```
arc_userid:  
arc_version:  
web_posting: false  
recipient_name:  
----- End sap_step_parms JobItem -----  
----- Begin sap_recipients JobItem -----  
express: false  
send_company:  
no_forward: false  
no_print: false  
recipient: myAddress@ca.com  
rcp_type: 1  
blind_copy: false  
copy: false  
----- End sap_recipients JobItem -----  
----- End Response -----
```

Note: If numeric fields are null in the database, a constant value is returned to the caller. This value is mapped to NULL_INT_VALUE or NULL_LONG_VALUE as defined in the AsJobConstants class.

SapRecipientItem

This constructor has the following syntax:

```
public SapRecipientItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

SapStepParmItem

extends

JobItem

SapStepParmItem is the container class for specifying SAP step parameters in job definitions. This JobItem subclass directly maps to the sap_step_parms jil keyword.

Example: Sample Code for Using the SapStepParmItem Object

```
private static void sampleUsage(AsApi appServer) {  
    AsJobProperties jobProperties = new AsJobProperties();  
    InsertJobReq insertReq = new InsertJobReq();  
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();  
    GetJobsWithFilterRsp getRsp;  
    IApiResponseSet set;  
  
    // Try and insert the job.  
    try {  
        // Set up the base job definition.  
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);  
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,  
AsJobConstants.JOB_TYPE_SAP);  
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");  
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");  
        insertReq.set(AsJobConstants.KEYWORD_SAP_CLIENT, "235");  
        insertReq.set(AsJobConstants.KEYWORD_SAP_EVENT_ID, "myEventId");  
        insertReq.set(AsJobConstants.KEYWORD_SAP_JOB_NAME, "mySapJob");  
  
        // Set up the sap_step_parms of the job definition and  
        // attach it to the base job definition.  
        SapStepParmItem sspi = new SapStepParmItem();  
        sspi.set(AsJobConstants.KEYWORD_ABAP_LANG, "EN");  
        sspi.set(AsJobConstants.KEYWORD_ABAP_NAME, "RSLG0000");  
        insertReq.add(sspi);  
  
        // Ready the request and execute it.  
        insertReq.setRequest(false);  
        insertReq.execute(appServer);  
    } catch (AsFieldValidationException e) {  
        e.printStackTrace();  
    } catch (AsException e) {  
        e.printStackTrace();  
    }  
  
    // Try and retrieve the job.  
    try {  
        // Set up the job name mask.
```

```
JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

// Set up what attributes are to be returned.
int attributes[] = {
    AsJobConstants.KEYWORD_JOB_NAME,
    AsJobConstants.KEYWORD_ALARM_IF_FAIL,
    AsJobConstants.KEYWORD_MACHINE,
    AsJobConstants.KEYWORD_JOB_CLASS,
    AsJobConstants.KEYWORD_SAP_CLIENT,
    AsJobConstants.KEYWORD_SAP_EVENT_ID,
    AsJobConstants.KEYWORD_SAP_JOB_NAME,
    AsJobConstants.KEYWORD_SAP_STEP_PARMS
};

// Ready the request and execute it.
getReq.setRequest(filter, attributes);
set = getReq.execute(appServer);

// Process the returned responses.
while(set.hasNext()) {
    getRsp = (GetJobsWithFilterRsp)set.next();
    int rspProperties[] = getRsp.getAttributes();

    // Process job level properties.
    System.out.println("----- Begin Response -----");
    for(int i=0; i<rspProperties.length; ++i) {
        String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
        switch(getRsp.getAttributeType(rspProperties[i])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:
                System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_INT:
                System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_LONG:
                System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
                System.out.println("\n----- Begin " + keyword + " JobItem
-----");
                break;
        }
    }
}
```

```
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {

System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:

System.out.println(ji.getString(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_INT:

System.out.println(ji.getInt(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:

System.out.println(ji.getBool(itemProperties[j]));
            break;
        }

System.out.println("----- End " + keyword + " JobItem
-----\n");
        break;
    }
    System.out.println("----- End Response -----");
}

} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: sap_step_parms_sample
alarm_if_fail: true
machine: localhost
job_class: myClass
sap_client: 235
sap_event_id: myEventId
sap_job_name: mySapJob

----- Begin sap_step_parms JobItem -----
sap_fail_msg:
```

```
sap_success_msg:  
abap_name: RSLG0000  
prt_arc_mode: -2147483647  
sap_banner: -2147483647  
banner_page: false  
expiration: -2147483647  
num_columns: -2147483647  
copies: -2147483647  
dept:  
dest:  
dataset:  
footer: false  
format:  
host_page: -2147483647  
print_imm: false  
num_lines: -2147483647  
new_spool: false  
print_priority: -2147483647  
release: false  
req_type:  
title:  
spool_name:  
abap_lang: EN  
step_user:  
variant:  
arc_client: -2147483647  
arc_connect:  
arc_date:  
arc_doc_class:  
arc_doc_type:  
arc_format:  
arc_host_link:  
arc_info:  
arc_obj_type:  
arc_path:  
arc_printer:  
arc_protocol:  
arc_report:  
arc_service:  
arc_storage:  
arc_text:  
arc_userid:  
arc_version:  
web_posting: false  
recipient_name:  
----- End sap_step_parms JobItem -----  
----- End Response -----
```

Note: If numeric fields are null in the database, a constant value is returned to the caller. This value is mapped to NULL_INT_VALUE or NULL_LONG_VALUE as defined in the AsJobConstants class.

SapStepParmItem

This constructor has the following syntax:

```
public SapStepParmItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

SendEvt

SendEvt sends an event to CA Workload Automation AE.

Example: Sample Code for Executing the SendEvt API

```
private void sample(AsApi api) {
    SendEvtReq req = new SendEvtReq();
    req.setEventType(req.EVENT_COMMENT);
    req.setComment("This is a comment from Java");
    req.setPriority(1);
    req.setMaxSendTries(10);
    req.setJobQuePriority(5);
    req.setEventTime(new Date());
    req.setRequest();

    try {
        IApiResponseSet set = req.execute(api);

        while(set.hasNext()) {
            SendEvtRsp rsp = (SendEvtRsp)set.next();
            System.out.println("Object Name: " + rsp.getObjectName());
        }
    } catch (AsGeneralErrorException ge) {
        System.out.println("AsGeneralErrorException: " + ge.getMessage());
    } catch (AsSecurityException se) {
        System.out.println("AsSecurityException: " + se.getMessage());
    } catch (AsFieldValidationException fve) {
        System.out.println("AsFieldValidationException: " + fve.getMessage());
    } catch (AsException e) {
        System.out.println("AsException: " + e.getMessage());
    }
}
```

SendEvtReq

extends
EventRequest
implements
ISendEvtReq

Execution of this request and the subsequent interrogation of the response set can result in the following exceptions:

- AsGeneralErrorException - A general error has occurred.
- AsSecurityException - The set user does not have authority to send the event.
- AsFieldValidationException - The API returned an error. The event was not executed.

setRequest

The setRequest method in the SendEvtReq class sets up the request. This method has the following syntax:

```
public void setRequest()
```

SendEvtRsp

extends
ApiResponse
implements
ISendEvtRsp

getObjectName

The getObjectName method in the SendEvtRsp class returns the object name in the response. This method has the following syntax:

```
public java.lang.String getObjectName()
```

SnmpValueItem

extends

JobItem

SnmpValueItem is the container class for specifying SNMP values in job definitions. This JobItem subclass directly maps to the snmp_value jil keyword.

Example: Sample Code for Using the SnmpValueItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_SNMP_SET);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_SNMP_HOST, "mySNMPHost");
        insertReq.set(AsJobConstants.KEYWORD_SNMP_OID, "myOid");

        // Set up the snmp_value of the job definition and
        // attach it to the base job definition.
        SnmpValueItem svi = new SnmpValueItem();
        svi.set(AsJobConstants.KEYWORD_SNMP_VALUE, "addr=myAddr");
        svi.set(AsJobConstants.KEYWORD_SNMP_VALUE, "uint=myUnsignedInt");
        insertReq.add(svi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
        JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);
```

```
// Set up what attributes are to be returned.  
int attributes[] = {  
    AsJobConstants.KEYWORD_JOB_NAME,  
    AsJobConstants.KEYWORD_ALARM_IF_FAIL,  
    AsJobConstants.KEYWORD_MACHINE,  
    AsJobConstants.KEYWORD_JOB_CLASS,  
    AsJobConstants.KEYWORD_SNMP_HOST,  
    AsJobConstants.KEYWORD_SNMP_OID,  
    AsJobConstants.KEYWORD_SNMP_VALUE  
};  
  
// Ready the request and execute it.  
getReq.setRequest(filter, attributes);  
set = getReq.execute(appServer);  
  
// Process the returned responses.  
while(set.hasNext()) {  
    getRsp = (GetJobsWithFilterRsp)set.next();  
    int rspProperties[] = getRsp.getAttributes();  
  
    // Process job level properties.  
    System.out.println("----- Begin Response -----");  
    for(int i=0; i<rspProperties.length; ++i) {  
        String keyword =  
jobProperties.getKeywordFromConstant(rspProperties[i]);  
        switch(getRsp.getAttributeType(rspProperties[i])) {  
            case AsConstants.ATTRIBUTE_TYPE_STRING:  
                System.out.println(keyword + ": " +  
getRsp.getString(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_INT:  
                System.out.println(keyword + ": " +  
getRsp.getInt(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
                System.out.println(keyword + ": " +  
getRsp.getBoolean(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_LONG:  
                System.out.println(keyword + ": " +  
getRsp.getLong(rspProperties[i]));  
                break;  
            case AsConstants.ATTRIBUTE_TYPE_JOBITEM:  
                System.out.println("\n----- Begin " + keyword + " JobItem  
-----");  
                JobItem ji = getRsp.getJobItem(rspProperties[i]);  
                int itemProperties[] = ji.getAttributes();  
        }  
    }  
}
```

```
// Process job item level properties.  
for(int j=0; j<itemProperties.length; ++j) {  
  
    System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");  
  
    switch(jobProperties.getKeywordDataType(itemProperties[j])) {  
        case AsConstants.ATTRIBUTE_TYPE_STRING:  
  
            System.out.println(ji.getString(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_INT:  
  
            System.out.println(ji.getInt(itemProperties[j]));  
            break;  
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:  
  
            System.out.println(ji.getBool(itemProperties[j]));  
            break;  
    }  
}  
System.out.println("----- End " + keyword + " JobItem  
-----\n");  
break;  
}  
}  
System.out.println("----- End Response -----");  
}  
} catch (AsInitializationException e) {  
    e.printStackTrace();  
} catch (AsException e) {  
    e.printStackTrace();  
}  
}  
}
```

Sample output

```
----- Begin Response -----  
job_name: snmp_value_sample  
alarm_if_fail: true  
machine: localhost  
job_class: myClass  
snmp_host: mySNMPHost  
snmp_oid: myOid  
  
----- Begin snmp_value JobItem -----  
snmp_value: addr=myAddr  
snmp_value: uint=myUnsignedInt  
----- End snmp_value JobItem -----  
  
----- End Response -----
```

SnmpValueItem

This constructor has the following syntax:

```
public SnmpValueItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

.StoredProcedureArgumentItem

extends

JobItem

.StoredProcedureArgumentItem is the container class for specifying stored procedure arguments in job definitions. This JobItem subclass directly maps to the sp_arg jil keyword.

Example: Sample Code for Using the `.StoredProcedureArgumentItem` Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_DB_STORED_PROCEDURE);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_SP_NAME, "myProcName");

        // Set up the sp_arg of the job definition and
        // attach it to the base job definition.
        StoredProcedureArgumentItem spai = new StoredProcedureArgumentItem();
        spai.set(AsJobConstants.KEYWORD_ARGLTYPE,
        AsJobConstants.VALUE__DBARGTYPE__IN_OUT);
        spai.set(AsJobConstants.KEYWORD_DATATYPE,
        AsJobConstants.VALUE__DBDATATYPE__CHAR);
        spai.set(AsJobConstants.KEYWORD_IGNORE, true);
        spai.set(AsJobConstants.KEYWORD_NAME, "myName");
        spai.set(AsJobConstants.KEYWORD_VALUE, "myValue");
        insertReq.add(spai);

        // Set up a second sp_arg for the the job definition and
        // attach it to the base job definition.
        StoredProcedureArgumentItem spai2 = new StoredProcedureArgumentItem();
        spai2.set(AsJobConstants.KEYWORD_ARGLTYPE,
        AsJobConstants.VALUE__DBARGTYPE__IN);
        spai2.set(AsJobConstants.KEYWORD_DATATYPE,
        AsJobConstants.VALUE__DBDATATYPE__SMALLINT);
        spai2.set(AsJobConstants.KEYWORD_IGNORE, true);
        spai2.set(AsJobConstants.KEYWORD_NAME, "myName2");
        spai2.set(AsJobConstants.KEYWORD_VALUE, "myValue2");
    }
}
```

```

        insertReq.add(spai2);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

        // Set up what attributes are to be returned.
        int attributes[] = {
            AsJobConstants.KEYWORD_JOB_NAME,
            AsJobConstants.KEYWORD_ALARM_IF_FAIL,
            AsJobConstants.KEYWORD_MACHINE,
            AsJobConstants.KEYWORD_JOB_CLASS,
            AsJobConstants.KEYWORD_SP_NAME,
            AsJobConstants.KEYWORD_SP_ARG
        };
    }

    // Ready the request and execute it.
    getReq.setRequest(filter, attributes);
    set = getReq.execute(appServer);

    // Process the returned responses.
    while(set.hasNext()) {
        getRsp = (GetJobsWithFilterRsp)set.next();
        int rspProperties[] = getRsp.getAttributes();

        // Process job level properties.
        System.out.println("----- Begin Response -----");
        for(int i=0; i<rspProperties.length; ++i) {
            String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
            switch(getRsp.getAttributeType(rspProperties[i])) {
                case AsConstants.ATTRIBUTE_TYPE_STRING:
                    System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                    break;
                case AsConstants.ATTRIBUTE_TYPE_INT:
                    System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
            }
        }
    }
}

```

```
        break;
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
        System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_LONG:
        System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
        System.out.println("\n----- Begin " + keyword + " JobItem
-----");
        JobItem ji = getRsp.getJobItem(rspProperties[i]);
        int itemProperties[] = ji.getAttributes();

        // Process job item level properties.
        for(int j=0; j<itemProperties.length; ++j) {

System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:

System.out.println(ji.getString(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_INT:

System.out.println(ji.getInt(itemProperties[j]));
            break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:

System.out.println(ji.getBool(itemProperties[j]));
            break;
        }
    }
    System.out.println("----- End " + keyword + " JobItem
-----\n");
    break;
}
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: sp_arg_sample
alarm_if_fail: true
machine: localhost
job_class: myClass
sp_name: myProcName

----- Begin sp_arg JobItem -----
ignore: true
name: myName
argtype: 3
datatype: 1
value: myValue
----- End sp_arg JobItem -----


----- Begin sp_arg JobItem -----
ignore: true
name: myName2
argtype: 1
datatype: 9
value: myValue2
----- End sp_arg JobItem -----


----- End Response -----
```

.StoredProcedureArgumentItem

This constructor has the following syntax:

```
public StoredProcedureArgumentItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString

- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

TimeOfDay

extends

java.lang.Object

TimeOfDay is the container class that represents the time of day in hours and minutes.

TimeOfDay

This constructor has the following syntax:

```
public TimeOfDay(int hour, int minute)
```

getHour

This method gets the hour. This method has the following syntax:

```
public int getHour()
```

getMinute

This method gets the minute. This method has the following syntax:

```
public int getMinute()
```

toString

This method has the following syntax:

```
public java.lang.String toString()
```

This method overrides `toString` in the class `java.lang.Object`.

Tools

```
extends  
java.lang.Object
```

This class includes the common tools used by CA Workload Automation AE API users.

Tools

This constructor has the following syntax:

```
public Tools()
```

getDateFromString

This method converts a date string into a date object. This method has the following syntax:

```
public static java.util.Date getDateFromString(java.lang.String time_str)  
time_str
```

Specifies the string to express the date and time in the pattern MM/dd/yyyy
HH:mm

This method returns the object of Date class or null if time_str is not like MM/dd/yyyy
HH:mm.

getStringFromDate

This method converts a date object to a string expression in the pattern MM/dd/yyyy
HH:mm. This method has the following syntax:

```
public static java.lang.String getStringFromDate(java.util.Date date)  
date
```

Specifies the date object to convert.

getIntCode

This method gets the int code for a given type and string. This method has the following syntax:

```
public static int getIntCode(AsApi api, java.lang.String intCodeType,  
java.lang.String intCodeText)  
intCodeType
```

Specifies the IntCode type. Valid types are the following:

- INTCODE_TYPE_ALARM

- INTCODE_TYPE_EVENT
- INTCODE_TYPE_SOUND
- INTCODE_TYPE_STATUS

intCodeText

Specifies the intCode text that includes values such as "RUNNING", "STARTING", and so on.

This method returns the integer representation of the intcode or returns 0 if intCode is not found. This method throws an exception as explained in the AsException class if an exception is raised.

retrieveIntCodes

This method gets all of the intcodes from the given CA Workload Automation AE instance. This method has the following syntax:

```
public static java.util.Vector<IntCode> retrieveIntCodes(AsApi api)  
api
```

Specifies the connection to a CA Workload Automation AE instance.

This method returns the vector of intCode objects or if an exception is raised, throws an exception as explained in the AsException class.

getValueFromIntCode

This method gets the string value from an intcode. This method has the following syntax:

```
public static java.lang.String getValueFromIntCode(AsApi api, int code,  
java.lang.String type)
```

api

Specifies the connection to a CA Workload Automation AE instance.

code

Specifies the intcode to get the value for.

type

Specifies the intcode type. Valid types are the following:

- INTCODE_TYPE_ALARM
- INTCODE_TYPE_EVENT
- INTCODE_TYPE_SOUND
- INTCODE_TYPE_STATUS

This method returns the string value for the specified intcode/type or an empty string if undefined. This method throws AsException as explained in the AsException class if an exception is raised.

getIntCodeFromValue

This method gets the intcode from a string value. This method has the following syntax:

```
public static int getIntCodeFromValue(AsApi api, java.lang.String type,  
java.lang.String value)
```

api

Specifies the connection to an CA Workload Automation AE instance.

type

Specifies the intcode type. Valid types are the following:

- INTCODE_TYPE_ALARM
- INTCODE_TYPE_EVENT
- INTCODE_TYPE_SOUND
- INTCODE_TYPE_STATUS

value

Specifies the string value to get the intcode for.

This method returns the Integer code for the specified type/value combination or 0 if undefined. This method throws AsException as explained in the AsException class if an exception is raised.

INTCODE_TYPE_ALARM

This field has the following syntax:

```
public static final java.lang.String INTCODE_TYPE_ALARM
```

INTCODE_TYPE_EVENT

This field has the following syntax:

```
public static final java.lang.String INTCODE_TYPE_EVENT
```

INTCODE_TYPE_SOUND

This field has the following syntax:

```
public static final java.lang.String INTCODE_TYPE_SOUND
```

INTCODE_TYPE_STATUS

This field has the following syntax:

```
public static final java.lang.String INTCODE_TYPE_STATUS
```

UnsendEvt

UnsendEvt cancels the event when the specified event has not yet processed.

Example: Sample Code for Executing the UnsendEvt API

```
private void sample(AsApi api) {
    UnsendEvtReq req = new UnsendEvtReq();
    req.setEventType(req.EVENT_COMMENT);
    req.setPriority(1);
    req.setMaxSendTries(10);
    req.setJobQuePriority(5);
    req.setEventTime(eventTime);
    req.setRequest();

    try {
        IApiResponseSet set = req.execute(api);

        while(set.hasNext()) {
            UnsendEvtRsp rsp = (UnsendEvtRsp)set.next();
            System.out.println("Event Id: " + rsp.getEventId());
            System.out.println("Event Time: " +
Tools.getStringFromDate(rsp.getEventTime()));
        }
    } catch (AsGeneralErrorException ge) {
        System.out.println("AsGeneralErrorException: " + ge.getMessage());
    } catch (AsSecurityException se) {
        System.out.println("AsSecurityException: " + se.getMessage());
    } catch (AsFieldValidationException fve) {
        System.out.println("AsFieldValidationException: " + fve.getMessage());
    } catch (AsException e) {
        System.out.println("AsException: " + e.getMessage());
    }
}
```

UnsendEvtReq

extends

EventRequest

implements

IUnsendEvtReq

Execution of this request and the subsequent interrogation of the response set can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have authority to unsend the event.
- AsFieldValidationException—The API returned an error. The event was not executed.

UnsendEvtReq

The UnsendEvtReq constructor creates a UnsendEvtReq object. This constructor has the following syntax:

```
public UnsendEvtReq()
```

setRequest

The setRequest method in the UnsendEvtReq class sets up the request. This method has the following syntax:

```
public void setRequest()
```

UnsendEvtRsp

extends
 ApiResponse
implements
 IUnsendEvtRsp

UnsendEvtRsp

The UnsendEvtRsp constructor creates an UnsendEvtRsp object. This constructor has the following syntax:

```
public UnsendEvtRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

getEventTime

The getEventTime method in the UnsendEvtRsp class returns the requested event time. This method has the following syntax:

```
public java.util.Date getEventTime()
```

getEventId

The getEventID method in the UnsendEvtRsp class returns the requested event ID. This method has the following syntax:

```
public java.lang.String getEventId()
```

UnsendEvtRspSet

extends

ApiResponseSet

implements

IApiResponseSet

This is a response set produced from an unsend event request. This set returns response objects of type UnsendEvtRsp.

The following exceptions can be thrown when iterating over this set:

- AsApiGeneralException
- AsSecurityException—The set user does not have permission to send events for the defined job.

UnsendEvtRspSet

The UnsendEvtRspSet constructor creates a UnsendEvtRspSet object. This constructor has the following syntax:

`public UnsendEvtRspSet(com.ca.autosys.services.JResponseQ responseQ)`

responseQ

Defines the raw responseQ object.

UpdateJob

UpdateJob is the API for updating jobs in the database. This is the preferred method of updating jobs at CA Workload Automation AE r11.3 and later.

Example: Sample Code for Executing the UpdateJob API

```
private void sample(AsApi api) {
    UpdateJobReq req = new UpdateJobReq();
    UpdateJobRsp rsp;

    try {
        // Set up the job definition.
        req.set(AsJobConstants.KEYWORD_JOB_NAME, "sample_job");
        req.set(AsJobConstants.KEYWORD_COMMAND, "myNewCommand");
        req.set(AsJobConstants.KEYWORD_GROUP, "myGroup");
        req.setRequest(false);

        // Update the job.
        rsp = (UpdateJobRsp)req.execute(api);

        // Check if there were undefined dependent jobs.
        if(rsp.hasUndefinedDependentJobs()) {
            String dependents[] = rsp.getUndefinedDependentJobs();
            System.out.println("Dependent Jobs: ");
            for(int i=0; i<dependents.length; ++i) {
                System.out.println(dependents[i]);
            }
        }

        // Check if there were warnings on the update.
        if(rsp.hasWarnings()) {
            AsMessage msgs[] = rsp.getWarnings();
            System.out.println("Warnings: ");
            for(int i=0; i<msgs.length; ++i) {
                System.out.println(msgs[i]);
            }
        }
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }
}
```

UpdateJobReq

extends

Cat1Request

implements

IUpdateJobReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to update the named job.
- AsExternalValidationException—Failure in custom job validation.
- AsFieldValidationException—Field validation error.

UpdateJobReq

The UpdateJobReq constructor creates an UpdateJobReq object. This constructor has the following syntax:

```
public UpdateJobReq()
```

set

This set method in the UpdateJobReq class configures an integer job parameter. This method has the following syntax:

```
public void set(int keyword, int value)
```

keyword

Defines the keyword constant to be set.

value

Defines the integer value that keyword should be set to.

An AsFieldValidationException can be thrown.

set

This set method in the UpdateJobReq class configures a string job parameter. This method has the following syntax:

```
public void set(int keyword, java.lang.String value)
```

keyword

Defines the keyword constant to be set.

value

Defines the string value that keyword should be set to.

An AsFieldValidationException can be thrown.

set

This set method in the UpdateJobReq class configures a BLOB job parameter. This method has the following syntax:

```
public void set(int keyword, byte[] value)
```

keyword

Defines the keyword constant to be set.

value

Defines the byte array (BLOB) value that keyword should be set to.

An AsFieldValidationException can be thrown.

set

This set method in the UpdateJobReq class configures a Boolean job parameter. This method has the following syntax:

```
public void set(int keyword, boolean value)
```

keyword

Defines the keyword constant to be set.

value

Defines the Boolean value that keyword should be set to.

An AsFieldValidationException can be thrown.

set

This set method in the UpdateJobReq class configures a double job parameter. This method has the following syntax:

```
public void set(int keyword, double value)
```

keyword

Defines the keyword constant to be set.

value

Defines the double value that keyword should be set to.

An AsFieldValidationException can be thrown.

set

This set method in the UpdateJobReq class configures a long job parameter. This method has the following syntax:

```
public void set(int keyword, long value)
```

keyword

Defines the keyword constant to be set.

value

Defines the long value that keyword should be set to.

An AsFieldValidationException can be thrown.

setNull

The setNull method in the UpdateJobReq class sets a field to a null value. This will remove it from the job definition or set it to the default value. This method has the following syntax:

```
public void setNull(int keyword)
```

keyword

Defines the keyword constant to be set to null.

An AsFieldValidationException can be thrown.

setRequest

The setRequest method in the UpdateJobReq class configures the request and makes it ready for execution. This method has the following syntax:

```
public void setRequest(boolean verify)
```

verify

Specifies whether to validate job conditions. Possible values are:

- True—Validate the job conditions, reflect the results in the response
- False—Do not validate job conditions.

add

The add method in the UpdateJobReq class adds repeating job parameters to the job. This method has the following syntax:

```
public void add(JobItem itemObj)
```

itemObj

Defines the repeating job object that is to be added.

An AsFieldValidationException can be thrown.

UpdateJobRsp

extends

ApiResponse

implements

IUpdateJobRsp

UpdateJobRsp

The UpdateJobRsp constructor creates an UpdateJobRsp object. This constructor has the following syntax:

```
public UpdateJobRsp(com.ca.autosys.services.JResponse response)
```

response

Defines the raw response object.

getUndefinedDependentJobs

The getUndefinedDependentJobs method in the UpdateJobRsp class returns an array of job names specified in the job condition which are undefined. This array will only contain values if hasUndefinedDependentJobs() returns true. This method has the following syntax:

```
public java.lang.String[] getUndefinedDependentJobs()
```

hasUndefinedDependentJobs

The hasUndefinedDependentJobs method in the UpdateJobRsp class returns whether the conditions defined in the request specify undefined jobs. This can only be true if the client specified to "verify" in the request. This method has the following syntax:

```
public boolean hasUndefinedDependentJobs()
```

UpdateRealMachine

UpdateRealMachine lets you update a real machine. This API is used by jil to create real machines. The UpdateRealMachine API verifies that the caller has write access to the named machine. It then updates the machine and updates audit information.

Example: Sample Code for Executing the UpdateRealMachine API

```
private void sample(AsApi api) {
    // Set up the request.
    String machineName = "sampleRealMachine";
    IUpdateRealMachineReq req = new UpdateRealMachineReq();
    req.setRequest(machineName);

    // Execute the request, check for errors.
    try {
        req.execute(api);
        System.out.println("Machine \\" + machineName + "\\" has been updated.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

UpdateRealMachineReq

extends

Cat1Request

implements

IUpdateRealMachineReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to update the named machine.

UpdateRealMachineReq

The UpdateRealMachineReq constructor creates an UpdateRealMachineReq object. This constructor has the following syntax:

```
public UpdateRealMachineReq()
```

setRequest

The setRequest method in the UpdateRealMachineReq class configures the request on the specified machine. This method has the following syntax:

```
public void setRequest(java.lang.String mach_name)  
mach_name
```

Defines the name of the machine.

setType

The setType method in the UpdateRealMachineReq class configures the machine type. This method has the following syntax:

```
public void setType(int mach_type)  
mach_type
```

Defines the type of machine. Possible values are:

- MR_SYSTEM_AGENT—System Agent machine.
- MR_NT—Windows machine.
- MR_UNIX—UNIX machine.
- MR_UUJMA—NSM or a Universal Job Management Agent machine.
- MR_CONNECT—CA Workload Automation AE Connect machine.
- MR_UNIX_VIRTUAL—Virtual UNIX machine.

- MR_NT_LEGACY—NT legacy machine.
- MR_UNIX_LEGACY—UNIX legacy machine.

setMaxLoad

The setMaxLoad method in the UpdateRealMachineReq class defines the maximum load of the machine. This method has the following syntax:

```
public void setMaxLoad(int max_load)
```

max_load

Defines the maximum load of the machine.

setFactor

The setFactor method in the UpdateRealMachineReq class defines the weight factor of the machine. This method has the following syntax:

```
public void setFactor(double factor)
```

factor

Defines the weight factor of the machine.

Limits: 0-1024

setDescription

The setDescription method in the UpdateRealMachineReq class configures the machine description. This method has the following syntax:

```
public void setDescription(java.lang.String mach_desc)
```

mach_desc

Defines the description of the machine.

setnodeName

The setnodeName method in the UpdateRealMachineReq class configures the node name when the agent is running. This method has the following syntax:

```
public void setnodeName(java.lang.String nodename)
```

nodename

Defines the node name.

setPort

The setPort method in the UpdateRealMachineReq class configures the port number that the agent is listening on. This method has the following syntax:

```
public void setPort(int port)
```

port

Defines the listening port of the agent.

setCharacterCode

The setCharacterCode method in the UpdateRealMachineReq class configures the character encoding type. This method has the following syntax:

```
public void setCharacterCode(int char_code)
```

char_code

Defines the type of encoding. Possible values are:

- CC_ASCII—ASCII encoding
- CC_EBCDIC—EBCDIC encoding

setKeyToAgent

The setKeyToAgent method in the UpdateRealMachineReq class configures the encryption key used to send data to the agent. This method has the following syntax:

```
public void setKeyToAgent(java.lang.String key)
```

key

Defines the encryption key.

setEncryptionType

The setEncryptionType method in the UpdateRealMachineReq class configures the encryption algorithm type used by the agent. This method has the following syntax:

```
public void setEncryptionType(int type)
```

type

Defines the encryption type. Possible values are:

- ENCRYPTION_TYPE_NONE—No encryption.
- ENCRYPTION_TYPE_DEFAULT—Use default encryption algorithm and key.
- ENCRYPTION_TYPE_AES—Use AES 128-bit encryption algorithm.

setAgentName

The setAgentName method in the UpdateRealMachineReq class configures the internal name of the agent. This method has the following syntax:

```
public void setAgentName(java.lang.String agent_name)
```

agent_name

Defines the agent name.

setHeartbeatFreq

The setHeartbeatFreq method in the UpdateRealMachineReq class configures the heartbeat frequency. This method has the following syntax:

```
public void setHeartbeatFreq(int freq)
```

freq

Defines the number of seconds between heartbeat polls.

setHeartbeatAttempts

The setHeartbeatAttempts method in the UpdateRealMachineReq class configures the number of attempts to send a heartbeat. This method has the following syntax:

```
public void setHeartbeatAttempts(int attempts)
```

attempts

Defines the number of attempts to send a heartbeat.

setProvision

The setProvision method in the UpdateRealMachineReq class sets the flag to cause the agent to be provisioned automatically. This method has the following syntax:

```
public void setProvision(boolean provision)
```

provision

Defines the flag to indicate whether the machine should be provisioned.

setAdministrator

The setAdministrator method in the UpdateRealMachineReq class configures the administrator ID authorized to provision the agent. This method has the following syntax:

```
public void setAdministrator(java.lang.String admin)
```

admin

Defines the authorized administrator ID.

setOpSys

The `setOpSys` method in the `UpdateRealMachineReq` class configures the operating system. This method has the following syntax:

```
public void setOpSys(int opsys)  
opsys
```

Defines the operating system. Possible values are:

- `OPSYS_AIX`—AIX
- `OPSYS_HPUX`—HP-UX
- `OPSYS_I5OS`—i5/OS
- `OPSYS_LINUX`—Linux
- `OPSYS_OPENVMS`—OpenVMS
- `OPSYS_SOLARIS`—Solaris
- `OPSYS_TANDEM`—Tandem
- `OPSYS_WINDOWS`—Windows
- `OPSYS_ZOS`—z/OS

UpdateRealMachineRsp

extends

`ApiResponse`

UpdateRealMachineRsp

The `UpdateRealMachineRsp` constructor creates an `UpdateRealMachineRsp` object. This is a response for the update real machine request. This response contains no special information. This constructor has the following syntax:

```
public UpdateRealMachineRsp(com.ca.autosys.services.JResponse rsp)
```

response

Defines the raw response object.

A `java.lang.IllegalArgumentException` is thrown if a null response is given.

UpdateResource

UpdateResource lets you update resource information on a given machine.

Example: Sample Code for Executing the UpdateResource API

```
private void sample(AsApi api) {
    // Set up the request.
    String RESOURCE_NAME = "sampleResource";
    String MACHINE_NAME = "sampleMachine";
    IUpdateResourceReq req = new UpdateResourceReq();
    req.setName(RESOURCE_NAME);
    req.setDescription("");
    req.setMachine(MACHINE_NAME);
    req.setAmount("5");
    req.setRequest();
    req.setUser(TEST_USER, SERVER_NAME);

    // Execute the request, check for errors.
    try {
        req.execute(api);
        System.out.println("Resource " + RESOURCE_NAME + " has been inserted on
Machine " + MACHINE_NAME);
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

UpdateResourceReq

extends
Cat1Request
implements
IUpdateResourceReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error was encountered.
- AsSecurityException—The user does not have authority to modify the calendar. No modification will be performed.

UpdateResourceReq

The UpdateResourceReq constructor creates an UpdateResourceReq object. This constructor has the following syntax:

```
public UpdateResourceReq()
```

setRequest

The setRequest method in the UpdateResourceReq class configures the update resource request for processing. This method has the following syntax:

```
public void setRequest()
```

setName

The setName method in the UpdateResourceReq class is a setter method for the resource name to be updated. This method has the following syntax:

```
public void setName(java.lang.String resource_name)
```

resource_name

Specifies the name of the resource.

setMachine

The setMachine method in the UpdateResourceReq class is a setter method for the machine name on which the resource exists. This method has the following syntax:

```
public void setMachine(java.lang.String resource_machine)
```

resource_machine

Specifies the name of the machine.

setDescription

The setDescription method in the UpdateResourceReq class is a setter method for the resource description to be updated. This method has the following syntax:

```
public void setDescription(java.lang.String resource_description)  
resource_description
```

Specifies the description of the resource.

setAmount

The setAmount method in the UpdateResourceReq class is a setter method for resource amount to be updated. This method has the following syntax:

```
public void setAmount(java.lang.String resource_amount)  
resource_amount
```

Specifies the amount of the resource.

UpdateResourceRsp

extends

ApiResponse

implements

IUpdateResourceRsp

UpdateResourceRsp

The UpdateResourceRsp constructor creates an UpdateResourceRsp object. This constructor has the following syntax:

```
public UpdateResourceRsp(com.ca.autosys.services.JResponse response)  
response
```

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

UpdateVirtualMachine

UpdateVirtualMachine lets you update a virtual machine. This API is used by jil to create virtual machines. The UpdateVirtualMachine API verifies that the caller has write access to the named machine. It then updates the machine and updates audit information.

Example: Sample Code for Executing the UpdateVirtualMachine API

```
private void sample(AsApi api) {
    // Set up the component machines.
    ComponentMachine mach1 = new ComponentMachine("sampleCompMach1");
    ComponentMachine mach2 = new ComponentMachine("sampleCompMach2");
    ComponentMachine mach3 = new ComponentMachine("sampleCompMach3");
    ComponentMachine[] compMachines = {
        mach1, mach2, mach3
    };

    // Set up the request.
    String sampleMachine = "sampleVirtualMachine";
    UpdateVirtualMachineReq req = new UpdateVirtualMachineReq();
    req.setRequest(sampleMachine, compMachines);

    // Execute the request, check for errors.
    try {
        req.execute(api);
        System.out.println("Virtual machine \\" + sampleMachine + "\\" has been
updated.");
    } catch (AsGeneralErrorException genEx) {
        System.out.println("AsGeneralErrorException: " + genEx.getMessage());
    } catch (AsSecurityException secEx) {
        System.out.println("AsSecurityException: " + secEx.getMessage());
    } catch (AsException asEx) {
        System.out.println("AsException: " + asEx.getMessage());
    }
}
```

UpdateVirtualMachineReq

extends

Cat1Request

implements

IUpdateVirtualMachineReq

Execution of this request can result in the following exceptions:

- AsGeneralErrorException—A general error has occurred.
- AsSecurityException—The set user does not have permission to update the named machine.

UpdateVirtualMachineReq

The UpdateVirtualMachineReq constructor creates an UpdateVirtualMachineReq object. This constructor has the following syntax:

```
public UpdateVirtualMachineReq()
```

setRequest

The setRequest method in the UpdateVirtualMachineReq class configures the request. This method has the following syntax:

```
public void setRequest(java.lang.String machineName, ComponentMachine[] machines)  
machineName
```

Defines the name of the virtual machine.

```
machines
```

Defines an array of machines that will comprise this virtual machine.

UpdateVirtualMachineRsp

extends

ApiResponse

UpdateVirtualMachineRsp

The UpdateVirtualMachineRsp constructor creates an UpdateVirtualMachineRsp object. This is a response for the update virtual machine request. This response contains no special information. This constructor has the following syntax:

```
public UpdateVirtualMachineRsp(com.ca.autosys.services.JResponse rsp)  
rsp
```

Defines the raw response object.

A java.lang.IllegalArgumentException is thrown if a null response is given.

UserUpdateAlarm

UserUpdateAlarm updates the specified alarm to closed, acknowledged, or open. The user field in the alarm table is set to the user as indicated in ApiRequest.setUser(). Alarms are not secured, but the user that updates the alarm is tracked.

Example: Sample Code for Executing the UserUpdateAlarm API

```
private void sample(AsApi appServer) {
    // Set up the request to get all of the alarm eoids for the watcher1 job.
    GetAlarmsWithFilterReq req = new GetAlarmsWithFilterReq();
    int[] attributes = {
        GetAlarmsWithFilterReq.ATTR_EOID,
        GetAlarmsWithFilterReq.ATTR_STATE
    };
    AlarmFilterString filter = new
    AlarmFilterString(AlarmFilterString.FLT_JOB_NAME, "watcher1");
    req.setRequest(filter, attributes);

    // Set up the request object that will be used to set all of the
    // alarms to acknowledged.
    UserUpdateAlarmReq updReq;

    // Begin processing, and check for errors.
    try {
        // Get the list of alarms for the watcher1 job.
        ApiResponseSet rspSet = (ApiResponseSet)req.execute(appServer);

        // Spin through, get the eoids.
        // If the alarm is currently in OPEN state, mark it ACKNOWLEDGED.
        while(rspSet.hasNext()) {
            GetAlarmsWithFilterRsp rsp = (GetAlarmsWithFilterRsp)rspSet.next();

            // [0] in the array is the EOID.
            // [1] in the array is the STATE.
            int[] rspAttributes = rsp.getAttributes();

            // Get the update request object.
            updReq = new UserUpdateAlarmReq();

            // Check if the state was = OPEN.
            // If yes, process it.
            // If no, output a message with the current state of the EOID.
            if(rsp.getInt(rspAttributes[1]) == UserUpdateAlarmReq.OPEN) {
                updReq.setRequest(rsp.getString(rspAttributes[0]), "Setting alarm
to ACKNOWLEDGED", UserUpdateAlarmReq.ACKNOWLEDGED);
                updReq.execute(appServer);
                System.out.println("EOID " + rsp.getString(rspAttributes[0]) + "
successfully ACKNOWLEDGED");
            }
        }
    }
}
```

```
        } else {
            String state = "";

            // Get the current alarm state.
            switch(rsp.getInt(rspAttributes[1])) {
                case UserUpdateAlarmReq.ACNOWLEDGED:
                    state = "ACKNOWLEDGED";
                    break;
                case UserUpdateAlarmReq.CLOSED:
                    state = "CLOSED";
                    break;
                case UserUpdateAlarmReq.OPEN:
                    state = "OPEN";
                    break;
            }
            System.out.println("EOID " + rsp.getString(rspAttributes[0]) + " not altered, status was = " + state + ".");
        }
    } catch (AsGeneralErrorException genEx) {
    System.out.println("AsGeneralErrorException: " + genEx.getMessage());
} catch (AsNoAttributesException noAttrEx) {
    System.out.println("AsNoAttributesException: " + noAttrEx.getMessage());
} catch (AsBadAttributesException badAttrEx) {
    System.out.println("AsBadAttributesException: " + badAttrEx.getMessage());
} catch (AsBadFilterFieldsException badFiltEx) {
    System.out.println("AsBadFilterFieldsException: " + badFiltEx.getMessage());
} catch (AsException asEx) {
    System.out.println("AsException: " + asEx.getMessage());
}
}
```

UserUpdateAlarmReq

extends
Cat1Request
implements
IUserUpdateAlarmReq

Execution of this request can result in AsGeneralErrorException being thrown.

setRequest

The setRequest method in the UserUpdateAlarmReq class specifies the status of the alarm. This method has the following syntax:

```
void setRequest(java.lang.String alarmId,java.lang.String comment,int alarmState)  
alarmId
```

Identifies the Eoid of the alarm to update.

comment

Indicates the resolution.

alarmState

Sets the updated state of the alarm. Possible values are:

- OPEN
- ACKNOWLEDGED
- CLOSED

UserUpdateAlarmRsp

extends
ApiResponse

The UserUpdateAlarmRsp class comprises the response for the UserUpdateAlarm request. This response contains no special information. An exception is thrown if an error is detected.

WebParameterItem

extends

JobItem

WebParameterItem is the container class for specifying web parameters in job definitions. This JobItem subclass directly maps to the web_parameter jil keyword.

Example: Sample Code for Using the WebParameterItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_J2EE_WEB_SERVICE);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");
        insertReq.set(AsJobConstants.KEYWORD_JOB_CLASS, "myClass");
        insertReq.set(AsJobConstants.KEYWORD_ENDPOINT_URL, "myEndPoint");
        insertReq.set(AsJobConstants.KEYWORD_TARGET_NAMESPACE, "myNamespace");
        insertReq.set(AsJobConstants.KEYWORD_WSDL_OPERATION, "myOperation");
        insertReq.set(AsJobConstants.KEYWORD_WSDL_URL, "myUrl");

        // Set up the web_parameter of the job definition and
        // attach it to the base job definition.
        WebParameterItem wpi = new WebParameterItem();
        wpi.set(AsJobConstants.KEYWORD_WEB_PARAMETER, "parm1=p1");
        wpi.set(AsJobConstants.KEYWORD_WEB_PARAMETER, "parm2=p2");
        insertReq.add(wpi);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
```

```
JobFilterString filter = new
JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

// Set up what attributes are to be returned.
int attributes[] = {
    AsJobConstants.KEYWORD_JOB_NAME,
    AsJobConstants.KEYWORD_ALARM_IF_FAIL,
    AsJobConstants.KEYWORD_MACHINE,
    AsJobConstants.KEYWORD_JOB_CLASS,
    AsJobConstants.KEYWORD_ENDPOINT_URL,
    AsJobConstants.KEYWORD_TARGET_NAMESPACE,
    AsJobConstants.KEYWORD_WSDL_OPERATION,
    AsJobConstants.KEYWORD_WSDL_URL,
    AsJobConstants.KEYWORD_WEB_PARAMETER
};

// Ready the request and execute it.
getReq.setRequest(filter, attributes);
set = getReq.execute(appServer);

// Process the returned responses.
while(set.hasNext()) {
    getRsp = (GetJobsWithFilterRsp)set.next();
    int rspProperties[] = getRsp.getAttributes();

    // Process job level properties.
    System.out.println("----- Begin Response -----");
    for(int i=0; i<rspProperties.length; ++i) {
        String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
        switch(getRsp.getAttributeType(rspProperties[i])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:
                System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_INT:
                System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_LONG:
                System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
```

```
System.out.println("\n----- Begin " + keyword + " JobItem\n-----");
JobItem ji = getRsp.getJobItem(rspProperties[i]);
int itemProperties[] = ji.getAttributes();

// Process job item level properties.
for(int j=0; j<itemProperties.length; ++j) {

    System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

    switch(jobProperties.getKeywordDataType(itemProperties[j])) {
        case AsConstants.ATTRIBUTE_TYPE_STRING:
            System.out.println(ji.getString(itemProperties[j]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_INT:
            System.out.println(ji.getInt(itemProperties[j]));
            break;
        case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
            System.out.println(ji.getBool(itemProperties[j]));
            break;
    }
}
System.out.println("----- End " + keyword + " JobItem\n-----\n");
break;
}
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: web_parameter_sample
alarm_if_fail: true
machine: localhost
job_class: myClass
endpoint_url: myEndPoint
target_namespace: myNamespace
wsdl_operation: myOperation
wsdl_url: myUrl
```

```
----- Begin web_parameter JobItem -----
web_parameter: parm1=p1
web_parameter: parm2=p2
----- End web_parameter JobItem -----

----- End Response -----
```

WebParameterItem

This constructor has the following syntax:

```
public WebParameterItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

ZOsConditionCodeItem

extends

JobItem

ZOsConditionCodeItem is the container class for specifying condition code arguments in job definitions. This JobItem subclass directly maps to the condition_code jil keyword.

Example: Sample Code for Using the ZOsConditionCodeItem Object

```
private static void sampleUsage(AsApi appServer) {
    AsJobProperties jobProperties = new AsJobProperties();
    InsertJobReq insertReq = new InsertJobReq();
    GetJobsWithFilterReq getReq = new GetJobsWithFilterReq();
    GetJobsWithFilterRsp getRsp;
    IApiResponseSet set;

    // Try and insert the job.
    try {
        // Set up the base job definition.
        insertReq.set(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME);
        insertReq.set(AsJobConstants.KEYWORD_JOB_TYPE,
        AsJobConstants.JOB_TYPE_ZOS_REGULAR);
        insertReq.set(AsJobConstants.KEYWORD_MACHINE, "localhost");

        // Set up the condition_code of the job definition and
        // attach it to the base job definition.
        ZOsConditionCodeItem zocci = new ZOsConditionCodeItem();
        zocci.set(AsJobConstants.KEYWORD_RC, "4");
        zocci.set(AsJobConstants.KEYWORD_RESULT,
        AsJobConstants.VALUE__CONDITION_CODE_RESULT__FAILURE);
        zocci.set(AsJobConstants.KEYWORD_ACTION,
        AsJobConstants.VALUE__CONDITION_CODE_ACTION_STOP);
        zocci.set(AsJobConstants.KEYWORD_PROC_STEP, "PROC");
        zocci.set(AsJobConstants.KEYWORD_PROGRAM, "PROGRAM");
        zocci.set(AsJobConstants.KEYWORD_STEP_NAME, "STEP1");
        insertReq.add(zocci);

        // Ready the request and execute it.
        insertReq.setRequest(false);
        insertReq.execute(appServer);
    } catch (AsFieldValidationException e) {
        e.printStackTrace();
    } catch (AsException e) {
        e.printStackTrace();
    }

    // Try and retrieve the job.
    try {
        // Set up the job name mask.
        JobFilterString filter = new
        JobFilterString(AsJobConstants.KEYWORD_JOB_NAME, JOB_NAME, JobFilterString.EQUAL);

        // Set up what attributes are to be returned.
        int attributes[] = {
            AsJobConstants.KEYWORD_JOB_NAME,
            AsJobConstants.KEYWORD_ALARM_IF_FAIL,
            AsJobConstants.KEYWORD_MACHINE,
```

```
AsJobConstants.KEYWORD_CONDITION_CODE
};

// Ready the request and execute it.
getReq.setRequest(filter, attributes);
set = getReq.execute(appServer);

// Process the returned responses.
while(set.hasNext()) {
    getRsp = (GetJobsWithFilterRsp)set.next();
    int rspProperties[] = getRsp.getAttributes();

    // Process job level properties.
    System.out.println("---- Begin Response ----");
    for(int i=0; i<rspProperties.length; ++i) {
        String keyword =
jobProperties.getKeywordFromConstant(rspProperties[i]);
        switch(getRsp.getAttributeType(rspProperties[i])) {
            case AsConstants.ATTRIBUTE_TYPE_STRING:
                System.out.println(keyword + ": " +
getRsp.getString(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_INT:
                System.out.println(keyword + ": " +
getRsp.getInt(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
                System.out.println(keyword + ": " +
getRsp.getBoolean(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_LONG:
                System.out.println(keyword + ": " +
getRsp.getLong(rspProperties[i]));
                break;
            case AsConstants.ATTRIBUTE_TYPE_JOBITEM:
                System.out.println("\n---- Begin " + keyword + " JobItem
----");
                JobItem ji = getRsp.getJobItem(rspProperties[i]);
                int itemProperties[] = ji.getAttributes();

                // Process job item level properties.
                for(int j=0; j<itemProperties.length; ++j) {

System.out.print(jobProperties.getKeywordFromConstant(itemProperties[j]) + ": ");

switch(jobProperties.getKeywordDataType(itemProperties[j])) {
                case AsConstants.ATTRIBUTE_TYPE_STRING:

System.out.println(ji.getString(itemProperties[j]));

```

```
        break;
    case AsConstants.ATTRIBUTE_TYPE_INT:
        System.out.println(ji.getInt(itemProperties[j]));
        break;
    case AsConstants.ATTRIBUTE_TYPE_BOOLEAN:
        System.out.println(ji.getBool(itemProperties[j]));
        break;
    }
}
System.out.println("----- End " + keyword + " JobItem
-----\n");
break;
}
}
System.out.println("----- End Response -----");
}
} catch (AsInitializationException e) {
    e.printStackTrace();
} catch (AsException e) {
    e.printStackTrace();
}
}
```

Sample output

```
----- Begin Response -----
job_name: condition_code_sample
alarm_if_fail: true
machine: localhost

----- Begin condition_code JobItem -----
action: 2
proc_step: PROC
program: PROGRAM
rc: 4
result: 2
step_name: STEP1
----- End condition_code JobItem -----
----- End Response -----
```

ZOsConditionCodeItem

This constructor has the following syntax:

```
public ZOsConditionCodeItem()
```

Methods Inherited

This class inherits the following methods from the JobItem class:

- add
- getAttributes
- getBool
- getInt
- getItemType
- getJobItem
- getLoggableItem
- getString
- hasSubItems
- set(int keyword, boolean value)
- set(int keyword, int value)
- set(int keyword, java.lang.String value)
- toString

Chapter 3: C++ APIs

This chapter provides an alphabetical reference to each of the C++ Application Programming Interfaces (APIs) that make up the CA Workload Automation AE Software Development Kit (SDK).

ApiExceptionWithInfo

The ApiExceptionWithInfo class is used to relay a CA Workload Automation AE message from the CA Workload Automation AE API to client applications.

Header

AsPublicSDK.h

what

The what method in the ApiExceptionWithInfo class returns information and error messages that explain the reason for the exception. This method has the following prototype:

```
what () const [virtual]
```

ApiRequest

Header

AsPublicSDK.h

ApiRequest is the base class for the common functionality of all API requests. It is not used directly because it is not associated with any particular API. Only its subclasses associated with APIs are used.

setUser

The setUser method in the ApiRequest class defines the user in this request. This method has the following syntax:

```
void setUser(const std::string& user, const std::string& machine);
```

user

Sets the user on whose behalf this request is executed. The setUser method requires a prior call to AsApi.authenticateWithPassword.

machine

Sets the computer on which this user is defined.

ApiResponse

Header

AsPublicSDK.h

ApiResponse is the base class for the common functionality of all API response classes.

info

The info method in the ApiResponse class returns information and error messages from the application server. This method has the following syntax:

```
std::string info() const;
```

Note: Use the hasInfo method to determine whether a message string is available.

hasInfo

The hasInfo method in the ApiResponse class returns TRUE if the response contains a message. This method has the following syntax:

```
bool hasInfo() const;
```

Note: Use the info method to retrieve the message.

status

The status method in the ApiResponse class returns the response status. This method has the following syntax:

```
eResponseStatus status() const;
```

The eResponseStatus enumerations are as follows:

- RS_NO_OVERRIDE_WARNING
- RS_SUCCESS
- RS_FAILURE
- RS_ACCESS_DENIED
- RS_EXTERNAL_VALIDATION_FAILURE
- RS_FIELD_VALIDATION_FAILURE
- RS_NO_ATTRIBUTES
- RS_BAD_ATTRIBUTES
- RS_BAD_FILTER_FIELDS
- RS_OBJECT_DOESNT_EXIST

- RS_BAD_JOB_TYPE
- RS_OTHER_FAILURE

ApiResponseSet

Header

AsPublicSDK.h

ApiResponseSet is responsible for making multiple responses possible from APIs.

hasMore

The hasMore method in the ApiResponseSet class indicates if there are more responses to be processed. This method has the following syntax:

```
bool hasMore();
```

cancel

The cancel method in the ApiResponseSet class is used to prematurely end processing of the response set. A program that is processing a response set typically dequeues each response with the next method. The cancel method enables a program to dequeue the remaining responses without having to repeatedly call the next method. This method has the following syntax:

```
void cancel();
```

info

The info method in the ApiResponseSet class returns information, warning, and error messages from the application server. This method has the following syntax:

```
std::string info() const;
```

Note: Use the hasInfo method to determine whether a message string is available.

hasInfo

The hasInfo method in the ApiResponseSet class returns TRUE if the response contains a message. This method has the following syntax:

```
bool hasInfo() const;
```

Note: Use the info method to retrieve the message.

status

The status method in the ApiResponseSet class returns the response status. This method has the following syntax:

```
eResponseStatus status() const;
```

The eResponseStatus enumerations are as follows:

- RS_NO_OVERRIDE_WARNING
- RS_SUCCESS
- RS_FAILURE
- RS_ACCESS_DENIED
- RS_EXTERNAL_VALIDATION_FAILURE
- RS_FIELD_VALIDATION_FAILURE
- RS_NO_ATTRIBUTES
- RS_BAD_ATTRIBUTES
- RS_BAD_FILTER_FIELDS
- RS_OBJECT_DOESNT_EXIST
- RS_BAD_JOB_TYPE
- RS_OTHER_FAILURE

ApiTimeout

The ApiTimeout class is used to indicate that a request timeout error has occurred when attempting to communicate with the application server.

AsApi

AsApi represents the connection to the CA Workload Automation AE Application Server. There is no restriction on creation of multiple API objects. AsApi provides the ability to control the overhead imposed by the API. If you have only a limited interaction with the API, create the object only when it is needed and then discard it. For an application that interacts constantly with the API, create the object once at startup.

Requests can be issued on behalf of other users. By default, a request is executed on behalf of the user that is currently running the process as identified by the operating system. However, if the code calls ApiRequest.setUser(), it causes the request to be done on behalf of the specified user. This works only if the user has previously called AsApi.authenticateWithPassword() for the given user.

AsApi

This AsApi constructor creates a thin client API object. This constructor has the following syntax:

```
AsApi();
```

openLocal

This openLocal method opens a connection to a local application server. This constructor has the following syntax:

```
void openLocal(const std::string& app_name);
```

openRemote

This openRemote method opens a connection to a remote application server. This constructor has the following syntax:

```
void openRemote(int port, const std::string& Host;
```

port

Specifies the CA Workload Automation AE server port.

Host

Specifies the CA Workload Automation AE server host.

openRemote

This openRemote method opens a connection to a remote application server defined in the local configuration. This constructor has the following syntax:

```
void openRemote();
```

authenticateWithArtifact

The authenticateWithArtifact method in the AsApi class authenticates an EEM artifact and returns the AuthenticatedUser object that contains the user and node. This method has the following syntax:

```
eAuthResult authenticateWithArtifact(const std::string& artifact_in, std::string& user_out, std::string& node_out);
```

artifact

Specifies the artifact string to authenticate.

Returns one of the following (see the eAuthResult enumeration in AsPublicSDK.h):

- AR_VALIDATION_SUCCESS - user_out will contain the authorized user to be used in future API requests node_out will contain the authorized node to be used in future API requests
- AR_VALIDATION_FAILURE - artifact is not valid or the EEM security server is unreachable
- AR_VALIDATION_INTERNALERR - CA Workload Automation AE Instance has not been configured to run under EEM Toolkit Security
- AR_TIMEOUT – the CA Workload Automation AE application server is not responding
- AR_TX_SYSTEM_FAILURE

authenticateWithPassword

The authenticateWithPassword method in the AsApi class authenticates a user against CA Workload Automation AE security. This method has the following syntax:

```
eAuthResult authenticateWithPassword(const std::string& userName, const std::string& node, const std::string& password);
```

userName

Specifies the fully qualified user name.

node

Specifies the CA Workload Automation AE node on which to authenticate the user.

password

Specifies the user password.

Returns one of the following (see the eAuthResult enumeration in AsPublicSDK.h):

- AR_VALIDATION_SUCCESS - user_out will contain the authorized user to be used in future API requests node_out will contain the authorized node to be used in future API requests
- AR_VALIDATION_FAILURE - artifact is not valid or the EEM security server is unreachable
- AR_VALIDATION_INTERNALERR - CA Workload Automation AE Instance has not been configured to run under EEM Toolkit Security
- AR_TIMEOUT – the CA Workload Automation AE application server is not responding
- AR_TX_SYSTEM_FAILURE

isOpen

The isOpen method indicates whether a connection to an application server is open. This method has the following syntax:

```
bool isOpen() const;
```

close

The close method closes the connection to an application server. This method has the following syntax:

```
void close();
```

AsExternalValidationException

The AsExternalValidationException class is used to relay external validation failure messages from the CA Workload Automation AE API to client applications. This failure occurs when the external job validation callout indicates a failure.

AsFieldValidationException

The AsFieldValidationException class is used to relay field validation failure messages from the CA Workload Automation AE API to client applications. For example, when creating a job, this exception occurs if you try to create a command job using the InsertJob API without specifying a machine.

AsGeneralErrorException

The AsGeneralErrorException class is used to relay general error messages from the CA Workload Automation AE API to client applications. You need to examine the contents to understand the error that generated this exception.

AsJobProperties

The AsJobProperties class is a utility class which is used to retrieve the valid keywords for a job type. For example, AsJobProperties could be used to retrieve all of the valid keywords and the corresponding enumerated values for the command job type (see the sample below).

The data contained in the AsJobProperties object is static, read-only, and tied to the CA Workload Automation AE release. This makes the data re-usable across instances that are running the same release of CA Workload Automation AE.

Creating this object can be resource intensive. We recommend that users of this object create it once, and keep re-using that reference throughout their applications.

Example: Sample Code for Executing the AsJobProperties API

```
AsJobProperties jobprops;
std::vector<std::string> keywords_;
keywords_ = jobprops.getKeywordsForJobType(TYPES_CMD);
std::cout << "-----" << jobprops.getJobTypeString(TYPES_CMD) << " -----"
<< std::endl;
for(size_t i = 0; i < keywords_.size(); i++ ) {
    std::cout << "Keyword=" << keywords_[i] << ", keyword enum value=" <<
jobprops.getConstantFromKeyword(keywords_[i],TYPES_CMD) << std::endl;
}
```

getConstantFromKeyword

The `getConstantFromKeyword` method in the `AsJobProperties` class returns the integer constant for the specified keyword. This method has the following prototype:

```
int getConstantFromKeyword (const std::string & keyword, int jobtype)
```

keyword

Specifies the keyword, for example, "KW JOB NAME". See the `eKeywords` enumeration in `AsPublicSDK.h` for the complete list of keywords.

jobtype

Specifies the jobtype that the keyword is associated with, for example, "EN_JOBTYPE_SQL". See the `e_JOBTYPE` enumeration in `AsPublicSDK.h` for the complete list of job types.

This method returns an integer representation of the keyword or -1 if the keyword is undefined.

getKeywordFromConstant

The `getKeywordFromConstant` method in the `AsJobProperties` class returns the keyword from the specified integer constant. This method has the following prototype:

```
std::string getKeywordFromConstant (int constant)
```

constant

Specifies the integer constant

This method returns the string representation of the keyword or null if the constant is undefined.

getKeywordsForJobType

The `getKeywordsForJobType` method in the `AsJobProperties` class returns a list of all valid keywords for a given job type. This method has the following prototype:

```
std::vector<std::string> getKeywordsForJobType (int jobtype)
```

jobtype

Specifies the job type to be retrieved, for example, "EN_JOBTYPE_CMD".

This method returns a string vector array containing the valid keywords for the job type, or null if job type is undefined.

getJobTypeString

The getJobTypeString method in the AsJobProperties class returns the string value for the given job type constant. This method has the following prototype:

```
std::string getJobTypeString (int jobtype)
```

jobtype

Specifies the job type.

This method returns a string representation of the given job type.

getJobTypeInt

The getJobTypeInt method in the AsJobProperties class returns the numeric value for the given job type string. This method has the following prototype:

```
int getJobTypeInt (const std::string jobtype)
```

jobtype

Specifies the job type.

This method returns the integer representation of the given job type.

getKeywordDataType

The getKeywordDataType method in the AsJobProperties class returns the data type of a keyword value. This method has the following prototype:

```
int getKeywordDataType (int keyword)
```

keyword

Specifies the integer constant for the desired keyword.

This method returns an enumerated representation of a keyword's data type. See the eDBType enumeration in ASPublicSDK.h for more information.

AsObjectDoesNotExistException

The AsObjectDoesNotExistException class is used to indicate that the requested object does not exist.

AsSecurityException

The AsSecurityException class is used to relay security related error and warning messages from the CA Workload Automation AE API to client applications.

AsUnknownErrorException

The AsUnknownErrorException class catches exceptions for unknown API errors. This exception is not meant to be used, but is provided as a fail-safe for unexpected errors. One possible scenario for this exception could be as follows: if the client libraries are communicating with a server that implements a newer version of the API that has additional error codes.

Cat2StartFailure

The Cat2StartFailure class is used to relay a failure to start the thread responsible for filling an ApiResponseSet.

CreateParameterItem

The CreateParameterItem class is the container class for specifying create parameter arguments in job definitions.

Example: Sample Code for Executing the CreateParameterItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string)"myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_SESSBEAN);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_BEAN_NAME, (std::string) "mybean");
    req.set(KW_INITIAL_CONTEXT_FACTORY, (std::string) "mycontext");
    req.set(KW_METHOD_NAME, (std::string) "mymethod");
    req.set(KW_PROVIDER_URL, (std::string) "myurl");
    CreateParameterItem cp;
    cp.set(KW_CREATE_PARAMETER, (std::string) "param1=p1");
    cp.set(KW_CREATE_PARAMETER, (std::string) "param1=p2");
    cp.set(KW_CREATE_PARAMETER, (std::string) "param1=p2");
    req.add(&cp);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_CREATE_PARAMETER};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) / 
sizeof(int) );

// Create filter to select only jobs that begin with "my".

FilterString* name_filter = new FilterString(J_JobName, "my%");
try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);
    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
        rset.next(rsp);
        for (int i=0; i < rsp.getAttributes(); i++) {
```

```

        // Display keywords that are of type string.

        if( rsp.getAttributeType(i) == AtString ) {
            std::cout << "String attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
        }

        // Display job items.

        else if( rsp.getAttributeType(i) == AtJobItem ) {
            autosys::JobItem myJobitem = rsp.getJobItem(i);
            std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
            for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                // Only displaying jobitem keywords that are of type string.
                std::cout << "Jobitem string attribute [" <<
rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                myJobitem.getString( (int)keywords[j] ) << "]" <<
std::endl;
            }
        }
    }
}

catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
std::endl;
}

```

Sample output:

```

String attribute [job_name] has value [myjob]
String attribute [machine] has value [localhost]
Jobitem string attribute [create_parameter] has value [param1=p1]
Jobitem string attribute [create_parameter] has value [param1=p2]
Jobitem string attribute [create_parameter] has value [param1=p2]

```

CredsNotAuthorized

The CredsNotAuthorized class is used to indicate that an attempt was made to execute an API with credentials that were not previously authorized with a call to AsApi::authUser().

DeleteBox

DeleteBox deletes a box and all of the jobs it contains. DeleteBox verifies that the caller has delete permissions for the specified box, then retrieves all contained jobs and deletes them. The name of each job is returned in individual responses as it is deleted. This API is equivalent to the jil "delete_box" command.

DeleteBox is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the DeleteBox API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);

    DeleteBoxReq req;
    req.setName("mybox");
    DeleteBoxRspSet rset;
    req.execute(api, rset);

    // Loop thru response set and print names of deleted jobs.
    while (rset.hasMore()) {
        DeleteBoxRsp rsp;
```

```
rset.next(rsp);
if( rsp.hasInfo() )
    std::cout << rsp.info() << std::endl;
if( rsp.jobDeleted() )
    std::cout << "DeleteJob: " << rsp.getDeletedJob() << std::endl;
}
}
catch(ApiTimeout&)
{
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&)
{
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&)
{
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&)
{
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&)
{
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&)
{
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&)
{
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&)
{
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&)
{
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&)
{
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&)
{
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&)
{
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&)
{
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&)
{
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
```

```
        catch(ResponseDequeueFailure&) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
        api.close();
```

Sample output:

```
DeleteJob: mybox
DeleteJob: job1
DeleteJob: job2
```

DeleteBoxReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the DeleteBoxReq class performs the API call. This method has the following prototype:

 void execute (AsApi &*api*, DeleteBoxRspSet &*rset*)

api

 Specifies an AsApi object.

rset

 Specifies a DeleteBoxRspSet object.

The execute status is returned in the inherited response set method *rset.status()*.

setListDependentJobs

The `setListDependentJobs` method in the `DeleteBoxReq` class specifies whether a list of jobs dependent on the job being deleted is to be returned. The list of dependent jobs is returned in the `DeleteBoxRsp` object. This method has the following prototype:

```
void setListDependentJobs (bool bList)
```

bList

Possible values are:

true

Return a list of jobs that are dependent on the job being deleted.

false

Do not return a list of dependent jobs in the response object.

setName

The `setName` method in the `DeleteBoxReq` class specifies the name of the box to delete. This method has the following prototype:

```
void setName (const std::string &strBoxName)
```

setRestrictDeleteBox

The `setRestrictDeleteBox` method in the `DeleteBoxReq` class aborts the `DeleteBox` if a job that is contained in the box is in ACTIVATED, STARTING or RUNNING state. This method has the following prototype:

```
void setRestrictDeleteBox (bool bRestrictDeleteBox)
```

bRestrictDeleteBox

Specifies that the delete should fail if a job is in ACTIVATED, STARTING, or RUNNING state. Possible values are:

true

Fail delete if the job is in an ACTIVATED, STARTING, or RUNNING state.

false

Do not restrict DeleteBox.

setRestrictDeleteDependentJob

The setRestrictDeleteDependentJob method in the DeleteBoxReq class aborts the DeleteBox if a job is encountered that still has dependent jobs. This method has the following prototype:

```
void setRestrictDeleteDependentJob (bool bRestrictDeleteDependentJob)
```

bRestrictDeleteDependentJob

Specifies whether the API should fail if the job being deleted has dependent jobs.
Possible values are:

true

Fails if a job is encountered that still has dependent jobs.

false

Do not restrict DeleteBox.

setVerify

The setVerify method in the DeleteBoxReq class lets you run DeleteBox in test mode so that you can obtain a list of jobs that are contained in the box being deleted. This method has the following prototype:

```
void setVerify (bool bVerify)
```

bVerify

Indicates that DeleteBox should run in test mode. Possible values are:

true

Retrieves the list of jobs that would be deleted instead of actually deleting those jobs.

false

Delete the jobs.

DeleteBoxRsp

```
extends  
    ApiResponse  
header  
    AsPublicSDK.h
```

getDeletedJob

The getDeletedJob method in the DeleteBoxRsp class returns the name of a job deleted as a result of deleting the box. This method has the following prototype:

```
std::string getDeletedJob () const
```

getDependentJob

The getDependentJob method in the DeleteBoxRsp class returns a job that depends on the job that was deleted. This method has the following prototype:

```
std::string getDependentJob ( int i )const
```

i

Specifies the index into the job list.

getDependentJobCount

The getDependentJobCount method in the DeleteBoxRsp class returns the number of jobs that depend on the job that was deleted. This method has the following prototype:

```
int getDependentJobCount (void) const
```

jobDeleted

The jobDeleted method in the DeleteBoxRsp class returns TRUE if there is a job name to display when looping through the response set. It returns FALSE if no job name was found. This method has the following prototype:

```
bool jobDeleted () const
```

DeleteCalendarDate

DeleteCalendarDate deletes dates from a calendar. DeleteCalendarDate checks "delete" permissions for a specified calendar name. After the calendar dates are deleted, audit information is appropriately modified, and events are sent or deleted as necessary for jobs affected by this calendar change. This API is equivalent to the autocal_asc "Delete Calendar dates" option.

DeleteCalendarDate is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteCalendarDate API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteCalendarDateReq req;
    req.setName("mycalendar");
    req.appendDate( "12/08/2006 12:01" );
    DeleteCalendarDateRsp rsp;
    req.execute(api, rsp);
    std::cout << "Date from \"mycalendar\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteCalendarDateReq

extends
 ApiRequest
header
 AsPublicSDK.h

appendDate

The appendDate method in the DeleteCalendarDateReq class adds a date to the list of dates to delete from a standard calendar. This method has the following prototype:

```
void appendDate (const std::string &strDate)  
strDate  
    Specifies the dates to delete, using the standard format: MM/DD/YYYY HH:MM.
```

execute

The execute method in the DeleteCalendarDateReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteCalendarDateRsp &rsp)  
api  
    Specifies an AsApi object.  
rsp  
    Specifies a DeleteCalendarDateRsp object.
```

setName

The setName method in the DeleteCalendarDateReq class specifies the name of the standard calendar whose dates to delete. This method has the following prototype:

```
void setName (const std::string &strCalendarName)  
strCalendarName  
    Specifies the standard calendar name whose dates are to be deleted.
```

DeleteCalendarDateRsp

extends

ApiResponse

header

AsPublicSDK.h

getBadDates

The getBadDates method in the DeleteCalendarDateRsp class returns a list of dates that were not deleted. If ApiResponse::status returns RS_OBJECT_DOESNT_EXIST, this method returns a list of dates in the standard format of MM/DD/YYYY HH:MM. This method has the following prototype:

```
std::vector< std::string > getBadDates () const
```

DeleteCycleCalendarDays

DeleteCycleCalendarDays deletes a cycle calendar from the CA Workload Automation AE database. DeleteCycleCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteCycleCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteCycleCalendarDaysReq req;
    req.setName("MyCycCal");
    DeleteCycleCalendarDaysRsp rsp;
    req.execute(api, rsp);
    std::cout << "Cycle Calendar \"MyCycCal\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteCycleCalendarDaysReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the DeleteCycleCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteCycleCalendarDaysRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a DeleteCycleCalendarDaysRsp object

setName

The setName method in the DeleteCycleCalendarDaysReq class sets the name of the cycle calendar to delete. This method has the following prototype:

```
void setName (const std::string &strCalName)
```

strCalName

Specifies the name of the cycle calendar to delete.

DeleteEntireCalendarDate

DeleteEntireCalendarDate removes a standard calendar from the CA Workload Automation AE database. DeleteEntireCalendarDate is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteEntireCalendarDate API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteEntireCalendarDateReq req;
    req.setName("MyCalendar");
    DeleteEntireCalendarDateRsp rsp;
    req.execute(api, rsp);
    std::cout << "Calendar \"MyCalendar\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteEntireCalendarDateReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the DeleteEntireCalendarDateReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteEntireCalendarDateRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a DeleteEntireCalendarDateRsp object.

setName

The setName method in the DeleteEntireCalendarDateReq class specifies the name of the standard calendar to delete. This method has the following prototype:

```
void setName (const std::string &strCalendarName)
```

DeleteExtendedCalendarDays

DeleteExtendedCalendarDays deletes an extended calendar from the CA Workload Automation AE database. DeleteExtendedCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteExtendedCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteExtendedCalendarDaysReq req;
    req.setName("MyExtCal");
    DeleteExtendedCalendarDaysRsp rsp;
    req.execute(api, rsp);
    std::cout << "Extended Calendar \"MyExtCal\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteExtendedCalendarDaysReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the DeleteExtendedCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteExtendedCalendarDaysRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a DeleteExtendedCalendarDaysRsp object.

setName

The setName method in the DeleteExtendedCalendarDaysReq class sets the name of the extended calendar to delete. This method has the following prototype:

```
void setName (const std::string &strCalName)
```

strCalName

Specifies the name of the extended calendar to delete.

DeleteExternalInstance

DeleteExternalInstance deletes external instances from the CA Workload Automation AE database. DeleteExternalInstance is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteExternalInstance API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteExternalInstanceReq req;
    req.setName("EXT");
    DeleteExternalInstanceRsp rsp;
    req.execute(api, rsp);
    std::cout << "External instance \"EXT\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteExternalInstanceReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the DeleteExternalInstanceReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteExternalInstanceRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a DeleteExternalInstanceRsp object.

setName

The setName method in the DeleteExternalInstanceReq class sets the name of the external instance to delete. This method has the following prototype:

```
void setName (const std::string &name)
```

name

Specifies the name of the external instance to delete.

DeleteGbloc

DeleteGbloc is used to delete a global blob from the CA Workload Automation AE database. DeleteGbloc is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteGbloc API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteGblocReq req;
    req.setName("myglob");
    DeleteGblocRsp rsp;
    req.execute(api, rsp);
    std::cout << "Glob \"myglob\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteGblobReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the DeleteGblobReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, DeleteGblobRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a DeleteGblobRsp object.

setName

The setName method in the DeleteGblobReq class sets the name of the global BLOB to delete. This method has the following prototype:

`void setName (const std::string &strGblobName)`

strGblobName

Specifies the name of the global BLOB to delete.

setVerify

The setVerify method in the DeleteGblobReq class lets you run DeleteGblob in test mode. This method has the following prototype:

`void setVerify (bool bVerify)`

bVerify

Specifies the mode in which to run the job. Possible values are:

true

Indicates that DeleteGblob should run in test mode.

false

Perform the DeleteGblob operation.

DeleteJblob

DeleteJblob is used to delete a job blob from the CA Workload Automation AE database. DeleteJblob is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteJblob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteJblobReq req;
    req.setName("myblob");
    DeleteJblobRsp rsp;
    req.execute(api, rsp);
    std::cout << "Blob \"myblob\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteJblobReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the DeleteJblobReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteJblobRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a DeleteJblobRsp object.

setDeleteLatestInputBlob

The setDeleteLatestInputBlob method in the DeleteJblobReq class specifies whether to delete the latest version of the input BLOB or the STDOUT and STDERR BLOBS. This method has the following prototype:

```
void setDeleteLatestInputBlob (bool bDeleteLatestInputBlob)
```

bDeleteLatestInputBlob

Specifies the BLOB to delete. Possible values are:

true

Deletes the latest version of the input BLOB.

false

Deletes both the STDOUT and STDERR BLOBS.

setName

The setName method in the DeleteJblobReq class specifies the job name to which the BLOB is tied. This method has the following prototype:

```
void setName (const std::string &strJobName)
```

strJobName

Specifies the job name to which the BLOB is tied.

setVerify

The `setVerify` method in the `DeleteJblobReq` class lets you run `DeleteJblob` in test mode. This method has the following prototype:

```
void setVerify (bool bVerify)
```

bVerify

Indicates that `DeleteJblob` should run in test mode. Possible values are:

true

Do not delete the Jblob (test mode).

false

Delete the Jblob.

DeleteJblobRsp

extends

ApiResponse

header

AsPublicSDK.h

getNumberOfVersions

The `getNumberOfVersions` method in the `DeleteJblobRsp` class returns the number of input blob versions. This is valid only when `bDeleteLatestInputBlob` is set to TRUE in the request. This method has the following prototype:

```
int getNumberOfVersions () const
```

DeleteJob

DeleteJob deletes a specified job. It does not delete dependent jobs because they will be orphaned. DeleteJob verifies that the caller has delete permission for the specified job before deleting it.

When requested, the DeleteJob API also performs the following:

- Returns a list of jobs that depend on the job being deleted.
- Fails the delete when the job is in the ACTIVATED, STARTING, or RUNNING state. With jil, this is controlled by the RESTRICT_DELETE_JOB environment variable.
- Fails the delete when a job still has dependent jobs. With jil, this is controlled by the RESTRICT_DELETE_DEPENDENT_JOB environment variable.

DeleteJob is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteJob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteJobReq req;
    req.setName("myjob");
    DeleteJobRsp rsp;
    req.execute(api, rsp);
    std::cout << "Job \"myjob\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
```

```
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    api.close();
```

DeleteJobReq

extends

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the DeleteJobReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, DeleteJobRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a DeleteJobRsp object.

setName

The setName method in the DeleteJobReq class specifies the name of the job to be deleted. This method has the following prototype:

```
void setName (const std::string &strJobName)
```

setRestrictDeleteDependentJob

The setRestrictDeleteDependentJob method in the DeleteJobReq class indicates that the DeleteJob should abort if a job is encountered that still has dependent jobs. In JIL, this is controlled by the RESTRICT_DELETE_DEPENDENT_JOB environment variable. This method has the following prototype:

```
void setRestrictDeleteDependentJob (bool bRestrictDeleteDependentJob)  
bRestrictDeleteDependentJob
```

Specifies whether the API should fail if the job being deleted has dependent jobs.
Possible values are:

TRUE

Fail if a job is encountered that has dependent jobs.

FALSE

Do not fail DeleteJob if a job is encountered that has dependents.

setRestrictDeleteJob

The setRestrictDeleteJob method in the DeleteJobReq class forces the DeleteJob to abort if the job being deleted is in an ACTIVATED, STARTING or RUNNING state. In JIL, this is controlled by the RESTRICT_DELETE_JOB environment variable. This method has the following prototype:

```
void setRestrictDeleteJob (bool bRestrictDeleteJob)  
bRestrictDeleteJob
```

Specifies whether the API should fail if the job being deleted is in ACTIVATED, STARTING, or RUNNING state. Possible values are:

TRUE

Fail the DeleteJob if the job is in an ACTIVATED, STARTING, or RUNNING state.

FALSE

Do not fail the DeleteJob.

setVerify

The `setVerify` method in the `DeleteJobReq` class lets you run `DeleteJob` in test mode. This helps you fetch the list of jobs dependent on the job being deleted. This method has the following prototype:

```
void setVerify (bool bVerify)
```

bVerify

Indicates that `DeleteJob` should run in test mode. Possible values are:

true

Runs in test mode and retrieves the list of jobs dependent on the job being deleted.

false

Deletes the job.

DeleteJobRsp

extends

ApiResponse

header

AsPublicSDK.h

getDependentJob

The `getDependentJob` method in the `DeleteJobRsp` class returns a job that is dependent on the deleted job. Call `getDependentJobCount` to obtain the number of dependent jobs, and then call `getDependentJob` to retrieve the names of dependent jobs. This method has the following prototype:

```
std::string getDependentJob ( int i ) const
```

i

Specifies an index into the dependent job list.

getDependentJobCount

The `getDependentJobCount` method in the `DeleteJobRsp` class returns the number of jobs that depend on the job that was deleted. This method has the following prototype:

```
int getDependentJobCount (void) const
```

DeleteMachine

DeleteMachine is used to delete a machine definition from the CA Workload Automation AE database. Use this API to delete real machines, entire virtual machines, or component machines contained in a virtual machine.

DeleteMachine is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteMachine API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteMachineReq req;
    req.setName("MyMachine");
    DeleteMachineRsp rsp;
    req.execute(api, rsp);
    std::cout << "Machine \"MyMachine\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteMachineReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the DeleteMachineReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, DeleteMachineRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a DeleteMachineRsp object.

setName

The setName method in the DeleteMachineReq class specifies the name of the real or virtual machine to delete. This method has the following prototype:

`void setName (const std::string &strMachineName)`

setName

This version of the setName method specifies the name of a virtual machine with specific component machines to delete. This method has the following prototype:

`void setName (const std::string &strMachineName, const std::vector< std::string > componentMachines)`

strMachineName

Specifies the name of the virtual machine.

componentMachines

Specifies the names of component machines to delete from the specified virtual machine.

DeleteMachineRsp

extends
ApiResponse
header
AsPublicSDK.h

getDeleteReturnCode

The getDeleteReturnCode method in the DeleteMachineRsp class returns a Delete return code. Machine names are returned when performing a delete of component names. The return codes in this list have corresponding component names returned so that you can determine individual delete return codes. This method has the following prototype:

```
int getDeleteReturnCode (int i) const
```

i

Specifies an index into the list of returned delete return codes.

getDeleteReturnCodeCount

The getDeleteReturnCodeCount method in the DeleteMachineRsp class returns the number of return codes from the delete request. This method has the following prototype:

```
int getDeleteReturnCodeCount (void) const
```

getMachineName

The getMachineName method in the DeleteMachineRsp class returns the names of the component machines that have been deleted from a virtual machine. The machines in this list have corresponding return codes returned in the return code list. This method has the following prototype:

```
std::string getMachineName (int i) const
```

i

Specifies an index into the list of returned component machine names.

getMachineNameCount

The getMachineNameCount method in the DeleteMachineRsp class returns the number of machine names returned from the delete component machine request. This method has the following prototype:

```
int getMachineNameCount (void) const
```

DeleteResource

DeleteResource is used to delete a real or virtual resource from the CA Workload Automation AE database. DeleteResource is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the DeleteResource API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    DeleteResourceReq req;
    req.setName("MyResource");
    DeleteResourceRsp rsp;
    req.execute(api, rsp);
    std::cout << "Resource \"MyResource\" has been deleted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

DeleteResourceReq

extends
 ApiRequest
header
 AsPublicSDK.h

setName

The setName method in the DeleteResourceReq class specifies the name of the virtual resource to be deleted. This method has the following prototype:

setName(std::string *str resourceName*)

str resourceName

Specifies the name of the resource to be deleted.

setMachine

The setMachine method in the DeleteResourceReq class specifies the machine to be deleted. This method has the following prototype:

setMachine(const std::string & *str Machine*)

str Machine

Specifies the ID of the machine to be deleted.

execute

The execute method in the DeleteResourceReq class performs the API call. This method has the following prototype:

execute(AsApi & *api*, DeleteResourceRsp & *rsp*)

api

Specifies an AsApi object.

rsp

Specifies a DeleteResourceRsp object

DeleteResourceRsp

extends
ApiResponse
header
AsPublicSDK.h

EnvironmentVariableItem

The EnvironmentVariableItem class is the container class for specifying environment variables in job definitions.

Example: Sample Code for Executing the EnvironmentVariableItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string)"myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_CMD);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_COMMAND, (std::string) "mycommand");
    EnvironmentVariableItem ev;
    ev.set(KW_ENVVARS, (std::string) "var1=value1");
    ev.set(KW_ENVVARS, (std::string) "var2=value2");
    ev.set(KW_ENVVARS, (std::string) "var3=value3");
    req.add(&ev);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_COMMAND, KW_ENVVARS};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) / 
sizeof(int) );
```

```
// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
        rset.next(rsp);
        for (int i=0; i < rsp.getAttributes(); i++) {

            // Display keywords that are of type string.
            if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                    rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }

            // Display job items.
            else if( rsp.getAttributeType(i) == AtJobItem ) {
                autosys::JobItem myJobitem = rsp.getJobItem(i);
                std::vector<int> keywords =
                    myJobitem.getVectorOfKeywordConstants();
                for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                    // Only displaying jobitem keywords that are of type string.
                    std::cout << "Jobitem string attribute [" <<
                    rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                        myJobitem.getString( (int)keywords[j] ) << "]" <<
                    std::endl;
                }
            }
        }
    }
}

catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```

FailedToSetCreds

The FailedToSetCreds class is used to indicate that the internal client code tried to set the request credentials based on the currently logged in user (as opposed to using `ApiRequest::setUser()`).

FetchGbloc

FetchGbloc is used to retrieve global BLOBs from the CA Workload Automation AE database. FetchGbloc is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the FetchGbloc API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    FetchGblocReq req;
    req.setName("mygblob");
    FetchGblocRsp rsp;
    req.execute(api, rsp);
    std::cout << "Create User: " << rsp.getCreateUser() << std::endl;
    std::cout << "Create Time: " << rsp.getCreateTime() << std::endl;
    std::cout << "Gbloc size : " << rsp.getGblocSize() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

FetchGblobReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the FetchGblobReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, FetchGblobRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a FetchGblobRsp object.

setName

The setName method in the FetchGblobReq class specifies the name of the global blob to retrieve. This method has the following prototype:

```
void setName (const std::string &strGblobName)
```

strGblobName

Specifies the name of the global blob.

FetchGblobRsp

extends
 ApiResponse
header
 AsPublicSDK.h

getCreateTime

The getCreateTime method in the FetchGblobRsp class returns the creation time. This method has the following prototype:

```
std::string getCreateTime ()
```

getCreateUser

The `getCreateUser` method in the `FetchGblocRsp` class returns the name of the user who created the global blob. This method has the following prototype:

```
std::string getCreateUser ()
```

getGbloc

The `getGbloc` method in the `FetchGblocRsp` class returns the fetched blob. This method has the following prototype:

```
void * getGbloc ()
```

getGblocSize

The `getGblocSize` method in the `FetchGblocRsp` class returns the length of the fetched blob. This method has the following prototype:

```
int getGblocSize ()
```

getModifyTime

The `getModifyTime` method in the `FetchGblocRsp` class returns the modify time. This method has the following prototype:

```
std::string getModifyTime ()
```

getModifyUser

The `getModifyUser` method in the `FetchGblocRsp` class returns the name of the user who modified the global blob. This method has the following prototype:

```
std::string getModifyUser ()
```

FetchJblob

FetchJblob retrieves a BLOB that is tied to a CA Workload Automation AE job.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the FetchJblob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    FetchJblobReq req;
    req.setName("myblob");
    FetchJblobRsp rsp;
    req.execute(api, rsp);
    std::cout << "Jblob size : " << rsp.getJblobSize()<< std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

FetchJblobReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the FetchJblobReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, FetchJblobRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a FetchJblobRsp object.

setName

The setName method in the FetchJblobReq class specifies the job name to which the blob is tied. This method has the following prototype:

`void setName (const std::string &strJobName)`

strJobName

Specifies the existing job name to which the blob is tied.

setType

The setType method in the FetchJblobReq class specifies the blob type. This method has the following prototype:

`void setType (eJobBlobType eJblobType)`

eJblobType

Specifies the type of blob to retrieve. Possible values are:

- JbStdin
- JbStdout
- JbStderr

setVersion

The setVersion method in the FetchJblobReq class specifies the version of the input blob to fetch. This method has the following prototype:

```
void setVersion (const int iVersion)
```

iVersion

Specifies the version number of the input blob to fetch.

Note: This parameter is ignored if the blob type is not equal to JbStdin.

FetchJblobRsp

extends

ApiResponse

header

AsPublicSDK.h

getJblob

The getJblob method in the FetchJblobRsp class returns the fetched blob. This method has the following prototype:

```
void * getJblob ()
```

getJblobSize

The getJblobSize method in the FetchJblobRsp class returns the length of the fetched blob. This method has the following prototype:

```
int getJblobSize ()
```

FinderParameterItem

The FinderParameterItem class is the container class for specifying finder parameter arguments in job definitions.

Example: Sample Code for Executing the FinderParameterItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_ENTRYBEAN);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_BEAN_NAME, (std::string) "myBean");
    req.set(KW_INITIAL_CONTEXT_FACTORY, (std::string) "myContext");
    req.set(KW_METHOD_NAME, (std::string) "myMethod");
    req.set(KW_PROVIDER_URL, (std::string) "myurl");
    req.set(KW_OPERATION_TYPE, EN_JMSOPERATION_UPDATE);
    FinderParameterItem fp;
    fp.set(KW_FINDER_PARAMETER, (std::string) "var1=value1");
    fp.set(KW_FINDER_PARAMETER, (std::string) "var2=value2");
    fp.set(KW_FINDER_PARAMETER, (std::string) "var3=value3");
    req.add(&fp);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_BEAN_NAME, KW_FINDER_PARAMETER};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
```

```
rset.next(rsp);
for (int i=0; i < rsp.getAttributes(); i++) {

    // Display keywords that are of type string.
    if( rsp.getAttributeType(i) == AtString ) {
        std::cout << "String attribute [" <<
        rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
            rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
    }

    // Display job items.
    else if( rsp.getAttributeType(i) == AtJobItem ) {
        autosys::JobItem myJobitem = rsp.getJobItem(i);
        std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
        for( int j=0; j < myJobitem.getAttributes(); j++ ) {
            // Only displaying jobitem keywords that are of type string.
            std::cout << "Jobitem string attribute [" <<
            rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                myJobitem.getString( (int)keywords[j] ) << "]" <<
            std::endl;
        }
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```

GetAdapterJobStats

GetAdapterJobStats retrieves the statuses of the product's adapter jobs. This API is used by the autostataad command.

GetAdapterJobStats verifies list security for the AUTOSTAT security resource. If that passes, the API does not verify security for each job. Otherwise, the API verifies read access for each job. If read access verification fails, the API returns an empty response (zero-length job name).

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetAdapterJobStats API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetAdapterJobStatsReq req;
    req.setNamePattern("ALL");
    GetAdapterJobStatsRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetAdapterJobStatsRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo())
```

```
        std::cout << rsp.info() << std::endl;
    if (rsp.haveAccess()) {
        std::cout << "Adapter job name : " << rsp.getAdapterJobName() << std::endl;
        std::cout << "Adapter job status: " << rsp.getAdapterJobStatus() <<
std::endl;
        std::cout << "Status time      : " << rsp.getStatusTime() << std::endl;
        std::cout << "-----" << std::endl;
    }
}
}
}
}
catch(ApiTimeout&)
{
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&)
{
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&)
{
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&)
{
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&)
{
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&)
{
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&)
{
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&)
{
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&)
{
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&)
{
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&)
{
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&)
{
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&)
{
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
```

```
        catch(NoResponseAvailable&) {
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
        catch(ResponseDequeueFailure&) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
    api.close();
```

GetAdapterJobStatsReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetAdapterJobStatsReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetAdapterJobStatsRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetAdapterJobStatsRspSet object.

setNamePattern

The setNamePattern method in the GetAdapterJobStatsReq class specifies the name or name pattern of an adapter job or jobs to retrieve. This method has the following prototype:

```
void setNamePattern (const std::string &name)
```

name

Specifies the fully-qualified name or pattern of the adapter jobs to retrieve.

GetAdapterJobStatsRsp

extends
ApiResponse
header
AsPublicSDK.h

getAdapterJobName

The getAdapterJobName method in the GetAdapterJobStatsRsp class returns the job name. This method has the following prototype:

```
std::string getAdapterJobName () const
```

getAdapterJobStatus

The getAdapterJobStatus method in the GetAdapterJobStatsRsp class returns the job status. This method has the following prototype:

```
std::string getAdapterJobStatus () const
```

getStatusTime

The getStatusTime method in the GetAdapterJobStatsRsp class returns the status time. This method has the following prototype:

```
int getStatusTime () const
```

haveAccess

The haveAccess method in the GetAdapterJobStatsRsp class returns TRUE if you have read access, otherwise returns FALSE. This method has the following prototype:

```
bool haveAccess () const
```

GetAlarmsWithFilter

GetAlarmsWithFilter retrieves a set of alarm attributes based on the provided filter object. GetAlarmsWithFilter creates an SQL query based on the filter object and binds to and returns the variables specified in the Request. Security is checked and jobs that the requesting user does not have read permission to will not be returned.

GetAlarmsWithFilter is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetAlarmsWithFilter API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetAlarmsWithFilterReq req;

    // Set up the attributes fields (see eAlarmAttribute in asapi_cmn.h for
    attributes).
#define NumAttr 8
    int attr[NumAttr] = {
        AaAlarmType,
        AaEoid,
        AaEventComment,
```

```

        AaJobName,
        AaModUser,
        AaResponse,
        AaState,
        AaJoid
    );
    std::vector<int> attributes(attr, attr+NumAttr);
    req.setAttributes(attributes);

    FilterAnd* al = new FilterAnd;
    al->add(new FilterString(autosys::A_JobName, "%"));
    req.setFilter(al);

    GetAlarmsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetAlarmsWithFilterRsp rsp;
        rset.next(rsp);
        int iNumAttributes = rsp.getAttributes();
        for (int i=0; i < iNumAttributes; i++) {
            if( rsp.getAttributeType(i) == AtInt ) {
                std::cout << "Int attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getInt(rsp.getAttributeName(i)) << "]" << std::endl;
            }
            else if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }
        }
    }
    catch(ApiTimeout&) {
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(AsExternalValidationException&) {
        std::cerr << "AsExternalValidationException thrown." << std::endl;
    }
    catch(AsFieldValidationException&) {
        std::cerr << "AsFieldValidationException thrown." << std::endl;
    }
    catch(AsGeneralErrorException&) {
        std::cerr << "AsGeneralErrorException thrown." << std::endl;
    }
    catch(AsObjectDoesNotExistException&) {
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&) {

```

```
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&) {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&) {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&) {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&) {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&) {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

GetAlarmsWithFilterReq

extends

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the GetAlarmsWithFilterReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetAlarmsWithFilterRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetAlarmsWithFilterRspSet object.

setAttributes

The setAttributes method in the GetAlarmsWithFilterReq class sets the attributes on which to report. This method has the following prototype:

```
void setAttributes (std::vector< int > attributes)
```

attributes

Specifies an integer array containing one or more of the following attributes:

- AaAlarmTime
- AaAlarmType
- AaEoid
- AaEventComment
- AaJobName
- AaModUser
- AaResponse
- AaState
- AaStateTime
- AaJoid

setFilter

The setFilter method in the GetAlarmsWithFilterReq class specifies a pointer to a filter object. This method has the following prototype:

```
void setFilter (Filter *pFilter)  
pFilter
```

Specifies the pointer to a filter object. If setFilter is not called or called with a NULL value, all alarms are returned.

GetAlarmsWithFilterRsp

extends

ApiResponse

header

AsPublicSDK.h

getAttributeName

The getAttributeName method in the GetAlarmsWithFilterRsp class retrieves an attribute name. This method has the following prototype:

```
eAlarmAttribute getAttributeName (int index) const  
index
```

Specifies the attribute name to retrieve. Possible eAlarmAttribute enumerated values returned are:

- AaAlarmTime
- AaAlarmType
- AaEoid
- AaEventComment
- AaJobName
- AaModUser
- AaResponse
- AaState
- AaStateTime
- AaJoid

getAttributeNameString

The `getAttributeNameString` method in the `GetAlarmsWithFilterRsp` class returns the string version of the attribute name. This method has the following prototype:

```
std::string getAttributeNameString (eAlarmAttribute i) const
```

Possible `eAlarmAttribute` enumerated values are:

- `AaAlarmTime`
- `AaAlarmType`
- `AaEoid`
- `AaEventComment`
- `AaJobName`
- `AaModUser`
- `AaResponse`
- `AaState`
- `AaStateTime`
- `AaJoid`

getAttributes

The `getAttributes` method in the `GetAlarmsWithFilterRsp` class returns the number of attributes to process. This method has the following prototype:

```
int getAttributes () const
```

getAttributeType

The `getAttributeType` method in the `GetAlarmsWithFilterRsp` class returns an attribute type. This method has the following prototype:

```
eAttributeType getAttributeType (int index) const
```

index

Specifies the type of attribute to retrieve. Possible `eAttributeType` enumerated values returned are:

- `AtInt`
- `AtString`
- `AtDate`

getAttributeTypeString

The `getAttributeTypeString` method in the `GetAlarmsWithFilterRsp` class returns the string version of the attribute type. This method has the following prototype:

```
std::string getAttributeTypeString (eAttributeType i) const
```

Possible `eAttributeType` enumerated values are:

- `AtInt`
- `AtString`
- `AtDate`

getDate

The `getDate` method in the `GetAlarmsWithFilterRsp` class returns the date value for the given alarm attribute. This method has the following prototype:

```
time_t getDate (int index) const
```

index

Specifies the type of attribute to retrieve. Possible values are:

- `AaAlarmTime`
- `AaStateTime`

getInt

The `getInt` method in the `GetAlarmsWithFilterRsp` class returns the int value for the given alarm attribute. This method has the following prototype:

```
int getInt (int index) const
```

index

Specifies the type of attribute to retrieve. Possible values are:

- `AaAlarmType`
- `AaState`

getString

The `getString` method in the `GetAlarmsWithFilterRsp` class returns the string value for the given alarm attribute. This method has the following prototype:

```
std::string getString (int index) const  
index
```

Specifies the type of attribute to retrieve. Possible values are:

- `AaEoid`
- `AaEventComment`
- `AaJobName`
- `AaModUser`
- `AaResponse`

GetCalendarDays

`GetCalendarDays` retrieves all the dates in the requested standard calendar. `GetCalendarDays` checks to see if the user has read access to the requested calendar and then returns each date.

`GetCalendarDays` is a CAT2 API.

Executing this API can result in the following exceptions:

- `ApiExceptionWithInfo`
- `ApiTimeout`
- `AsExternalValidationException`
- `AsFieldValidationException`
- `AsGeneralErrorException`
- `AsObjectDoesNotExistException`
- `AsSecurityException`
- `AsUnknownErrorException`
- `Cat2StartFailure`
- `CredsNotAuthorized`
- `FailedToSetCreds`
- `InternalResponseError`
- `InternalUnexpected`

- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetCalendarDaysReq req;
    req.setName("MyCalendar");
    GetCalendarDaysRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetCalendarDaysRsp rsp;
        rset.next(rsp);
        std::cout << "Date: " << rsp.getDate() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
```

```
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
catch(ResponseDequeueFailure&) {
    std::cerr << "ResponseDequeueFailure thrown." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
api.close();
```

GetCalendarDaysReq

extends

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the GetCalendarDaysReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, GetCalendarDaysRspSet &rset)`

api

Specifies an AsApi object.

rset

Specifies a GetCalendarDaysRspSet object.

setName

The setName method in the GetCalendarDaysReq class specifies the name of the standard calendar to retrieve. This method has the following prototype:

`void setName (const std::string &name)`

name

Specifies the fully-qualified name of the standard calendar to retrieve. This API call does not accept wildcards.

GetCalendarDaysRsp

extends

ApiResponse

header

AsPublicSDK.h

getDate

The getDate method in the GetCalendarDaysRsp class returns a date from the requested calendar. The date is in the format MM/DD/YYYY HH:MM, regardless of locale. This method has the following prototype:

`std::string getDate () const`

GetCalendarNames

GetCalendarNames returns a list of all defined calendar names from the CA Workload Automation AE database. GetCalendarNames is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetCalendarNames API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetCalendarNamesReq req;
    GetCalendarNamesRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetCalendarNamesRsp rsp;
        rset.next(rsp);
        std::cout << "Calendar Name: " << rsp.getCalendarName() << std::endl;
    }
}
```

```
        catch(ApiTimeout&) {
            std::cerr << "ApiTimeout thrown." << std::endl;
        }
        catch(AsExternalValidationException&) {
            std::cerr << "AsExternalValidationException thrown." << std::endl;
        }
        catch(AsFieldValidationException&) {
            std::cerr << "AsFieldValidationException thrown." << std::endl;
        }
        catch(AsGeneralErrorException&) {
            std::cerr << "AsGeneralErrorException thrown." << std::endl;
        }
        catch(AsObjectDoesNotExistException&) {
            std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
        }
        catch(AsSecurityException&) {
            std::cerr << "AsSecurityException thrown." << std::endl;
        }
        catch(AsUnknownErrorException&) {
            std::cerr << "AsUnknownErrorException thrown." << std::endl;
        }
        catch(Cat2StartFailure&) {
            std::cerr << "AsUnknownErrorException thrown." << std::endl;
        }
        catch(CredsNotAuthorized&) {
            std::cerr << "CredsNotAuthorized thrown." << std::endl;
        }
        catch(FailedToSetCreds&) {
            std::cerr << "FailedToSetCreds thrown." << std::endl;
        }
        catch(InternalResponseError&) {
            std::cerr << "InternalResponseError thrown." << std::endl;
        }
        catch(InternalUnexpected&) {
            std::cerr << "InternalUnexpected thrown." << std::endl;
        }
        catch(MissingFinalResponse&) {
            std::cerr << "MissingFinalResponse thrown." << std::endl;
        }
        catch(NoResponseAvailable&) {
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
    }
```

```

        catch(ResponseDequeueFailure& ) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
        api.close();
    }
}

```

GetCalendarNamesReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetCalendarNamesReq class performs the API call. This method has the following prototype:

void execute (AsApi &*api*, GetCalendarNamesRspSet &*rset*)

api

Specifies an AsApi object.

rset

Specifies a GetCalendarNamesRspSet object.

GetCalendarNamesRsp

extends

 ApiResponse

header

 AsPublicSDK.h

getCalendarName

The getCalendarName method in the GetCalendarNamesRsp class returns the calendar name. This method has the following prototype:

std::string getCalendarName () const

GetCycleCalendarDays

GetCycleCalendarDays retrieves all the dates in a requested cycle calendar. GetCycleCalendarDays checks to see if the user has read access to the requested calendar and then returns each date.

GetCycleCalendarDays is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetCycleCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetCycleCalendarDaysReq req;
    req.setName("MyCycleCalendar");
    GetCycleCalendarDaysRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetCycleCalendarDaysRsp rsp;
        rset.next(rsp);
        std::cout << "Cycle Calendar Name: " << rsp.getCalName() << std::endl;
        std::cout << "Start Date: " << rsp.getStartDate() << std::endl;
        std::cout << "End Date : " << rsp.getEndDate() << std::endl;
```

```
        }
    }
    catch(ApiTimeout&)
    {
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(AsExternalValidationException&)
    {
        std::cerr << "AsExternalValidationException thrown." << std::endl;
    }
    catch(AsFieldValidationException&)
    {
        std::cerr << "AsFieldValidationException thrown." << std::endl;
    }
    catch(AsGeneralErrorException&)
    {
        std::cerr << "AsGeneralErrorException thrown." << std::endl;
    }
    catch(AsObjectDoesNotExistException&)
    {
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&)
    {
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&)
    {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&)
    {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&)
    {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&)
    {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&)
    {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&)
    {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&)
    {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&)
    {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&)
    {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e)
    {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    }
}
```

```
    std::cerr << e.what() << std::endl;
}
api.close();
```

GetCycleCalendarDaysReq

extends

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the GetCycleCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute(AsApi& api, GetCycleCalendarDaysRspSet& rset);
```

api

Specifies an AsApi object.

rset

Specifies a GetCycleCalendarDaysRspSet object.

Returns

- RS_SUCCESS
- RS_FAILURE
- RS_ACCESS_DENIED

The execute status is returned in the inherited response method status().

setName

The setName method in the GetCycleCalendarDaysReq class sets the name of cycle calendar to retrieve. This method has the following prototype:

```
void setName(const std::string& name);
```

name

Specifies the cycle calendar name.

GetCycleCalendarDaysRsp

extends
ApiResponse
header
AsPublicSDK.h

getCycCalName

The getCycCalName method in the GetCycleCalendarDaysRsp class returns the cycle calendar name. This method has the following prototype:

```
std::string  getCycCalName () const
```

getEndDate

The getEndDate method in the GetCycleCalendarDaysRsp class returns the end date of the cycle. This method has the following prototype:

```
std::string  getEndDate () const
```

getStartDate

The getStartDate method in the GetCycleCalendarDaysRsp class returns the start date of the cycle. This method has the following prototype:

```
std::string  getStartDate () const
```

GetExtendedCalendarDays

GetExtendedCalendarDays retrieves the days that are defined in an extended calendar. GetExtendedCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the GetExtendedCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetExtendedCalendarDaysReq req;
    req.setName("ExtCalTest");
    GetExtendedCalendarDaysRsp rsp;
    req.execute(api, rsp);
    if( rsp.status() == RS_SUCCESS )
    {
        std::cout << "Calendar Name: " << rsp.getCalendarName() << std::endl;
        std::cout << "Workdays: " << rsp.getWorkdays() << std::endl;
        std::cout << "Holiday Calendar: " << rsp.getHolidayCalendar() << endl;
        std::cout << "Cycle Cal Name: " << rsp.getCycleCalName() << std::endl;
        std::cout << "Date Con: " << rsp.getDateCon() << std::endl;
        std::cout << "Adjustment: " << rsp.getAdjustment() << std::endl;
        std::cout << "Holiday Action: " << rsp.getHolidayAction() << std::endl;
        std::cout << "Weekend Action: " << rsp.getWeekendAction() << std::endl;
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
```

```
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    api.close();
```

GetExtendedCalendarDaysReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetExtendedCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetExtendedCalendarDaysRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a GetExtendedCalendarDaysRsp object.

setName

The setName method in the GetExtendedCalendarDaysReq class specifies the name of the external calendar to retrieve. This method has the following prototype:

```
void setName (const std::string &name)
```

GetExtendedCalendarDaysRsp

extends
ApiResponse
header
AsPublicSDK.h

getAdjustment

The getAdjustment method in the GetExtendedCalendarDaysRsp class returns the adjustment for the specified condition. This method has the following prototype:

```
int getAdjustment () const
```

getCalendarName

The getCalendarName method in the GetExtendedCalendarDaysRsp class returns the name of the calendar. This method has the following prototype:

```
std::string getCalendarName () const
```

getCycleCalName

The getCycleCalName method in the GetExtendedCalendarDaysRsp class returns the cycle calendar name. This method has the following prototype:

```
std::string getCycleCalName () const
```

getDateCon

The getDateCon method in the GetExtendedCalendarDaysRsp class returns the condition used to decide holiday and cycle usage. This method has the following prototype:

```
std::string getDateCon () const
```

getHolidayAction

The getHolidayAction method in the GetExtendedCalendarDaysRsp class returns the action to take when the work falls on a holiday. This method has the following prototype:

```
eCalendarAction getHolidayAction () const
```

Valid holiday actions are:

- CaDefault
- CaOnly

- CaSchedule
- CaNextDay
- CaNextWorkDay
- CaPreviousWorkDay

getHolidayCalendar

The getHolidayCalendar method in the GetExtendedCalendarDaysRsp class returns the name of the holiday calendar. This method has the following prototype:

```
std::string getHolidayCalendar () const
```

getWeekendAction

The getWeekendAction method in the GetExtendedCalendarDaysRsp class returns the action to take when the work falls on a non-workday. This method has the following prototype:

```
eCalendarAction getWeekendAction () const
```

Valid non-workday actions are:

- CaDefault
- CaOnly
- CaNextDay
- CaNextWorkDay
- CaPreviousWorkDay

getWorkdays

The getWorkdays method in the GetExtendedCalendarDaysRsp class returns the workdays in the calendar. This method has the following prototype:

```
std::string getWorkdays () const
```

GetExternalInstances

GetExternalInstances retrieves instance information for all external instances that match the given pattern. GetExternalInstances is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetExternalInstances API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetExternalInstancesReq req;
    req.setInstancePattern("%"); // Retrieve ALL instances.
    GetExternalInstancesRspSet rset;
    req.execute(api, rset);
    // Loop thru response set and print instance IDs.
    while (rset.hasMore()) {
        GetExternalInstancesRsp rsp;
        rset.next(rsp);
        if(rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "\n\n Instance ID: " << rsp.instanceId() << std::endl;
    }
}
catch(ApiTimeout&) {
```

```
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(AsExternalValidationException&) {
        std::cerr << "AsExternalValidationException thrown." << std::endl;
    }
    catch(AsFieldValidationException&) {
        std::cerr << "AsFieldValidationException thrown." << std::endl;
    }
    catch(AsGeneralErrorException&) {
        std::cerr << "AsGeneralErrorException thrown." << std::endl;
    }
    catch(AsObjectDoesNotExistException&) {
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&) {
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&) {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&) {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&) {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&) {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&) {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

GetExternalInstancesReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetExternalInstancesReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetExternalInstancesRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetExternalInstancesRspSet object.

setInstancePattern

The setInstancePattern method in the GetExternalInstancesReq class specifies an instance name or pattern. This method has the following prototype:

```
void setInstancePattern (const std::string &name)
```

name

Specifies the fully-qualified name or pattern of instances to retrieve. For example, use P% to retrieve instances beginning with the letter P. Use % to retrieve all instances.

GetExternalInstancesRsp

extends
ApiResponse
header
AsPublicSDK.h

cxnInfo

The cxnInfo method in the GetExternalInstancesRsp class returns the instance connect information. This method has the following prototype:

```
std::string cxnInfo () const
```

getConnectInfo

The getConnectInfo method in the GetExternalInstancesRsp class returns the external machine or connection information. This method has the following prototype:

```
getConnectInfo() const
```

getCryptType

The getCryptType method in the GetExternalInstancesRsp class returns the encryption type for the external instance. This method has the following prototype:

```
getCryptType() const
```

This method returns one of the following encryption types:

- XI_None
- XI_Default
- XI_AES

getMachine

The getMachine method in the GetExternalInstancesRsp class returns the machine name for the external instance. This method has the following prototype:

```
getMachine() const
```

getManager

The getManager method in the GetExternalInstancesRsp class returns the Manager Alias. This method has the following prototype:

```
getManager() const
```

getPort

The `getPort` method in the `GetExternalInstancesRsp` class returns the external instance's port number. This method has the following prototype:

```
getPort() const
```

instanceId

The `instanceId` method in the `GetExternalInstancesRsp` class returns the instance ID. This method has the following prototype:

```
std::string instanceId () const
```

type

The `type` method in the `GetExternalInstancesRsp` class returns the instance type. This method has the following prototype:

```
eInstanceType type () const
```

Valid instance types are:

- `XI_AppServer`
- `XI_Connect`
- `XI_Unicenter`

GetGblobList

GetGblobList retrieves all the gblobs and the associated meta data. GetGblobList does not retrieve the actual gblob contents. GetGblobList is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetGblobList API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetGblobListReq req;
    req.setName("ALL");
    GetGblobListRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetGblobListRsp rsp;
        rset.next(rsp);
        if( rsp.hasInfo() ) std::cout << rsp.info() << std::endl;
        std::cout << "Gblob Name=" << rsp.getName() << std::endl;
        std::cout << "Create Time=" << rsp.getCreateTime() << std::endl;
        std::cout << "Modify Time=" << rsp.getModifyTime() << std::endl;
        std::cout << "Create User=" << rsp.getCreateUser() << std::endl;
        std::cout << "Modify User=" << rsp.getModifyUser() << std::endl;
```

```
        }
    }
    catch(ApiTimeout&)
    {
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(AsExternalValidationException&)
    {
        std::cerr << "AsExternalValidationException thrown." << std::endl;
    }
    catch(AsFieldValidationException&)
    {
        std::cerr << "AsFieldValidationException thrown." << std::endl;
    }
    catch(AsGeneralErrorException&)
    {
        std::cerr << "AsGeneralErrorException thrown." << std::endl;
    }
    catch(AsObjectDoesNotExistException&)
    {
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&)
    {
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&)
    {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&)
    {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&)
    {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&)
    {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&)
    {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&)
    {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&)
    {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&)
    {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&)
    {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e)
    {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    }
}
```

```
    std::cerr << e.what() << std::endl;
}
api.close();
```

GetGblobListReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetGblobListReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetGblobListRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetGblobListRspSet object.

setName

The setName method in the GetGblobListReq class specifies the name of the global blob. This method has the following prototype:

```
void setName (const std::string &strGblobName)
```

strGblobName

Specifies the name of the global blob or ALL to retrieve all blobs.

GetGblobListRsp

```
extends  
    ApiResponse  
header  
    AsPublicSDK.h
```

getCreateTime

The getCreateTime method in the GetGblobListReq class returns the creation time. This method has the following prototype:

```
std::string getCreateTime ()
```

getCreateUser

The getCreateUser method in the GetGblobListReq class returns the name of the user who created the global blob. This method has the following prototype:

```
std::string getCreateUser ()
```

getModifyTime

The getModifyTime method in the GetGblobListReq class returns the modify time. This method has the following prototype:

```
std::string getModifyTime ()
```

getModifyUser

The getModifyUser method in the GetGblobListReq class returns the name of the user who modified the global blob. This method has the following prototype:

```
std::string getModifyUser ()
```

getName

The getName method in the GetGblobListReq class returns the name of the global blob. This method has the following prototype:

```
std::string getName ()
```

GetGeneratedExtendedCalendarDays

GetGeneratedExtendedCalendarDays returns the dates that would be generated for the given extended calendar definition. GetGeneratedExtendedCalendarDays is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetGeneratedExtendedCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetGeneratedExtendedCalendarDaysReq req;
    req.setCalendarName("CalName");
    req.setWorkdays("X_____");
    req.setHolidayCalendar("holcal");
    req.setCycleCalName("cycal");
    req.setHolidayAction(CaNextDay);
    req.setWeekendAction(CaPreviousWorkDay);
    req.setAdjustment(1);
    req.setDateCon("DAILY");
    GetGeneratedExtendedCalendarDaysRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetGeneratedExtendedCalendarDaysRsp rsp;
```

```
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "getCalendarDate: " << rsp.getCalendarDate() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
catch(ResponseDequeueFailure&) {
    std::cerr << "ResponseDequeueFailure thrown." << std::endl;
}
```

```
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

GetGeneratedExtendedCalendarDaysReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetGeneratedExtendedCalendarDaysReq class performs the API call. This method has the following prototype:

void execute (AsApi &*api*, GetGeneratedExtendedCalendarDaysRspSet &*rset*)

api

Specifies an AsApi object.

rset

Specifies a GetGeneratedExtendedCalendarDaysRspSet object.

setAdjustment

The `setAdjustment` method in the `GetGeneratedExtendedCalendarDaysReq` class specifies the number of days before or after the condition setting. This method has the following prototype:

```
void setAdjustment (int iAdjustment) const  
iAdjustment
```

Specifies the negative or positive number defining the days before or after the condition setting.

setCalendarName

The `setCalendarName` method in the `GetGeneratedExtendedCalendarDaysReq` class specifies the calendar name to insert. This method has the following prototype:

```
void setCalendarName (const std::string &strCalendarName) const  
strCalendarName
```

Specifies the name of the calendar to insert.

setCycleCalName

The `setCycleCalName` method in the `GetGeneratedExtendedCalendarDaysReq` class sets the name of the cycle calendar. This method has the following prototype:

```
void setCycleCalName (const std::string &strCycleCalName) const  
strCycleCalName
```

Specifies the name of the cycle calendar to use.

setDateCon

The `setDateCon` method in the `GetGeneratedExtendedCalendarDaysReq` class sets the condition keyword string. This method has the following prototype:

```
void setDateCon (const std::string &strDateCon) const  
strDateCon
```

Specifies the condition keyword string.

setHolidayAction

The `setHolidayAction` method in the `GetGeneratedExtendedCalendarDaysReq` class specifies the action to take when the work falls on a holiday. This method has the following prototype:

```
void setHolidayAction (int iHolidayAction) const  
iHolidayAction
```

Specifies the action to take when the work falls on a holiday. Valid holiday actions are:

- `CaDefault`
- `CaOnly`
- `CaSchedule`
- `CaNextDay`
- `CaNextWorkDay`
- `CaPreviousWorkDay`

setHolidayCalendar

The `setHolidayCalendar` method in the `GetGeneratedExtendedCalendarDaysReq` class specifies the name of the holiday calendar to use. This method has the following prototype:

```
void setHolidayCalendar (const std::string &strHolidayCalendar) const  
strHolidayCalendar
```

Specifies the name of the holiday calendar.

setWeekendAction

The `setWeekendAction` method in the `GetGeneratedExtendedCalendarDaysReq` class specifies the action to take when the work falls on a non-workday. This method has the following prototype:

```
void setWeekendAction (int iWeekendAction) const
```

iWeekendAction

Specifies the action to take when the work falls on a non-workday. Valid weekend actions are:

- `CaDefault`
- `CaOnly`
- `CaNextDay`
- `CaNextWorkDay`
- `CaPreviousWorkDay`

setWorkdays

The `setWorkdays` method in the `GetGeneratedExtendedCalendarDaysReq` class sets the `workdays` string. This method has the following prototype:

```
void setWorkdays (const std::string &strWorkdays) const
```

strWorkdays

Specifies the requested workdays. For example:

- To select Monday through Friday, pass the string `XXXXX__` to `setWorkdays`.
- To select Monday, Wednesday and Friday, pass the string `X_X_X__` to `setWorkdays`.
- To select Monday and Friday, pass the string `X__X__` to `setWorkdays`.

GetGeneratedExtendedCalendarDaysRsp

extends
ApiResponse
header
AsPublicSDK.h

getCalendarDate

The getCalendarDate method in the GetGeneratedExtendedCalendarDaysRsp class returns the generated calendar date. This method has the following prototype:

```
std::string getCalendarDate () const
```

GetGlobalConditionsForJob

GetGlobalConditionsForJob retrieves global conditions for the specified job. This is a CAT2 API that returns one response for each global condition in its condition field. For each global returned, read permissions are checked. If denied, only the name and expected value are returned and getIsGlobalDetailAvailable returns FALSE and the secured fields will be empty strings. If granted, all global fields are returned and getIsGlobalDetailAvailable returns TRUE.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected

- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetGlobalConditionsForJob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetGlobalConditionsForJobReq req;
    req.setName("MyJob");
    GetGlobalConditionsForJobRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetGlobalConditionsForJobRsp rsp;
        rset.next(rsp);
        std::cout << "getName: " << rsp.getName() << std::endl;
        std::cout << "getExpectedValue: " << rsp.getExpectedValue() << std::endl;
        std::cout << "getOperator: " << rsp.getOperator() << std::endl;
        std::cout << "getDetailOwner: " << rsp.getDetailOwner() << std::endl;
        std::cout << "getDetailValue: " << rsp.getDetailValue() << std::endl;
        std::cout << "getDetailPermission: " << rsp.getDetailPermission() <<
std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
```

```

        catch(AsSecurityException&) {
            std::cerr << "AsSecurityException thrown." << std::endl;
        }
        catch(AsUnknownErrorException&) {
            std::cerr << "AsUnknownErrorException thrown." << std::endl;
        }
        catch(Cat2StartFailure&) {
            std::cerr << "AsUnknownErrorException thrown." << std::endl;
        }
        catch(CredsNotAuthorized&) {
            std::cerr << "CredsNotAuthorized thrown." << std::endl;
        }
        catch(FailedToSetCreds&) {
            std::cerr << "FailedToSetCreds thrown." << std::endl;
        }
        catch(InternalResponseError&) {
            std::cerr << "InternalResponseError thrown." << std::endl;
        }
        catch(InternalUnexpected&) {
            std::cerr << "InternalUnexpected thrown." << std::endl;
        }
        catch(MissingFinalResponse&) {
            std::cerr << "MissingFinalResponse thrown." << std::endl;
        }
        catch(NoResponseAvailable&) {
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
        catch(ResponseDequeueFailure&) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
    }
    api.close();
}

```

GetGlobalConditionsForJobReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetGlobalConditionsForJobReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetGlobalConditionsForJobRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetGlobalConditionsForJobRspSet object.

setJobID

The setJobID method in the GetGlobalConditionsForJobReq class sets the Job ID of the job whose global conditions to retrieve. This method has the following prototype:

```
void setJobID (const int iJobID)
```

iJobID

Specifies the ID of the job whose global conditions to retrieve.

setName

The setName method in the GetGlobalConditionsForJobReq class specifies the name of the job whose global conditions are to be retrieved. This method has the following prototype:

```
setName (const std::string & strJobName)
```

strName

Specifies the name of the job whose global conditions are to be retrieved.

GetGlobalConditionsForJobRsp

extends

ApiResponse

header

AsPublicSDK.h

getDetailOwner

The getDetailOwner method in the GetGlobalConditionsForJobRsp class returns the global variable owner. This method has the following prototype:

```
std::string getDetailOwner () const
```

getDetailPermission

The `getDetailPermission` method in the `GetGlobalConditionsForJobRsp` class returns the global variable permission. This method has the following prototype:

```
std::string getDetailPermission () const
```

getDetailValue

The `getDetailValue` method in the `GetGlobalConditionsForJobRsp` class returns the global variable value. This method has the following prototype:

```
std::string getDetailValue () const
```

getExpectedValue

The `getExpectedValue` method in the `GetGlobalConditionsForJobRsp` class returns the global variable expected value. This method has the following prototype:

```
std::string getExpectedValue () const
```

getIsGlobalDetailAvailable

The `getIsGlobalDetailAvailable` method in the `GetGlobalConditionsForJobRsp` class returns TRUE if you have access to detailed information about the global condition. This method has the following prototype:

```
bool getIsGlobalDetailAvailable () const
```

getName

The `getName` method in the `GetGlobalConditionsForJobRsp` class returns the global variable name. This method has the following prototype:

```
std::string getName () const
```

getOperator

The `getOperator` method in the `GetGlobalConditionsForJobRsp` class returns the operator for this global condition comparison. This method has the following prototype:

```
std::string getOperator () const
```

GetGlobals

GetGlobals retrieves global variables and associated attributes based on the requested name pattern. Either ALL or % will retrieve all defined global variables.

If the requesting user has appropriate access to the AUTOREP list security resource, the API does not verify security for each global variable returned. Otherwise, the API verifies read access for each global variable. If security is denied, the API returns an empty response (i.e. the name and value fields will be zero-length strings).

GetGlobals is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetGlobals API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetGlobalsReq req;
    req.setNamePattern("ALL");
    GetGlobalsRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetGlobalsRsp rsp;
```

```
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "\n " << rsp.name() << " = " << rsp.value() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
catch(ResponseDequeueFailure&) {
    std::cerr << "ResponseDequeueFailure thrown." << std::endl;
```

```
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

GetGlobalsReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetGlobalsReq class performs the API call. This method has the following prototype:

void execute (AsApi &*api*, GetGlobalsRspSet &*rsp*)

api

Specifies an AsApi object.

rsp

Specifies a GetGlobalsRspSet object.

setNamePattern

The setNamePattern method in the GetGlobalsReq class specifies the name pattern of global variables to retrieve. This method has the following prototype:

void setNamePattern (const std::string &*name*)

name

Specifies the fully-qualified name or a name pattern of the global variables to retrieve.

GetGlobalsRsp

```
extends  
    ApiResponse  
header  
    AsPublicSDK.h
```

haveAccess

The haveAccess method in the GetGlobalsRsp class returns TRUE if the caller has read access, otherwise returns FALSE. This method has the following prototype:

```
bool haveAccess () const
```

modifyTime

The modifyTime method in the GetGlobalsRsp class returns the time of the last change to the global variable. This method has the following prototype:

```
time_t modifyTime () const
```

name

The name method in the GetGlobalsRsp class returns the global variable name. This method has the following prototype:

```
std::string name () const
```

value

The value method in the GetGlobalsRsp class returns the global variable value. This method has the following prototype:

```
std::string value () const
```

GetIntCodes

GetIntCodes retrieves the intcode values. In the product there are certain integer properties that have special meaning and are commonly referred to as intcodes. An intcode has a type, and integer value, and a string representation. An example is job status. A job's status is an integer value, 1 for example. This indicates that the job is in RUNNING. That knowledge is embodied in the intcode concept. In this example, the intcodes type is 'status'; its integer value is 1; and its string representation is RUNNING.

These values are stored in the database, and are subject to change. The entire list of intcodes can be retrieved using this CAT2 API. The intent is that the client application would call this API once and cache the values as a table of some sort. Then when an intcode is retrieved, the corresponding string representation of that integer can be retrieved.

There are no security checks done for this API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetIntCodes API

```
AsApi api;  
try {  
    api.openRemote(waae_port, waae_server);  
    GetIntCodesReq req;
```

```
GetIntCodesRspSet rset;
req.execute(api, rset);
while (rset.hasMore()) {
    GetIntCodesRsp rsp;
    rset.next(rsp);
    if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
    std::cout << "getIntCodeType: " << rsp.getIntCodeType() << std::endl;
    std::cout << "getIntCodeText: " << rsp.getIntCodeText() << std::endl;
    std::cout << "getIntCode : " << rsp.getIntCode() << std::endl;
}
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
```

```
    }
    catch(NoResponseAvailable&)
    {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&)
    {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e)
    {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

GetIntCodesReq

extends

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the GetIntCodesReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetIntCodesRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetIntCodesRspSet object.

GetIntCodesRsp

```
extends
    ApiResponse
header
    AsPublicSDK.h
```

getIntCode

The getIntCode method in the GetIntCodesRsp class returns the int code value. This method has the following prototype:

```
int getIntCode () const
```

getIntCodeText

The getIntCodeText method in the GetIntCodesRsp class returns the int code text, for example, FORKFAIL, KILLJOB, or ACTIVATED. This method has the following prototype:

```
std::string getIntCodeText () const
```

getIntCodeType

The getIntCodeType method in the GetIntCodesRsp class returns the int code type. This method has the following prototype:

```
std::string getIntCodeType () const
```

Valid types are:

- Alarm
- Event
- Sound
- Status

GetJobLog

GetJobLog requests a specific job log from the application server. GetJobLog is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the GetJobLog API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetJobLogReq req;
    req.setName("MyJob");
    req.setJobLogType(JlAgent);
    req.setJobRunNumber(-1); // get the latest.
    req.setJobRetryNumber(-1);
    GetJobLogRsp rsp;
    req.execute(api, rsp);
    if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
    std::cout << (char*)rsp.getLog() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

GetJobLogReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetJobLogReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetJobLogRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a GetJobLogRsp object.

setJobLogType

The setJobLogType method in the GetJobLogReq class specifies the type of job log to retrieve. This method has the following prototype:

```
void setJobLogType (eJobLogType jobLogType)
```

jobLogType

Specifies the type of job log requested. Possible values are:

JIAgent

Specifies the remote Agent file corresponding to the given run number and nTry.

JlJobStdout

Specifies the standard out file.

JlJobStderr

Specifies the standard error file.

JIAgentProfile

Specifies the profile used to source this job.

setJobRetryNumber

The setJobRetryNumber method in the GetJobLogReq class specifies the number of times to try to retrieve the job log. This method has the following prototype:

```
void setJobRetryNumber (int iJobRetryNumber)
```

iJobRetryNumber

Specifies the job retry number.

setJobRunNumber

The setJobRunNumber method in the GetJobLogReq class specifies the job run number. This method has the following prototype:

```
void setJobRunNumber (int iJobRunNumber)
```

iJobRunNumber

Specifies the job run number. -1 requests the latest job run.

setName

The setName method in the GetJobLogReq class specifies the name of the job for which to retrieve the job log. This method has the following prototype:

```
void setName (const std::string &strJobName)
```

strJobName

Specifies the name of the job for which to retrieve the job log.

GetJobLogRsp

extends

ApiResponse

header

AsPublicSDK.h

getLog

The getLog method in the GetJobLogRsp class returns the requested log contents. This method has the following prototype:

```
void * getLog ()
```

getLogSize

The `getLogSize` method in the `GetJobLogRsp` class returns the length of the fetched job log. This method has the following prototype:

```
int getLogSize ()
```

GetJobRunsWithFilter

`GetJobRunsWithFilter` retrieves the current dynamic status of the requested jobs. The API verifies read access for each job returned. If access is denied, a placeholder response is returned, indicating that access was denied.

`GetJobRunsWithFilter` is a CAT2 API.

Executing this API can result in the following exceptions:

- `ApiExceptionWithInfo`
- `ApiTimeout`
- `AsExternalValidationException`
- `AsFieldValidationException`
- `AsGeneralErrorException`
- `AsObjectDoesNotExistException`
- `AsSecurityException`
- `AsUnknownErrorException`
- `Cat2StartFailure`
- `CredsNotAuthorized`
- `FailedToSetCreds`
- `InternalResponseError`
- `InternalUnexpected`
- `MissingFinalResponse`
- `NoResponseAvailable`
- `ResponseDequeueFailure`

Example: Sample Code for Executing the GetJobRunsWithFilter API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetJobRunsWithFilterReq req;

    // Set up the attributes fields (see eJobRunAttribute in AsPublicSDK.h for
    attributes).
#define NumAttr 5
    int attr[NumAttr] = {
        JraEvent,
        JraJobName,
        JraApplication,
        JraNtry,
        JraStatus
    };
    std::vector<int> attributes(attr, attr+NumAttr);
    req.setAttributes(attributes);

    FilterOr* o1 = new FilterOr;
    o1->add(new FilterInt(autosys::JR_Event, 101, autosys::FoEqual));
    o1->add(new FilterInt(autosys::JR_Event, 102, autosys::FoEqual));
    req.setFilter(o1);

    GetJobRunsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobRunsWithFilterRsp rsp;
        rset.next(rsp);
        int iNumAttributes = rsp.getAttributes();
        for (int i=0; i < iNumAttributes; i++) {
            if( rsp.getAttributeType(i) == AtInt ) {
                std::cout << "Int attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getInt(rsp.getAttributeName(i)) << "]" << std::endl;
            }
            else if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }
        }
    }
}
```

```
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
```

```
        catch(MissingFinalResponse&)
            std::cerr << "MissingFinalResponse thrown." << std::endl;
        }
        catch(NoResponseAvailable&)
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
        catch(ResponseDequeueFailure&)
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e)
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
    api.close();
```

GetJobRunsWithFilterReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetJobRunsWithFilterReq class performs the API call. This method has the following prototype:

void execute (AsApi &*api*, GetJobRunsWithFilterRspSet &*rset*)

api

Specifies an AsApi object.

rset

Specifies a GetJobRunsWithFilterRspSet object.

setAttributes

The `setAttributes` method in the `GetJobRunsWithFilterReq` class sets attributes on which to report. This method has the following prototype:

```
void setAttributes (std::vector< int > attributes)
```

attributes

Specifies an integer array containing one or more of the following attributes:

- `JraEvent`
- `JraJobName`
- `JraGroup`
- `JraApplication`
- `JraJobType`
- `JraJobTypeChar`
- `JraTimezone`
- `JraTargetMachine`
- `JraJoid`
- `JraRunNum`
- `JraNtry`
- `JraStartTime`
- `JraEndTime`
- `JraStatus`
- `JraExitCode`
- `JraRunMachine`
- `JraSvcDeskStatus`
- `JraSvcDeskHandle`
- `JraOverrideNum`
- `JraBoxJoid`
- `JraBoxName`
- `JraEventTime`
- `JraQueStatus`
- `JraEventText`
- `JraEventStatus`
- `JraDescription`
- `JraEventQueStamp`

setFilter

The setFilter method in the GetJobRunsWithFilterReq class specifies the filter for the job. This method has the following prototype:

```
void setFilter (Filter *pFilter)  
pFilter
```

Specifies the jobruns filter. If setFilter is not called or called with a NULL value, all job runs are returned.

GetJobRunsWithFilterRsp

extends

ApiResponse

header

AsPublicSDK.h

getAttributeName

The getAttributeName method in the GetJobRunsWithFilterRsp class returns an attribute name. This method has the following prototype:

```
eJobRunAttribute getAttributeName (int index) const  
index
```

Specifies the attribute to retrieve. Possible eJobRunAttribute enumerated values are:

- JraEvent
- JraJobName
- JraGroup
- JraApplication
- JraJobType
- JraJobTypeChar
- JraTimezone
- JraTargetMachine
- JraJoid
- JraRunNum

- JraNtry
- JraStartTime
- JraEndTime
- JraStatus
- JraExitCode
- JraRunMachine
- JraSvcDeskStatus
- JraSvcDeskHandle
- JraOverrideNum
- JraBoxJoid
- JraBoxName
- JraEventTime
- JraQueStatus
- JraEventText
- JraEventStatus
- JraDescription
- JraEventQueStamp

getAttributeNameString

The `getAttributeNameString` method in the `GetJobRunsWithFilterRsp` class returns the string version of the attribute name. This method has the following prototype:

```
std::string getAttributeNameString (eJobRunAttribute i) const
```

Possible `eJobRunAttribute` enumerated values are:

- JraEvent
- JraJobName
- JraGroup
- JraApplication
- JraJobType
- JraJobTypeChar

- JraTimezone
- JraTargetMachine
- JraJoid
- JraRunNum
- JraNtry
- JraStartTime
- JraEndTime
- JraStatus
- JraExitCode
- JraRunMachine
- JraSvcDeskStatus
- JraSvcDeskHandle
- JraOverrideNum
- JraBoxJoid
- JraBoxName
- JraEventTime
- JraQueStatus
- JraEventText
- JraEventStatus
- JraDescription
- JraEventQueStamp

getAttributeTypeString

The `getAttributeTypeString` method in the `GetJobRunsWithFilterRsp` class returns the string version of the attribute type. This method has the following prototype:

```
std::string getAttributeTypeString (eAttributeType i) const
```

Possible `eAttributeType` enumerated values are:

- `AtInt`
- `AtString`
- `AtDate`

getAttributes

The `getAttributes` method in the `GetJobRunsWithFilterRsp` class returns the number of attributes to process. This method has the following prototype:

```
int getAttributes () const
```

getAttributeType

The `getAttributeType` method in the `GetJobRunsWithFilterRsp` class returns an attribute type. This method has the following prototype:

```
eAttributeType getAttributeType (int index) const
```

index

Specifies the attribute to retrieve. Possible `eAttributeType` enumerated values are:

- `AtInt`
- `AtString`
- `AtDate`

getDate

The getDate method in the GetJobRunsWithFilterRsp class returns the date value for the given job attribute. This method has the following prototype:

```
time_t getDate (int index) const  
index
```

Specifies the job attribute. Possible values are:

- JraStartTime
- JraEndTime
- JraEventTime
- JraEventQueStamp

getInt

The getInt method in the GetJobRunsWithFilterRsp class returns the integer value. This method has the following prototype:

```
int getInt (int index) const  
index
```

Specifies the attribute value to retrieve. Possible values are:

- JraEvent
- JraJobTyp
- JraJoid
- JraRunNum
- JraNtry
- JraStatus
- JraExitCod
- JraSvcDeskStatus
- JraBoxJoid
- JraQueStatus
- JraEventStatus

getString

The `getString` method in the `GetJobRunsWithFilterRsp` class returns the string value for the given job attribute. This method has the following prototype:

```
std::string getString (int index) const  
index
```

Specifies the job attribute. Possible values are:

- `JraEventText`
- `JraJobName`
- `JraGroup`
- `JraApplication`
- `JraTimezone`
- `JraTargetMachine`
- `JraRunMachine`
- `JraSvcDeskHandle`
- `JraOverrideNum`
- `JraBoxName`
- `JraDescription`

GetJobStatus

GetJobStatus retrieves the current status of a job. This API is used by the autostatus. GetJobStatus is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the GetJobStatus API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetJobStatusReq req;
    req.setName("MyJob");
    GetJobStatusRsp rsp;
    req.execute(api, rsp);
    std::cout << "Job status for MyJob is: " << rsp.jobStatus() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

GetJobStatusReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetJobStatusReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, GetJobStatusRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a GetJobStatusRsp object.

setName

The setName method in the GetJobStatusReq class sets the name of the job whose status to retrieve. This method has the following prototype:

`void setName (const std::string &name)`

name

Specifies the name of the job whose status to retrieve.

GetJobStatusRsp

extends
 ApiResponse
header
 AsPublicSDK.h

jobStatus

The jobStatus method in the GetJobStatusRsp class returns an integer that represents the job status. This method has the following prototype:

`int jobStatus () const`

GetJobsWithFilter

GetJobsWithFilter retrieves a set of job attributes based on the provided job filter and attribute array. If no filter is specified, all jobs are returned. Security is checked and jobs that the requesting user does not have read permission to will not be returned. GetJobsWithFilter is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetJobsWithFilter API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    // Setup filter -- get all jobs where jobname begins with "Test" and
    // machine begins with "local". For each job, return job name, command and machine.
    FilterAnd* myAndFilter = new FilterAnd;
    myAndFilter->add(new FilterString(autosys::J_JobName, "Test%"));
    myAndFilter->add(new FilterString(autosys::J_Machine, "local%"));

    // Select the attributes to be returned.
    std::vector<int> attributes;
    attributes.push_back(KW_JOB_NAME);
    attributes.push_back(KW_COMMAND);
    attributes.push_back(KW_MACHINE);
```

```
GetJobsWithFilterReq req;
req.setAttributes(attributes);
req.setFilter(myAndFilter);

GetJobsWithFilterRspSet rset;
req.execute(api, rset);

// Loop thru each response object and display job properties selected in the
filter.
while( rset.hasMore() ) {
    GetJobsWithFilterRsp rsp;
    rset.next(rsp);

    // Display returned attributes.
    int iNumAttributes = rsp.getAttributes();
    for (int i=0; i < iNumAttributes; i++) {
        if( rsp.getAttributeType(i) == AtString ) {
            std::cout << "String attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
        }
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
```

```
        catch(CredsNotAuthorized&) {
            std::cerr << "CredsNotAuthorized thrown." << std::endl;
        }
        catch(FailedToSetCreds&) {
            std::cerr << "FailedToSetCreds thrown." << std::endl;
        }
        catch(InternalResponseError&) {
            std::cerr << "InternalResponseError thrown." << std::endl;
        }
        catch(InternalUnexpected&) {
            std::cerr << "InternalUnexpected thrown." << std::endl;
        }
        catch(MissingFinalResponse&) {
            std::cerr << "MissingFinalResponse thrown." << std::endl;
        }
        catch(NoResponseAvailable&) {
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
        catch(ResponseDequeueFailure&) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
    }
    api.close();
```

GetJobsWithFilterReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetJobsWithFilterReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetJobsWithFilterRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetJobsWithFilterRspSet object.

setAttributes

The `setAttributes` method in the `GetJobsWithFilterReq` class sets the attributes on which to report. This method has the following prototype:

```
void setAttributes (std::vector< int > attributes)
```

attributes

Specifies an integer array containing one or more of the following attributes:

- `JaApplication`
- `JaBoxFailure`
- `JaBoxName`
- `JaBoxSuccess`
- `JaChkFiles`
- `JaChkFilesString`
- `JaCommand`
- `JaCondition`
- `JaDaysOfWeek`
- `JaDaysOfWeekString`
- `JaDescription`
- `JaExcludeCalendar`
- `JaGroup`
- `JaJobName`
- `JaJobType`
- `JaJoid`
- `JaBoxJoid`
- `JaMachine`
- `JaOwner`
- `JaPermission`
- `JaPermissionString`
- `JaProfile`
- `JaRunCalendar`
- `JaRunMachine`
- `JaRunWindow`
- `JaRunWindowState`
- `JaStartMins`

- JaStartMinsString
- JaStartTimes
- JaStartTimesString
- JaStdErrFile
- JaStdInFile
- JaStdOutFile
- JaTimezone
- JaWatchFile
- JaAlarmIfFail
- JaAppINTry
- JaAutoDelete
- JaAutoHold
- JaBoxTerminator
- JaExitCode
- JaHeartbeatInterval
- JaJcPid
- JaJobLoad
- JaJobTerminator
- JaLastEnd
- JaLastHeartbeat
- JaLastStart
- JaMaxExitSuccess
- JaMaxRunAlarm
- JaMinRunAlarm
- JaNRetrys
- JaNextStart
- JaNTry
- JaPid
- JaPriority
- JaRunNum
- JaRunPriority
- JaStatus
- JaStatusTime

- JaTermRunTime
- JaWatchFileMinSize
- JaWatchInterval
- JaQueName
- JaNextPriority
- JaDateConditions
- JaSvcDeskEnable
- JaSvcDeskImpact
- JaSvcDeskSeverity
- JaSvcDeskPriority
- JaSvcDeskDescription
- JaSvcDeskAttribute
- JaNotifyCondition
- JaNotifyId
- JaNotifyMsg

setFilter

The setFilter method in the GetJobsWithFilterReq class specifies the filter for the job. This method has the following prototype:

```
void setFilter (Filter *pFilter)  
pFilter
```

Specifies a pointer to a filter object. If setFilter is not called, or called with a NULL value, all jobs are returned.

setMaxCount

The setMaxCount method in the GetJobsWithFilterReq class specifies the max count. This method has the following prototype:

```
setMaxCount(int iMaxCnt)  
iMaxCnt
```

Specifies the number of jobs to return.

setRetrieveLongFormOfConditions

The setRetrieveLongFormOfConditions method in the GetJobsWithFilterReq class specifies whether to retrieve the long form of conditions. This method has the following prototype:

```
void setRetrieveLongFormOfConditions (bool bLong)
```

bLong

Specifies which form to retrieve. Possible values are:

TRUE

Retrieves the long form.

FALSE

Retrieves the short form.

[setSkipCount](#)

The `setSkipCount` method in the `GetJobsWithFilterReq` class specifies the skip count. This method has the following prototype:

```
setSkipCount(int iSkipCnt)
```

iSkipCnt

Specifies the number of jobs to skip.

[setSortKeys](#)

The `setSortKeys` method in the `GetJobsWithFilterReq` class sets the sort keys. This method has the following prototype:

```
setSortKeys(std::vector< int > & sortkeys)
```

sortkeys

Specifies an integer array containing one or more sort keys (see `eJobAttribute` in `asapi_cmn.h`):

[GetJobsWithFilterRsp](#)

extends

`ApiResponse`

header

`AsPublicSDK.h`

getAttributeName

The `getAttributeName` method in the `GetJobsWithFilterRsp` class returns an attribute name. This method has the following prototype:

```
std::string getAttributeName (int index) const  
index
```

Specifies the index of the attribute to retrieve.

This method returns the keyword enumeration value.

getAttributeNameString

This `getAttributeNameString` method in the `GetJobsWithFilterRsp` class returns the string version of the attribute name. This method has the following prototype:

```
std::string getAttributeNameString (eJobAttribute attribute) const  
attribute
```

Specifies the attribute name to retrieve. Possible `eJobAttribute` enumerated values returned are:

- `JaApplication`
- `JaBoxFailure`
- `JaBoxName`
- `JaBoxSuccess`
- `JaChkFiles`
- `JaChkFileString`
- `JaCommand`
- `JaCondition`
- `JaDaysOfWeek`
- `JaDaysOfWeekString`
- `JaDescription`
- `JaExcludeCalendar`
- `JaGroup`
- `JaJobName`

- JaJobType
- JaJoid
- JaBoxJoid
- JaMachine
- JaOwner
- JaPermission
- JaPermissionString
- JaProfile
- JaRunCalendar
- JaRunMachine
- JaRunWindow
- JaRunWindowString
- JaStartMins
- JaStartMinsString
- JaStartTimes
- JaStartTimesString
- JaStdErrFile
- JaStdInFile
- JaStdOutFile
- JaTimezone
- JaWatchFile
- JaAlarmIfFail
- JaApplNTry
- JaAutoDelete
- JaAutoHold
- JaBoxTerminator
- JaExitCode
- JaHeartbeatInterval
- JaJcPid
- JaJobLoad

- JaJobTerminator
- JaLastEnd
- JaLastHeartbeat
- JaLastStart
- JaMaxExitSuccess
- JaMaxRunAlarm
- JaMinRunAlarm
- JaNRetrys
- JaNextStart
- JaNTry
- JaPid
- JaPriority
- JaRunNum
- JaRunPriority
- JaStatus
- JaStatusTime
- JaTermRunTime
- JaWatchFileMinSize
- JaWatchInterval
- JaQueName
- JaNextPriority
- JaDateConditions
- JaSvcDeskEnable
- JaSvcDeskImpact
- JaSvcDeskSeverity
- JaSvcDeskPriority
- JaSvcDeskDescription
- JaSvcDeskAttribute
- JaNotifyCondition
- JaNotifyId
- JaNotifyMsg

getAttributeNameString

This `getAttributeNameString` method in the `GetJobsWithFilterRsp` class returns the string value of the attribute name. This method has the following prototype:

```
std::string getAttributeNameString (eKeywords kw) const
```

kw

Specifies a valid keyword enumeration.

This method returns the string equivalent of the `eJobAttribute` value, e.g. "KW_JOB_NAME".

getAttributes

The `getAttributes` method in the `GetJobsWithFilterRsp` class returns the number of attributes contained in the `getJobsWithFilterRsp` object. This method has the following prototype:

```
int getAttributes () const
```

getAttributeType

The `getAttributeType` method in the `GetJobsWithFilterRsp` class returns an attribute type. This method has the following prototype:

```
eAttributeType getAttributeType (int index) const
```

index

Specifies the index of the attribute to retrieve.

Possible `eAttributeType` enumerated values returned are:

- AtInt
- AtString
- AtDate
- AtBoolean
- AtIntList
- AtTimeOfDayList
- AtChkFilesList

getAttributeTypeString

The `getAttributeTypeString` method in the `GetJobsWithFilterRsp` class returns the string version of the attribute type. This method has the following prototype:

```
std::string getAttributeTypeString (eAttributeType type) const  
type
```

Specifies the attribute type to retrieve. Possible `eAttributeType` enumerated values are:

- `AtInt`
- `AtString`
- `AtDate`
- `AtBoolean`
- `AtIntList`
- `AtTimeOfDayList`
- `AtChkFilesList`

getBoolean

The `getBoolean` method in the `GetJobsWithFilterRsp` class returns the Boolean value for the given job attribute. This method has the following prototype:

```
bool getBoolean (int index) const  
index
```

Specifies the index of the job attribute to retrieve. Possible values are:

- `JaAlarmIfFail`
- `JaAutoDelete`
- `JaAutoHold`
- `JaBoxTerminator`
- `JaJobTerminator`

getChar

The `getChar` method in the `GetJobsWithFilterRsp` class returns the character value for the given job attribute. This method has the following prototype:

```
char getChar (int attribute) const
```

attribute

Specifies the job attribute to retrieve. Valid attributes are:

- `JaJobType`
- `JaNotifyCondition`

getChkFiles

The `getChkFiles` method in the `GetJobsWithFilterRsp` class returns the list of check files and the corresponding permissions. This method has the following prototype:

```
std::vector<std::string> getChkFiles (int attribute) const
```

attribute

Specifies the job attribute to retrieve. The valid attribute value is `JaChkFiles`.

getDate

The `getDate` method in the `GetJobsWithFilterRsp` class returns the date value for the given job attribute. This method has the following prototype:

```
time_t getDate (int index) const
```

index

Specifies the index of the job attribute to retrieve. Possible values are:

- `JaLastEnd`
- `JaLastHeartbeat`
- `JaLastStart`
- `JaStatusTime`

getInt

The getInt method in the GetJobsWithFilterRsp class returns the integer value. This method has the following prototype:

```
int getInt (int index) const  
index
```

Specifies the index of the attribute to retrieve. Possible values are:

- JaJobType
- JaJoid
- JaApplNTry
- JaExitCode
- JaHeartbeatInterval
- JaJcPid
- JaJobLoad
- JaMaxExitSuccess
- JaMaxRunAlarm
- JaMinRunAlarm
- JaNRetrys
- JaNTry
- JaPid
- JaPriority
- JaRunNum
- JaRunPriority
- JaStatus
- JaTermRunTime
- JaWatchInterval

GetInts

The `GetInts` method in the `GetJobsWithFilterRsp` class returns the list of integers. This method has the following prototype:

```
std::vector<int> getInts (int attribute) const
```

attribute

Specifies the job attribute to retrieve. Possible values are:

- `JaDaysOfWeek`
- `JaStartMins`

getJobItem

The `getJobItem` method in the `GetJobsWithFilterRsp` class returns the job parm value for the given job attribute. This method has the following prototype:

```
JobItem getJobItem (int index, int jp_index) const
```

index

Specifies the attribute index.

jp_index

Specifies the repeating job parm index.

getLong

The `getLong` method in the `GetJobsWithFilterRsp` class returns a 64-bit integer value. This method has the following prototype:

```
long long getLong(int index) const
```

index

Specifies the index of the attribute to retrieve. This method applies to the `JaWatchFileMinSize` attribute.

getString

The `getString` method in the `GetJobsWithFilterRsp` class returns the string value for the given job attribute. This method has the following prototype:

```
std::string getString (int index) const  
index
```

Specifies the index of the job attribute to retrieve. Possible values are:

- `JaApplication`
- `JaBoxFailure`
- `JaBoxName`
- `JaBoxSuccess`
- `JaChkFilesString`
- `JaCommand`
- `JaCondition`
- `JaDaysOfWeekString`
- `JaDescription`
- `JaExcludeCalendar`
- `JaGroup`
- `JaJobName`
- `JaMachine`
- `JaOwner`
- `JaPermissionString`
- `JaProfile`
- `JaRunMachine`
- `JaRunWindowString`
- `JaStartMinsString`
- `JaStartTimesString`
- `JaStdErrFile`
- `JaStdInFile`
- `JaStdOutFile`
- `JaTimezone`
- `JaWatchFile`

getTimesOfDay

The `getTimesOfDay` method in the `GetJobsWithFilterRsp` class returns the list of hours and minutes. This method has the following prototype:

```
std::vector<std::string> getTimesOfDay (int attribute) const  
attribute
```

Specifies the job attribute to retrieve. Possible values are:

- `JaRunWindow`
- `JaStartTimes`

GetJobTypesDetail

`GetJobTypesDetail` retrieves user-defined job type details from the CA Workload Automation AE database. `GetJobTypesDetail` is a CAT2 API.

Executing this API can result in the following exceptions:

- `ApiExceptionWithInfo`
- `ApiTimeout`
- `AsExternalValidationException`
- `AsFieldValidationException`
- `AsGeneralErrorException`
- `AsObjectDoesNotExistException`
- `AsSecurityException`
- `AsUnknownErrorException`
- `Cat2StartFailure`
- `CredsNotAuthorized`
- `FailedToSetCreds`
- `InternalResponseError`
- `InternalUnexpected`
- `MissingFinalResponse`
- `NoResponseAvailable`
- `ResponseDequeueFailure`

Example: Sample Code for Executing the GetJobTypesDetail API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetJobTypesDetailReq req;
    req.setJobType("ALL");
    GetJobTypesDetailRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetJobTypesDetailRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "getJobType: " << rsp.getJobType() << std::endl;
        std::cout << "getCommand: " << rsp.getCommand() << std::endl;
        std::cout << "getDescription: " << rsp.getDescription() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
```

```
        catch(Cat2StartFailure&) {
            std::cerr << "AsUnknownErrorException thrown." << std::endl;
        }
        catch(CredsNotAuthorized&) {
            std::cerr << "CredsNotAuthorized thrown." << std::endl;
        }
        catch(FailedToSetCreds&) {
            std::cerr << "FailedToSetCreds thrown." << std::endl;
        }
        catch(InternalResponseError&) {
            std::cerr << "InternalResponseError thrown." << std::endl;
        }
        catch(InternalUnexpected&) {
            std::cerr << "InternalUnexpected thrown." << std::endl;
        }
        catch(MissingFinalResponse&) {
            std::cerr << "MissingFinalResponse thrown." << std::endl;
        }
        catch(NoResponseAvailable&) {
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
        catch(ResponseDequeueFailure&) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
    }
    api.close();
}
```

GetJobTypesDetailReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetJobTypesDetailReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, GetJobTypesDetailRspSet &rset)`

api

Specifies an AsApi object.

rset

Specifies a GetJobTypesDetailRspSet object.

setJobType

The setJobType method in the GetJobTypesDetailReq class specifies the job type to retrieve. This method has the following prototype:

`void setJobType (const std::string &strType) const`

strType

Specifies the job type to retrieve. Specify ALL to get all job types.

GetJobTypesDetailRsp

extends
 ApiResponse
header
 AsPublicSDK.h

getCommand

The getCommand method in the GetJobTypesDetailRsp class returns the command associated with the job type. This method has the following prototype:

`std::string getCommand () const`

getDescription

The `getDescription` method in the `GetJobTypesDetailRsp` class returns the description of the job type. This method has the following prototype:

```
std::string getDescription () const
```

getJobType

The `getJobType` method in the `GetJobTypesDetailRsp` class returns the job type. This method has the following prototype:

```
std::string getJobType () const
```

GetLastEvtNum

`GetLastEvtNum` returns the event number of the most recent event in the event table. `GetLastEvtNum` is a CAT1 API.

Executing this API can result in the following exceptions:

- `ApiExceptionWithInfo`
- `ApiTimeout`
- `CredsNotAuthorized`
- `FailedToSetCreds`
- `InternalResponseError`

Example: Sample Code for Executing the GetLastEvtNum API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetLastEvtNumReq req;
    GetLastEvtNumRsp rsp;
    req.execute(api, rsp);
    std::cout << "Last Event Number=" << rsp.getLastEventNumber() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
```

```

        catch(ApiTimeout&) {
            std::cerr << "ApiTimeout thrown." << std::endl;
        }
        catch(CredsNotAuthorized&) {
            std::cerr << "CredsNotAuthorized thrown." << std::endl;
        }
        catch(FailedToSetCreds&) {
            std::cerr << "FailedToSetCreds thrown." << std::endl;
        }
        catch(InternalResponseError&) {
            std::cerr << "InternalResponseError thrown." << std::endl;
        }
    }
    api.close();
}

```

GetLastEvtNumReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetLastEvtNumReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetLastEvtNumRsp &rsp)
```

api
 Specifies an AsApi object.

rsp
 Specifies a GetLastEvtNumRsp object.

GetLastEvtNumRsp

extends
 ApiResponse
header
 AsPublicSDK.h

getLastEventNumber

The getLastEventNumber method in the GetLastEvtNumRsp class returns the most recent event number. This method has the following prototype:

```
int getLastEventNumber () const
```

GetMachDefs

GetMachDefs retrieves machine definitions for the requested machine. This is the API used by autorep -M -q.

This API does not accept wildcards. The code must specify a specific machine name or ALL to retrieve all machine definitions.

If the user has list access to the AUTOREP security resource, the API does not verify individual machine security. When the user does not have the appropriate list access and does not have appropriate access the individual machine, the API does not return a response.

Note that the machines contained in a virtual machine are returned within the same response. User will need to call the getCompMachines method to retrive the component machines within the virtual machine.

GetMachDefs is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetMachDefs API

```
AsApi api;  
try {
```

```
    api.openRemote(waae_port, waae_server);
    GetMachDefsReq req;
    req.setNamePattern("ALL");
    GetMachDefsRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetMachDefsRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo())      std::cout << rsp.info() << std::endl;
        if (rsp.haveAccess() ) {
            std::cout << "Machine Name   : " << rsp.getMachineName() << std::endl;
            std::cout << "Machine Type   : " << rsp.getType() << std::endl;
            std::list<ComponentMachine> compMachs = rsp.getCompMachines();
            if (compMachs.size())
            {
                std::list<ComponentMachine>::const_iterator iter;
                for (iter=compMachs.begin(); iter != compMachs.end(); iter++)
                {
                    std::cout << "Comp Mach Name : " << (*iter).getCompMachName() <<
std::endl;
                    std::cout << "Factor       : " << (*iter).getFactor() << std::endl;
                    std::cout << "Max Load     : " << (*iter).getMaxLoad() << std::endl;
                }
            }
        }
    }
    catch(ApiTimeout&){
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(AsExternalValidationException&){
        std::cerr << "AsExternalValidationException thrown." << std::endl;
    }
    catch(AsFieldValidationException&){
        std::cerr << "AsFieldValidationException thrown." << std::endl;
    }
    catch(AsGeneralErrorException&){
        std::cerr << "AsGeneralErrorException thrown." << std::endl;
    }
    catch(AsObjectDoesNotExistException&){
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&){
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&){
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&){
```

```
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&) {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&) {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&) {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&) {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&) {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
```

GetMachDefsReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetMachDefsReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetMachDefsRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetMachDefsRspSet object.

setNamePattern

The setNamePattern method in the GetMachDefsReq class specifies the machine name to use in the request. This method has the following prototype:

```
void setNamePattern (const std::string &name)
```

name

Specifies the name of the machine for which to request the machine definition. Set this value to ALL to return all machine definitions.

GetMachDefsRsp

extends

ApiResponse

header

AsPublicSDK.h

getAdministrator

The getAdministrator method in the GetMachDefsRsp class returns the administrator ID used for provisioning the CA Workload Automation AE agent through DCA. This method has the following prototype:

```
std::string getAdministrator() const
```

getAgentName

The getAgentName method in the GetMachDefsRsp class returns the agent name of the machine. This method has the following prototype:

```
std::string getAgentName() const
```

getAgentOS

The `getAgentOS` method in the `GetMachDefsRsp` class returns the agent operating system. This method has the following prototype:

```
eAgentOSType getAgentOS() const
```

Returns:

- AO_aix
- AO_hpx
- AO_linux
- AO_openvms
- AO_os400
- AO_solaris
- AO_tandem
- AO_windows
- AO_zos

getCharacterCode

The `getCharacterCode` method in the `GetMachDefsRsp` class returns the character code. This method has the following prototype:

```
int getCharacterCode() const
```

Returns one of the following character codes:

- CC_ASCII
- CC_EBCDIC

getCompMachines

The `getCompMachines` method in the `GetMachDefsRsp` class returns the Component Machines of the Virtual Machine. This method has the following prototype:

```
std::string getCompMachines() const
```

getDescription

The `getDescription` method in the `GetMachDefsRsp` class returns the description of the machine. This method has the following prototype:

```
std::string getDescription() const;
```

getEncryptionType

The `getEncryptionType` method in the `GetMachDefsRsp` class returns the encryption type. This method has the following prototype:

```
getEncryptionType() const
```

This method returns one of the following values:

- `ET_NONE`—Encryption disabled
- `ET_DEFAULT`—Default encryption key and type
- `ET_AES`—AES 128-bit encryption

getFactor

The `getFactor` method in the `GetMachDefsRsp` class returns the weighting factor of the machine. This method has the following prototype:

```
double getFactor () const
```

getHeartbeatAttempts

The `getHeartbeatAttempts` method in the `GetMachDefsRsp` class returns the number of heartbeat attempts of the machine. This value is the number of heartbeat signals the scheduler attempts before it sends an SNMP message indicating CA Workload Automation AE inactivity. This method has the following prototype:

```
int getHeartbeatAttempts() const;
```

getHeartbeatFrequency

The `getHeartbeatFrequency` method in the `GetMachDefsRsp` class returns the heartbeat frequency of the machine. This method has the following prototype:

```
int getHeartbeatFrequency() const
```

getMachineName

The `getMachineName` method in the `GetMachDefsRsp` class returns the name of the machine. This method has the following prototype:

```
std::string getMachineName () const
```

getMaxLoad

The `getMaxLoad` method in the `GetMachDefsRsp` class returns the maximum load of the machine. This method has the following prototype:

```
int getMaxLoad() const;
```

getNodeName

The `getNodeName` method in the `GetMachDefsRsp` class returns the node name of the machine. This method has the following prototype:

```
std::string getNodeName() const;
```

getParentName

The `getParentName` method in the `GetMachDefsRsp` class returns the name of the parent machine. This method has the following prototype:

```
std::string getParentName () const
```

getPortNumber

The `getPortNumber` method in the `GetMachDefsRsp` class returns the port number for the machine. This method has the following prototype:

```
int getPortNumber() const;
```

getProvision

The `getProvision` method in the `GetMachDefsRsp` class returns the provision flag. The provision flag is used to indicate whether the Data Center Automation (DCA) Manager should be requested to provision an CA Workload Automation AE agent on this machine. This method has the following prototype:

```
bool getProvision() const
```

getQueueName

The `getQueueName` method in the `GetMachDefsRsp` class returns the queue name for the machine. This method has the following prototype:

```
std::string getQueueName() const;
```

getType

The `getType` method in the `GetMachDefsRsp` class returns the machine type. This method has the following prototype:

```
char getType () const
```

Possible values are:

- r—real machine
- v—virtual machine

haveAccess

The haveAccess method in the GetMachDefsRsp class returns TRUE if the caller has read access to the current machine, otherwise returns FALSE. This method has the following prototype:

```
bool haveAccess () const
```

GetMachJobs

GetMachJobs retrieves the jobs currently running on the specified machines. This API is used by autorep -M -d.

The GetMachJobs API does not support wildcards. To retrieve load information for all machines, set the machine name to ALL. When the machine name is set to ALL, the API verifies security for the AUTOREP list security resource. If permission is granted, the API does not verify security for each job returned. If permission fails, the API verifies job access individually, and returns empty responses for jobs to which the calling user does not have read permission.

GetMachJobs is a CAT2 API.

Executing this API can result in the following exceptions:

- [ApiExceptionWithInfo](#)
- [ApiTimeout](#)
- [AsExternalValidationException](#)
- [AsFieldValidationException](#)
- [AsGeneralErrorException](#)
- [AsObjectDoesNotExistException](#)
- [AsSecurityException](#)
- [AsUnknownErrorException](#)
- [Cat2StartFailure](#)
- [CredsNotAuthorized](#)
- [FailedToSetCreds](#)
- [InternalResponseError](#)
- [InternalUnexpected](#)
- [MissingFinalResponse](#)
- [NoResponseAvailable](#)
- [ResponseDequeueFailure](#)

Example: Sample Code for Executing the GetMachJobs API

```

AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetMachJobsReq req;
    req.setName("ALL");
    GetMachJobsRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetMachJobsRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "Job Name      : " << rsp.getJobName() << std::endl;
        std::cout << "Machine Name   : " << rsp.getMachineName() << std::endl;
        std::cout << "Run Machine    : " << rsp.getRunMachine() << std::endl;
        std::cout << "Queue Name     : " << rsp.getQueueName() << std::endl;
        std::cout << "Job Status     : " << rsp.getJobStatus() << std::endl;
        std::cout << "Job Priority    : " << rsp.getJobPriority() << std::endl;
        std::cout << "Job Run Priority: " << rsp.getJobRunPriority() << std::endl;
        std::cout << "Job Defined Load: " << rsp.getJobDefinedLoad() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}

```

```
        catch(FailedToSetCreds&){  
            std::cerr << "FailedToSetCreds thrown." << std::endl;  
        }  
        catch(InternalResponseError&){  
            std::cerr << "InternalResponseError thrown." << std::endl;  
        }  
        catch(InternalUnexpected&){  
            std::cerr << "InternalUnexpected thrown." << std::endl;  
        }  
        catch(MissingFinalResponse&){  
            std::cerr << "MissingFinalResponse thrown." << std::endl;  
        }  
        catch(NoResponseAvailable&){  
            std::cerr << "NoResponseAvailable thrown." << std::endl;  
        }  
        catch(ResponseDequeueFailure&){  
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;  
        }  
        catch(ApiExceptionWithInfo& e){  
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;  
            std::cerr << e.what() << std::endl;  
        }  
    api.close();
```

[GetMachJobsReq](#)

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetMachJobsReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetMachJobsRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetMachJobsRspSet object.

setName

The setName method in the GetMachJobsReq class specifies the name of the machine jobs to retrieve. This method has the following prototype:

```
void setName (const std::string &name)
```

name

Specifies the name of the machine jobs to retrieve. Wildcards are not supported.
Set this value to ALL to retrieve information for all defined machines.

GetMachJobsRsp

extends

ApiResponse

header

AsPublicSDK.h

getJobDefinedLoad

The getJobDefinedLoad method in the GetMachJobsRsp class returns the defined load of the job. This method has the following prototype:

```
int getJobDefinedLoad () const
```

getJobName

The getJobName method in the GetMachJobsRsp class returns the the name of the job for which information is being returned. This method has the following prototype:

```
std::string getJobName () const
```

getJobPriority

The `getJobPriority` method in the `GetMachJobsRsp` class returns the priority of the job. This method has the following prototype:

```
int getJobPriority () const
```

getJobRunPriority

The `getJobRunPriority` method in the `GetMachJobsRsp` class returns the run priority of the job. This method has the following prototype:

```
int getJobRunPriority () const
```

getJobStatus

The `getJobStatus` method in the `GetMachJobsRsp` class returns the status of the job. This method has the following prototype:

```
int getJobStatus () const
```

getMachineName

The `getMachineName` method in the `GetMachJobsRsp` class returns the machine name from the job definition. This method has the following prototype:

```
std::string getMachineName () const
```

getQueueName

The `getQueueName` method in the `GetMachJobsRsp` class returns the queue name for queued jobs. This method has the following prototype:

```
std::string getQueueName () const
```

getRunMachine

The `getRunMachine` method in the `GetMachJobsRsp` class returns the name of the machine on which the job is running. This method has the following prototype:

```
std::string getRunMachine () const
```

haveAccess

The `haveAccess` method in the `GetMachJobsRsp` class returns TRUE if the caller has read access to the current machine, otherwise returns FALSE. This method has the following prototype:

```
bool haveAccess () const
```

GetMachRuns

GetMachRuns retrieves the dynamic load information for the requested machines. This API is used by autorep -M.

The GetMachRuns API does not support wildcards. To retrieve load information for all machines, set the machine name to ALL. When the machine name is set to ALL, the API verifies security for the AUTOREP list security resource. If permission is granted, the API does not verify security for each machine returned. If permission fails, the API verifies machine access individually, and returns empty responses (zero-length machine names) for machines to which the calling user does not have read permission.

GetMachRuns is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetMachRuns API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetMachRunsReq req;
    req.setName("ALL");
    GetMachRunsRspSet rset;
    req.execute(api, rset);
```

```
while (rset.hasMore()) {
    GetMachRunsRsp rsp;
    rset.next(rsp);
    if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
    std::cout << "Machine Name : " << rsp.getMachineName() << std::endl;
    std::cout << "Machine Type : " << rsp.getType() << std::endl;
    std::cout << "Parent Name : " << rsp.getParentName() << std::endl;
    std::cout << "Queue Name : " << rsp.getQueueName() << std::endl;
    std::cout << "Max Load : " << rsp.getMaxLoad() << std::endl;
    std::cout << "Factor : " << rsp.getFactor() << std::endl;
    std::string strMachStatus;
    if( rsp.getMachineStatus() == MS_ManualOffline )
        strMachStatus = "Manual Offline";
    else if( rsp.getMachineStatus() == MS_AutoOffline )
        strMachStatus = "Auto Offline";
    else if( rsp.getMachineStatus() == MS_Online )
        strMachStatus = "Online";
    else
        strMachStatus = "Unknown";
    std::cout << "Status : " << strMachStatus << std::endl;
}
}
catch(ApiTimeout&)
{
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&)
{
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&)
{
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&)
{
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&)
{
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&)
{
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&)
{
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&)
{
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&)
{
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
```

```

        catch(FailedToSetCreds&) {
            std::cerr << "FailedToSetCreds thrown." << std::endl;
        }
        catch(InternalResponseError&) {
            std::cerr << "InternalResponseError thrown." << std::endl;
        }
        catch(InternalUnexpected&) {
            std::cerr << "InternalUnexpected thrown." << std::endl;
        }
        catch(MissingFinalResponse&) {
            std::cerr << "MissingFinalResponse thrown." << std::endl;
        }
        catch(NoResponseAvailable&) {
            std::cerr << "NoResponseAvailable thrown." << std::endl;
        }
        catch(ResponseDequeueFailure&) {
            std::cerr << "ResponseDequeueFailure thrown." << std::endl;
        }
        catch(ApiExceptionWithInfo& e) {
            std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
            std::cerr << e.what() << std::endl;
        }
    }
    api.close();
}

```

GetMachRunsReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetMachRunsReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, GetMachRunsRspSet &rset)`

api

Specifies an AsApi object.

rset

Specifies a GetMachRunsRspSet object.

setName

The setName method in the GetMachRunsReq class specifies the name of the machine to retrieve. This method has the following prototype:

```
void setName (const std::string &name)  
name
```

Specifies the name of the machine to retrieve. Wildcards are not supported. Set this value to ALL to retrieve information for all defined machines.

GetMachRunsRsp

extends

ApiResponse

header

AsPublicSDK.h

getFactor

The getFactor method in the GetMachRunsRsp class returns the weighting factor of the machine. This method has the following prototype:

```
double getFactor () const
```

getMachineName

The getMachineName method in the GetMachRunsRsp class returns the name of the machine. This method has the following prototype:

```
std::string getMachineName () const
```

getMachineStatus

The getMachineStatus method in the GetMachRunsRsp class returns the machine status. This method has the following prototype:

```
int getMachineStatus ( ) const
```

Possible eMachineStatus enumerated values are:

- MS_UnknownStatus
- MS_ManualOffline
- MS_AutoOffline
- MS_Online

getMaxLoad

The `getMaxLoad` method in the `GetMachRunsRsp` class returns the maximum load of the machine. This method has the following prototype:

```
int getMaxLoad () const
```

getParentName

The `getParentName` method in the `GetMachRunsRsp` class returns the name of the parent machine. This method has the following prototype:

```
std::string getParentName () const
```

getQueueName

The `getQueueName` method in the `GetMachRunsRsp` class returns the queue name for queued jobs. This method has the following prototype:

```
std::string getQueueName () const
```

getType

The `getType` method in the `GetMachRunsRsp` class returns the machine type. This method has the following prototype:

```
char getType () const
```

Possible values are:

- r—real machine
- v—virtual machine

haveAccess

The `haveAccess` method in the `GetMachRunsRsp` class returns TRUE if you have read access to the current machine, otherwise returns FALSE. This method has the following prototype:

```
bool haveAccess () const
```

GetMonitorEvent

GetMonitorEvent provides direct programmatic access to all events in the system. Calling this API in a loop returns CA Workload Automation AE events as they occur.

GetMonitorEvent is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the GetMonitorEvent API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetMonitorEventReq req;
    req.setEventNumber(5939);
    GetMonitorEventRsp rsp;
    req.execute(api, rsp);
    std::cout << "Event number=" << rsp.getEventNumber() << std::endl;
    std::cout << "Job name=" << rsp.getJobName() << std::endl;
    std::cout << "Joid=" << rsp.getJoid() << std::endl;
    std::cout << "Text=" << rsp.getText() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

GetMonitorEventReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetMonitorEventReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, GetMonitorEventRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a GetMonitorEventRsp object.

setEventNumber

The setEventNumber method in the GetMonitorEventReq class specifies an event number to retrieve. This method has the following prototype:

`void setEventNumber (int iEventNumber)`

iEventNumber

Specifies the event number of the event to retrieve.

GetMonitorEventRsp

extends
 ApiResponse
header
 AsPublicSDK.h

getAlarm

The getAlarm method in the GetMonitorEventRsp class returns the alarm property. This method has the following prototype:

`int getAlarm () const`

getApplication

The `getApplication` method in the `GetMonitorEventRsp` class returns the application property. This method has the following prototype:

```
std::string getApplication () const
```

getAutoserv

The `getAutoserv` method in the `GetMonitorEventRsp` class returns the autoserv property. This method has the following prototype:

```
std::string getAutoserv () const
```

getBoxName

The `getBoxName` method in the `GetMonitorEventRsp` class returns the box name. This method has the following prototype:

```
std::string getBoxName () const
```

getEoid

The `getEoid` method in the `GetMonitorEventRsp` class returns the eoid property. This method has the following prototype:

```
std::string getEoid () const
```

getEvent

The `getEvent` method in the `GetMonitorEventRsp` class returns the event property. This method has the following prototype:

```
int getEvent () const
```

getEventNumber

The `getEventNumber` method in the `GetMonitorEventRsp` class returns the event number. This method has the following prototype:

```
int getEventNumber () const
```

getEventTimeGMT

The `getEventTimeGMT` method in the `GetMonitorEventRsp` class returns the event time gmt property. This method has the following prototype:

```
int getEventTimeGMT () const
```

getEventTimeGMTString

The `getEventTimeGMTString` method in the `GetMonitorEventRsp` class returns the event time gmt property as a string. This method has the following prototype:

```
std::string getEventTimeGMTString () const
```

getExitCode

The `getExitCode` method in the `GetMonitorEventRsp` class returns the event code property. This method has the following prototype:

```
int getExitCode () const
```

getGroup

The `getGroup` method in the `GetMonitorEventRsp` class returns the group property. This method has the following prototype:

```
std::string getGroup () const
```

getJcPid

The `getJcPid` method in the `GetMonitorEventRsp` class returns the jc_pid property. This method has the following prototype:

```
int getJcPid () const
```

getJobName

The `getJobName` method in the `GetMonitorEventRsp` class returns the job name property. This method has the following prototype:

```
std::string getJobName () const
```

getJoid

The `getJoid` method in the `GetMonitorEventRsp` class returns the joid property. This method has the following prototype:

```
int getJoid () const
```

getMachine

The `getMachine` method in the `GetMonitorEventRsp` class returns the machine name. This method has the following prototype:

```
std::string getMachine () const
```

getMaxSendTrys

The `getMaxSendTrys` method in the `GetMonitorEventRsp` class returns the `max send try` property. This method has the following prototype:

```
int getMaxSendTrys () const
```

getNTry

The `getNTry` method in the `GetMonitorEventRsp` class returns the `ntry` property. This method has the following prototype:

```
int getNTry () const
```

getPid

The `getPid` method in the `GetMonitorEventRsp` class returns the `pid` property. This method has the following prototype:

```
int getPid () const
```

getPriority

The `getPriority` method in the `GetMonitorEventRsp` class returns the `priority` property. This method has the following prototype:

```
int getPriority () const
```

getQueuePriority

The `getQueuePriority` method in the `GetMonitorEventRsp` class returns the `queue priority` property. This method has the following prototype:

```
int getQueuePriority () const
```

getRunNumber

The `getRunNumber` method in the `GetMonitorEventRsp` class returns the `run number` property. This method has the following prototype:

```
int getRunNumber () const
```

getSecondsInQueue

The `getSecondsInQueue` method in the `GetMonitorEventRsp` class returns the `seconds in queue` property. This method has the following prototype:

```
int getSecondsInQueue () const
```

getStatus

The getStatus method in the GetMonitorEventRsp class returns the status property. This method has the following prototype:

```
int getStatus () const
```

getText

The getText method in the GetMonitorEventRsp class returns the text property. This method has the following prototype:

```
std::string getText () const
```

getToAutoserv

The getToAutoserv method in the GetMonitorEventRsp class returns the ToAutoserv property. This method has the following prototype:

```
std::string getToAutoserv () const
```

GetObjectNames

GetObjectNames returns the object names that match the requested pattern. Read permissions are checked for each object and if the requesting user does not have permission, the name is not returned.

GetObjectNames is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds

- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

[Example: Sample Code for Executing the GetObjectNames API](#)

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetObjectNamesReq req;
    req.setNamePattern("%");
    GetObjectNamesRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetObjectNamesRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "Object Name : " << rsp.getObjectName() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
```

```
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
catch(ResponseDequeueFailure&) {
    std::cerr << "ResponseDequeueFailure thrown." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
api.close();
```

GetObjectNamesReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetObjectNamesReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetObjectNamesRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetObjectNamesRspSet object.

setNamePattern

The setNamePattern method in the GetObjectNamesReq class specifies the object name or pattern to retrieve. This method has the following prototype:

```
void setNamePattern (const std::string &name)
```

name

Specifies a fully-qualified name or name pattern that determines which objects to retrieve.

setObjectType

The setObjectType method in the GetObjectNamesReq class specifies the type of object to retrieve. This method has the following prototype:

```
void setObjectType (eObjectType objecttype)
```

objecttype

Specifies the object type. Possible values are:

- OT_JOB
- OT_MACHINE
- OT_CALENDAR
- OT_GLOBAL_VARIABLE
- OT_EXTENDED_CALENDAR
- OT_CYCLE_CALENDAR
- OT_GLOBAL_BLOB

GetObjectNamesRsp

extends

ApiResponse

header

AsPublicSDK.h

getObjectName

The getObjectName method in the GetObjectNamesReq class returns the name of the object. This method has the following prototype:

```
std::string getObjectName () const
```

GetResourcesWithFilter

GetResourcesWithFilter retrieves a set of resource attributes based on the provided filter and attribute array. If no filter is specified, all resources are returned. Security is checked and resources that the requesting user does not have read permission to will not be returned.

GetResourcesWithFilter is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetResourcesWith Filter API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);

    int attr[] =
{VRA_Name,VRA_Machine,VRA_Description,VRA_Type,VRA_AmountDefined,VRA_AmountInUse};
    std::vector<int> attributes(attr, attr+(sizeof(attr) / sizeof(int)));
    GetResourcesWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter( new autosys::FilterString(autosys::VR_Name, "%" ) );
```

```
GetResourcesWithFilterRspSet rset;
req.execute(api, rset);
while( rset.hasMore() ) {
    GetResourcesWithFilterRsp rsp;
    rset.next(rsp);
    int iNumAttributes = rsp.getAttributes();
    for (int i=0; i < iNumAttributes; i++) {
        if( rsp.getAttributeType(i) == AtInt ) {
            std::cout << "Int attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
            rsp.getInt(rsp.getAttributeName(i)) << "]" << std::endl;
        }
        else if( rsp.getAttributeType(i) == AtString ) {
            std::cout << "String attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
            rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
        }
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
```

```
    }
    catch(InternalResponseError&)
    {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&)
    {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&)
    {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&)
    {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&)
    {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e)
    {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

GetResourcesWithFilterReq

extends

 ApiRequest

header

 AsPublicSDK.h

setFilter

The setFilter method in the GetResourcesWithFilterReq class specifies a filter. This method has the following prototype:

 setFilter(Filter **pFilter*)

pFilter

 Specifies a filter object.

setAttributes

The setAttributes method in the GetResourcesWithFilterReq class sets attributes to report on. This method has the following prototype:

```
setAttributes(std::vector< int > &attributes)
```

attributes

Specifies an integer array containing one or more of the following (see eVirtResourceAttribute enumeration in AsPublicSDK.h):

- VRA_Name
- VRA_Machine
- VRA_Description
- VRA_Type
- VRA_AmountDefined
- VRA_AmountInUse

execute

The execute method in the GetResourcesWithFilterReq class performs the API call. This method has the following prototype:

```
execute(AsApi & api, GetResourcesWithFilterRspSet & rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetResourcesWithFilterRspSet object.

GetResourcesWithFilterRsp

extends

ApiResponse

header

AsPublicSDK.h

getAttributes

The getAttributes method in the GetResourcesWithFilterRsp class returns the attribute count. This method has the following prototype:

```
int getAttributes() const
```

getAttributeName

The `getAttributeName` method in the `GetResourcesWithFilterRsp` class returns the attribute name. This method has the following prototype:

```
eVirtResourceAttribute getAttributeName (int index) const  
index
```

Specifies the index of attribute to retrieve.

This method returns the keyword enumeration value.

getAttributeType

The `getAttributeType` method in the `GetResourcesWithFilterRsp` class returns an attribute type. This method has the following prototype:

```
eAttributeType getAttributeType ( int index ) const  
index
```

Specifies the index of the attribute to retrieve.

Returns:

The following are the possible `eAttributeType` enumerated values returned:

- `AtInt`
- `AtString`

getInt

The `getInt` method in the `GetResourcesWithFilterRsp` class returns the integer value. This method has the following prototype:

```
int getInt ( int index ) const  
index
```

Specifies the index of attribute value to retrieve. Valid attributes are:

- `VraType`
- `VraAmountDefined`
- `VraAmountInUse`

getString

The `getString` method in the `GetResourcesWithFilterRsp` class returns the String value for the given resource attribute. This method has the following prototype:

```
std::string getString ( int index ) const  
attribute
```

Specifies the resource attribute. Valid attributes are:

- `VRA_Name`
- `VRA_Machine`
- `VRA_Description`

getAttributeNameString

The `getAttributeNameString` method in the `GetResourcesWithFilterRsp` class returns the string version of the attribute name.

```
std::string getAttributeNameString (eVirtResourceAttribute attr) const  
attr
```

Specifies the attribute name. The following are the possible `eVirtResourceAttribute` enumerated values:

- `VRA_Name`
- `VRA_Machine`
- `VRA_Description`
- `VRA_Type`
- `VRA_AmountDefined`
- `VRA_AmountInUse`

Returns:

The string equivalent of the `eVirtResourceAttribute` value, for example, `VRA_Name`.

getAttributeTypeString

The `getAttributeTypeString` method in the `GetResourcesWithFilterRsp` class returns the string version of the attribute type. This method has the following prototype:

```
std::string getAttributeTypeString ( eAttributeType attrType ) const  
attrType
```

Specifies the attribute type. The following are the possible `eAttributeType` enumerated values:

- `AtInt`
- `AtString`

Returns:

The string equivalent of the `eAttributeType` value, for example, `AtInt`.

GetResourceUsage

`GetResourceUsage` retrieves a summary of resource usage information and optionally returns detail information. Summary resource usage information includes the resource name, machine name, amount defined and amount available. If detail information is requested, the job name, status, run machine and amount in use are returned.

This API accepts a fully qualified resource name, wildcards, or "ALL" to retrieve all resources. The API also accepts a fully qualified machine name, a machine name with wildcards, or "ALL" to include all machines.

If the user has list access to the AUTOREP security resource, the API does not verify individual machine security. When the user does not have the appropriate list access and does not have appropriate access the individual machine, the API does not return a response.

`GetResourcesWithFilter` is a CAT2 API.

Executing this API can result in the following exceptions:

- `ApiExceptionWithInfo`
- `ApiTimeout`
- `AsExternalValidationException`
- `AsFieldValidationException`
- `AsGeneralErrorException`
- `AsObjectDoesNotExistException`

- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds

- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetResourceUsage API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetResourceUsageReq req;
    req.setName("*");
    req.setDetailLevel(true); // Detail report

    GetResourceUsageRspSet rset;
    req.execute(api, rset);

    while (rset.hasMore()) {
        GetResourceUsageRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        for(int i=0; i < rsp.getSummaryCount(); i++) {
            std::cout << "Resource Name : " << rsp.getResName(i) << std::endl;
            std::cout << "Machine Name : " << rsp.getMachineName(i) << std::endl;
            std::cout << "Amount defined : " << rsp.getAmountDefined(i) << std::endl;
            std::cout << "Amount available: " << rsp.getAmountAvailable(i) <<
std::endl;
        }
        for(int i=0; i < rsp.getDetailCount(); i++) {
            std::cout << "Job Name : " << rsp.getDetailJobName(i) << std::endl;
            std::cout << "Job Status : " << rsp.getDetailJobStatus(i) << std::endl;
            std::cout << "Run Machine : " << rsp.getDetailRunMachine(i) << std::endl;
            std::cout << "Amount in use: " << rsp.getDetailAmountInUse(i) << std::endl;
        }
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
```

```

    }
    catch(AsObjectDoesNotExistException&) {
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&) {
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&) {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&) {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&) {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&) {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&) {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}

```

GetResourceUsageReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetResourceUsageReq class performs the API. This method has the following prototype:

```
execute (AsApi & api, GetResourceUsageRspSet & rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetResourceUsageRspSet object.

setDetailLevel

The setDetailLevel method in the GetResourceUsageReq class specifies the detail level for the usage report. This method has the following prototype:

```
void setDetailLevel ( bool bDetail )
```

bDetail

Returns, if TRUE, the job name, status, machine name and amount in use. This information is returned in addition to the summary information which includes resource name, machine name, amount defined and amount available.

setMachineName

The setMachineName method in the GetResourceUsageReq class specifies the machine name. This method has the following prototype:

```
void setMachineName (const std::string & MachineName)
```

MachineName

Specifies the machine name to retrieve. Only resources that contain the specified machine name are returned. This parameter accepts a fully qualified machine name, wildcards, or ALL to include any machine.

setName

The setName method in the GetResourceUsageReq class specifies the resource name. This method has the following prototype:

```
void setName ( const std::string & ResourceName )
```

ResourceName

Specifies the resources to retrieve. This parameter accepts a fully qualified name, a pattern, or ALL to retrieve all resources

GetResourceUsageRsp

extends
ApiResponse
header
AsPublicSDK.h

getAmountAvailable

The getAmountAvailable method in the GetResourceUsageRsp class returns the amount of available resources. This method has the following prototype:

```
intgetAmountAvailable (int index) const  
index  
Specifies the index of the resource summary row.
```

getAmountDefined

The getAmountDefined method in the GetResourceUsageRsp class returns the amount of defined resources. This method has the following prototype:

```
intgetAmountDefined (int index) const  
index  
Specifies the index of the resource summary row.
```

Returns:

The amount of resources that are currently defined.

getDetailAmountInUse

The getDetailAmountInUse method in the GetResourceUsageRsp class returns the amount of resources in use. This method has the following prototype:

```
intgetDetailAmountInUse ( int index ) const  
index  
Specifies the index of the detail summary row.
```

getDetailCount

The `getDetailCount` method in the `GetResourceUsageRsp` class returns the count of detail rows. This method has the following prototype:

```
int getDetailCount ( ) const
```

Returns:

The count of detail rows returned within a single response object. When `setDetailLevel` is set to true on the request, each response will contain zero or more detail rows. If `getDetailCount()` is greater than zero, then detail information can be obtained by calling detail methods, for example, `getDetailJobName`. Each detail row consists of the job name, job status, run machine, and amount in use.

getDetailJobName

The `getDetailJobName` method in the `GetResourceUsageRsp` class returns the job name. This method has the following prototype:

```
std::string getDetailJobName ( int index ) const
```

index

Specifies the index of the detailed summary row.

getDetailJobNtry

The `getDetailJobNtry` method in the `GetResourceUsageRsp` class returns the job ntry. This method has the following prototype:

```
int getDetailJobNtry(int index) const
```

index

Specifies the index of the detail row.

getDetailJobRunNum

The `getDetailJobRunNum` method in the `GetResourceUsageRsp` class returns the run number. This method has the following prototype:

```
int getDetailJobRunNum(int index) const
```

index

Specifies the index of the detail row.

getDetailJobStatus

The `getDetailJobStatus` method in the `GetResourceUsageRsp` class returns the status of the resource. This method has the following prototype:

```
std::string getDetailJobStatus ( int index ) const  
index
```

Specifies the index of the detail summary row.

getDetailResName

The `getDetailResName` method in the `GetResourceUsageRsp` class returns the resource name for the detail row. This method has the following prototype:

```
std::string getDetailResName(int index) const  
index
```

Specifies the index of the detail row.

This method returns the resource name.

getDetailRunMachine

The `getDetailRunMachine` method in the `GetResourceUsageRsp` class returns the name of the run machine. This method has the following prototype:

```
std::string getDetailRunMachine ( int index ) const  
index
```

Specifies the index of the detail summary row.

getMachineName

The `getMachineName` method in the `GetResourceUsageRsp` class returns the machine name. This method has the following prototype:

```
getMachineName (int index) const  
index
```

Specifies the index of the resource summary row.

[getResDesc](#)

The `getResDesc` method in the `GetResourceUsageRsp` class returns the description of the virtual resource. This method has the following prototype:

```
std::string getResDesc(int index) const
```

index

Specifies the index of the resource summary row.

This method returns the resource description.

[getResName](#)

The `getResName` method in the `GetResourceUsageRsp` class returns the name of the resource. This method has the following prototype:

```
std::string getResName(int index) const
```

index

Specifies the index of the resource summary row.

This method returns the resource name.

[getResType](#)

The `getResType` method in the `GetResourceUsageRsp` class returns the type of virtual resource. This method has the following prototype:

```
std::string getResType(int index) const
```

index

Specifies the index of the resource summary row.

This method returns the resource type.

[getSummaryCount](#)

The `getSummaryCount` method in the `GetResourceUsageRsp` class returns the count of summary rows. This method has the following prototype:

```
int getSummaryCount () const
```

Returns the count of resources returned within a single response object. For each response, there may be one or more resource summary rows depending on how many machines the resource is defined on. Each summary row consists of the resource name, machine name, amount defined and amount of resources currently available.

GetSchedulerLog

GetSchedulerLog retrieves the specified number of lines from the scheduler log with a possible keyword filter. GetSchedulerLog is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the GetSchedulerLog API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetSchedulerLogReq req;
    req.setNumberOfLines(5);
    GetSchedulerLogRsp rsp;
    req.execute(api, rsp);
    std::cout << (char*)rsp.getLog() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

GetSchedulerLogReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetSchedulerLogReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetSchedulerLogRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a GetSchedulerLogRsp object.

Returns

- RS_SUCCESS
- RS_FAILURE

The execute status is returned in the inherited response method status().

setNumberOfLines

The setNumberOfLines method in the GetSchedulerLogReq class sets the number of lines from the end of the EP log to return. This method has the following prototype:

```
void setNumberOfLines (int iLines)
```

iLines

Specifies the number of lines from the end of the EP log to return.

setPattern

The setPattern method in the GetSchedulerLogReq class sets the pattern on which to match each line of the EP log. This method has the following prototype:

```
void setPattern (const std::string &strPattern)
```

strPattern

Specifies the pattern on which to match each line of the EP log. This is used to filter the contents of the log.

GetSchedulerLogRsp

extends
ApiResponse
header
AsPublicSDK.h

getLog

The getLog method in the GetSchedulerLogRsp class returns the requested log contents. This method has the following prototype:

```
void * getLog ()
```

getLogSize

The getLogSize method in the GetSchedulerLogRsp class returns the length of the fetched Scheduler log. This method has the following prototype:

```
int getLogSize ()
```

GetStatistics

GetStatistics requests CA Workload Automation AE statistics. GetStatistics is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetStatistics API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    const int iSecondsPerDay = 86400;
    CTime startdate = 0;
    CTime enddate = CTime::GetCurrentTime();
    startdate = enddate;
    int iDays = iSecondsPerDay * 10;
    startdate -= iDays; // subtract days to get start date.
    GetStatisticsReq req;
    req.setDataType(SiHourly);
    tm tmTmp;
    req.setStartDate(mktime(startdate.GetGmtTm(&tmTmp)));
    req.setEndDate(mktime(enddate.GetGmtTm(&tmTmp)));

    // Set up the attributes fields (see eAggregatedStatistics in asapi_cmn.h for
    attributes).
```

```
#define NumAttr 3
int attr[NumAttr] = {
    AsJobsRunning,
    AsJobsStarting,
    AsJobsSuccess
};
std::vector<int> attributes(attr, attr+NumAttr);
req.setAttributes(attributes);

GetStatisticsRspSet rset;
req.execute(api, rset);
while( rset.hasMore() ) {
    GetStatisticsRsp rsp;
    rset.next(rsp);
    intiNumAttributes = rsp.getAttributes();
    for (int i=0; i < iNumAttributes; i++) {
        if( rsp.getAttributeType(i) == AtInt )
            std::cout << "attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getInt(rsp.getAttributeName(i)) << "]" << std::endl;
        else if( rsp.getAttributeType(i) == AtDouble )
            std::cout << "attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getDouble(rsp.getAttributeName(i)) << "]" << std::endl;
    }
}
catch(ApiTimeout&){
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&){
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&){
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&){
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&){
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&){
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&){
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&){
```

```
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
catch(ResponseDequeueFailure&) {
    std::cerr << "ResponseDequeueFailure thrown." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
api.close();
```

GetStatisticsReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the GetStatisticsReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetStatisticsRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetStatisticsRspSet object.

setAttributes

The setAttributes method in the GetStatisticsReq class sets the attributes on which to report. This method has the following prototype:

```
void setAttributes (std::vector< int > attributes)
```

attributes

Specifies an integer array containing one or more of the following attributes:

- AsJobsRunning
- AsJobsStarting
- AsJobsSuccess
- AsJobsFailure
- AsJobsTerminated
- AsJobsOnHold
- AsJobsOnIce
- AsJobsInactive
- AsJobRuns
- AsJobFailures
- AsJobForceStarts
- AsJobRestarts
- AsJobKills

- AsJobOpenSvcDesk
- AsAlarmResponseTimeAvg
- AsAlarmTotal
- AsAlarmsUnanswered
- AsAlarmsJobFailure
- AsAlarmsStartJobFailure
- AsAlarmsMaxRetrys
- AsAlarmsMaxRunTime
- AsAlarmsMinRunTime
- AsAlarmsDatabaseRollover
- AsAlarmsSchedulerRollover
- AsAlarmsSchedulerShutdown
- AsTotalEvents
- AsTotalEventLatency

[setDataType](#)

The `setDataType` method in the `GetStatisticsReq` class specifies the data type to retrieve. This method has the following prototype:

```
void setDataType (eStatisticsInterval datatype)
```

datatype

Specifies one of the following valid data types to retrieve (see the `eStatisticsInterval` enumeration in `AsPublicSDK.h`):

- SiHourly
- SiDaily
- SiWeekly
- SiMonthly

setEndDate

The `setEndDate` method in the `GetStatisticsReq` class specifies the end date for retrieval. This method has the following prototype:

```
void setEndDate (time_t enddate)
```

enddate

Specifies the end date for retrieval.

setStartDate

The `setStartDate` method in the `GetStatisticsReq` class specifies the start date for retrieval. This method has the following prototype:

```
void setStartDate (time_t startdate)
```

startdate

Specifies the start date for retrieval.

GetStatisticsRsp

extends
ApiResponse
header
AsPublicSDK.h

getAttributeName

The `getAttributeName` method in the `GetStatisticsRsp` class returns an attribute name. This method has the following prototype:

```
eAggregatedStatistics getAttributeName (int index) const  
index
```

Specifies the name of the attribute to retrieve. Valid `eAggregatedStatistics` enumerated values are:

- AsJobsRunning
- AsJobsStarting
- AsJobsSuccess
- AsJobsFailure
- AsJobsTerminated
- AsJobsOnHold
- AsJobsOnIce
- AsJobsInactive
- AsJobRuns
- AsJobFailures
- AsJobForceStarts
- AsJobRestarts
- AsJobKills
- AsJobOpenSvcDesk
- AsAlarmResponseTimeAvg
- AsAlarmTotal
- AsAlarmsUnanswered
- AsAlarmsJobFailure
- AsAlarmsStartJobFailure
- AsAlarmsMaxRetrys

- AsAlarmsMaxRunTime
- AsAlarmsMinRunTime
- AsAlarmsDatabaseRollover
- AsAlarmsSchedulerRollover
- AsAlarmsSchedulerShutdown
- AsTotalEvents
- AsTotalEventLatency

getAttributeNameString

The `getAttributeNameString` method in the `GetStatisticsRsp` class returns the string equivalent of the `eAggregatedStatistics` value. This method has the following prototype:

```
std::string getAttributeNameString (eAggregatedStatistics i) const
```

Valid `eAggregatedStatistics` enumerated values are:

- AsJobsRunning
- AsJobsStarting
- AsJobsSuccess
- AsJobsFailure
- AsJobsTerminated
- AsJobsOnHold
- AsJobsOnIce
- AsJobsInactive
- AsJobRuns
- AsJobFailures
- AsJobForceStarts
- AsJobRestarts
- AsJobKills
- AsJobOpenSvcDesk
- AsAlarmResponseTimeAvg
- AsAlarmTotal
- AsAlarmsUnanswered
- AsAlarmsJobFailure
- AsAlarmsStartJobFailure
- AsAlarmsMaxRetrys

- AsAlarmsMaxRunTime
- AsAlarmsMinRunTime
- AsAlarmsDatabaseRollover
- AsAlarmsSchedulerRollover
- AsAlarmsSchedulerShutdown
- AsTotalEvents
- AsTotalEventLatency

[getAttributes](#)

The `getAttributes` method in the `GetStatisticsRsp` class returns the number of attributes to process. This method has the following prototype:

```
int getAttributes () const
```

getAttributeType

The `getAttributeType` method in the `GetStatisticsRsp` class returns an attribute type. This method has the following prototype:

```
eAttributeType getAttributeType (int index) const  
index
```

Specifies the type of attribute to retrieve. Valid `eAttributeType` enumerated values are:

- `AtInt`
- `AtDouble`

getAttributeTypeString

The `getAttributesTypeString` method in the `GetStatisticsRsp` class returns the string equivalent of the `eAttributeType` value. This method has the following prototype:

```
std::string getAttributeTypeString (eAttributeType i) const  
eAttributeType
```

Specifies one of the following `eAttributeType` enumerated values:

- `AtInt`
- `AtDouble`

getDouble

The `getDouble` method in the `GetStatisticsRsp` class returns the double value. This method has the following prototype:

```
double getDouble (int index) const  
index
```

Specifies the attribute value to retrieve.

getInt

The `getInt` method in the `GetStatisticsRsp` class returns the integer value. This method has the following prototype:

```
int getInt (int index) const  
index
```

Specifies the integer value to retrieve.

GetUniqueNames

GetUniqueNames returns the unique list of names for the specified type. The type can be either owner, machine or run machine.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the GetUniqueNames API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    GetUniqueNamesReq req;
    req.setUniqueNameType(UNT_MACHINE);
    GetUniqueNamesRspSet rset;
    req.execute(api, rset);
    while (rset.hasMore()) {
        GetUniqueNamesRsp rsp;
        rset.next(rsp);
        std::cout << "getUniqueName: " << rsp.getUniqueName() << std::endl;
    }
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
```

```
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
catch(ResponseDequeueFailure&) {
    std::cerr << "ResponseDequeueFailure thrown." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
api.close();
```

GetUniqueNamesReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the GetUniqueNamesReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, GetUniqueNamesRspSet &rset)
```

api

Specifies an AsApi object.

rset

Specifies a GetUniqueNamesRspSet object.

setUniqueNameType

The setUniqueNameType method in the GetUniqueNamesReq class specifies the type of unique names to return. This method has the following prototype:

```
void setUniqueNameType (eUniqueNameType eType)
```

eType

Specifies the type of unique names to return. Possible values are:

UNT_OWNER

>Returns all unique job owners.

UNT_MACHINE

>Returns all unique machines defined in existing jobs.

UNT_RUN_MACHINE

>Returns all unique run machines for existing jobs.

GetUniqueNamesRsp

```
extends  
    ApiResponse  
header  
    AsPublicSDK.h
```

getUniqueName

The `getUniqueName` method in the `GetUniqueNamesRsp` class returns the unique name. This method has the following prototype:

```
std::string getUniqueName () const
```

I5OthersItem

The I5OthersItem is the container class for specifying i5 others arguments in job definitions.

Example: Sample Code for Executing the I5OthersItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME,(std::string)"myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_I5);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_I5_ACTION, EN_I5ACTION_COMMAND);
    req.set(KW_I5_CC_EXIT, EN_I5CCEXIT_USER);
    req.set(KW_I5_CURR_LIB, (std::string) "myLibrary");
    req.set(KW_I5_NAME, (std::string) "myCommand");
    I5OthersItem I5others;
    I5others.set(KW_I5_OTHERS, (std::string) "other1=val1");
    I5others.set(KW_I5_OTHERS, (std::string) "other2=val2");
    I5others.set(KW_I5_OTHERS, (std::string) "other3=val3");
    req.add(&I5others);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_I5_CURR_LIB, KW_I5_NAME,
KW_I5_OTHERS};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) / 
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    // Now retrieve the job that was inserted

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
```

```
GetJobsWithFilterRsp rsp;
rset.next(rsp);
for (int i=0; i < rsp.getAttributes(); i++) {

    // Display keywords that are of type string.
    if( rsp.getAttributeType(i) == AtString ) {
        std::cout << "String attribute [" <<
        rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
        rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
    }

    // Display job items.
    else if( rsp.getAttributeType(i) == AtJobItem ) {
        autosys::JobItem myJobitem = rsp.getJobItem(i);
        std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
        for( int j=0; j < myJobitem.getAttributes(); j++ ) {
            // Only displaying jobitem keywords that are of type string.
            std::cout << "Jobitem string attribute [" <<
            rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
            myJobitem.getString( (int)keywords[j] ) << "]" <<
            std::endl;
        }
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```

InsertCalendarDate

InsertCalendarDate is used to insert a standard calendar into the CA Workload Automation AE database.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the InsertCalendarDate API

```
// Setup calendar.  
std::vector<std::string> call_dates_;  
call_dates_.push_back("01/04/2006 00:00");  
call_dates_.push_back("03/18/2006 00:00");  
call_dates_.push_back("04/04/2006 00:00");  
call_dates_.push_back("05/18/2006 00:00");  
call_dates_.push_back("06/12/2006 00:00");  
call_dates_.push_back("06/18/2006 00:00");  
call_dates_.push_back("07/07/2006 00:00");  
call_dates_.push_back("08/12/2006 00:00");  
  
AsApi api;  
try {  
    api.openRemote(waae_port, waae_server);  
    InsertCalendarDateReq req;  
    req.setName("MyCalendar");
```

```
    std::vector<std::string>::size_type i = 0;
    for (i = 0; i < call_dates_.size(); ++i) {
        req.appendDate(call_dates_[i]);
    }
    InsertCalendarDateRsp rsp;
    req.execute(api, rsp);
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
    std::cerr << "NoResponseAvailable thrown." << std::endl;
}
```

```
    catch(ResponseDequeueFailure&) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
```

InsertCalendarDateReq

extends

 ApiRequest

header

 AsPublicSDK.h

InsertCalendarDateReq

This constructor has the following syntax:

```
InsertCalendarDateReq(const InsertCalendarDateReq & arg)
```

appendDate

The appendDate method in the InsertCalendarDateReq class adds a date to the list of dates that define the standard calendar. This method has the following prototype:

```
void appendDate (const std::string &strDate)
```

strDate

Specifies the date to append to the list, using the standard format MM/DD/YYYY
HH:MM.

execute

The execute method in the InsertCalendarDateReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, SimpleRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setName

The setName method in the InsertCalendarDateReq class sets the name of the standard calendar to create. This method has the following prototype:

```
void setName (const std::string &strCalendarName)
```

InsertCycleCalendarDays

InsertCycleCalendarDays appends cycle calendar dates into the CA Workload Automation AE database. InsertCycleCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertCycleCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertCycleCalendarDaysReq req;
    req.setName("CycleTest");
    req.appendCycle("07/07/2006 00:00:00", "07/27/2006 00:00:00");
    req.appendCycle("08/07/2006 00:00:00", "08/27/2006 00:00:00");
    InsertCycleCalendarDaysRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertCycleCalendarDaysReq

extends
 ApiRequest
header
 AsPublicSDK.h

appendCycle

The appendCycle method in the InsertCycleCalendarDaysReq class appends date ranges to the cycle calendar. This method has the following prototype:

```
void appendCycle (const std::string &strStartDate, const std::string &strEndDate)
```

strStartDate

Specifies the date to start the cycle. Must be of the standard format: MM/DD/YYYY HH:MM:SS.

strEndDate

Specifies the date to end the cycle. Must be of the standard format: MM/DD/YYYY HH:MM:SS.

execute

The execute method in the InsertCycleCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, InsertCycleCalendarDaysRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an InsertCycleCalendarDaysRsp object.

setName

The setName method in the InsertCycleCalendarDaysReq class specifies the name of the cycle calendar to create. This method has the following prototype:

```
void setName (const std::string &strCycleCalendarName)
```

strCycleCalendarName

Specifies the name of the cycle calendar to create.

InsertExtendedCalendarDays

InsertExtendedCalendarDays inserts an extended calendar into the CA Workload Automation AE database. InsertExtendedCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertExtendedCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertExtendedCalendarDaysReq req;
    req.setCalendarName("MyCalendar");
    req.setWorkdays("XXXXX_");
    req.setHolidayCalendar("hcal");
    req.setCycleCalName("cycal");
    req.setHolidayAction(CaPreviousWorkDay);
    req.setWeekendAction(CaNextDay);
    req.setAdjustment(1);
    req.setDateCon("DAILY");
    InsertExtendedCalendarDaysRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertExtendedCalendarDaysReq

extends
 ApiRequest
header
 AsPublicSDK.h

InsertExtendedCalendarDaysReq

This constructor has the following syntax:

```
InsertExtendedCalendarDaysReq (const InsertExtendedCalendarDaysReq & arg)
```

execute

The execute method in the InsertExtendedCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, SimpleRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setAdjustment

The setAdjustment method in the InsertExtendedCalendarDaysReq class specifies the number of days before or after the condition setting. This method has the following prototype:

```
void setAdjustment (int iAdjustment) const
```

iAdjustment

Specifies the positive or negative number that indicates the days before or after the condition setting

setCalendarName

The setCalendarName method in the InsertExtendedCalendarDaysReq class specifies the calendar name to insert. This method has the following prototype:

```
void setCalendarName (const std::string &strCalendarName) const
```

strCalendarName

Specifies the name of the calendar to be inserted

setCycleCalName

The `setCycleCalName` method in the `InsertExtendedCalendarDaysReq` class sets the name of the cycle calendar. This method has the following prototype:

```
void setCycleCalName (const std::string &strCycleCalName) const  
strCycleCalName
```

Specifies the name of the cycle calendar to use.

setDateCon

The `setDateCon` method in the `InsertExtendedCalendarDaysReq` class sets the condition keyword string. This method has the following prototype:

```
void setDateCon (const std::string &strDateCon) const  
strDateCon
```

Specifies the condition keyword string.

setHolidayAction

The `setHolidayAction` method in the `InsertExtendedCalendarDaysReq` class specifies the action to take when the work falls on a holiday. This method has the following prototype:

```
void setHolidayAction (eCalendarAction eHolidayAction) const  
eHolidayAction
```

Specifies the action to take when the work falls on a holiday. Valid actions are:

- `CaDefault`
- `CaOnly`
- `CaSchedule`
- `CaNextDay`
- `CaNextWorkDay`
- `CaPreviousWorkDay`

setHolidayCalendar

The `setHolidayCalendar` method in the `InsertExtendedCalendarDaysReq` class specifies the name of the holiday calendar to use. This method has the following prototype:

```
void setHolidayCalendar (const std::string &strHolidayCalendar) const  
strHolidayCalendar
```

Specifies the name of the holiday calendar.

setWeekendAction

The setWeekendAction method in the InsertExtendedCalendarDaysReq class specifies the action to take when the work falls on a non-workday. This method has the following prototype:

```
void setWeekendAction (eCalendarAction eWeekendAction) const
```

eWeekendAction

Specifies the action to take when the work falls on a non-workday. Valid actions are:

- CaDefault
- CaOnly
- CaNextDay
- CaNextWorkDay
- CaPreviousWorkDay

setWorkdays

The setWorkdays method in the InsertExtendedCalendarDaysReq class sets the condition keyword string. This method has the following prototype:

```
void setWorkdays (const std::string &strWorkdays) const
```

strWorkdays

Specifies the requested workdays. For example:

- To select Monday through Friday, pass the string XXXXX__ to setWorkdays.
- To select Monday, Wednesday, and Friday, pass the string X_X_X__ to setWorkdays.
- To select Monday and Friday, pass the string X__X__ to setWorkdays.

InsertGbloc

InsertGbloc inserts global blobs into the CA Workload Automation AE database.
InsertGbloc is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertGbloc API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertGblocReq req;
    req.setName("MyGbloc");
    const int BLOB_SIZE = 512;
    char p[BLOB_SIZE];
    for (int i = 0; i < BLOB_SIZE; ++i) { // Initialize memory
        p[i] = 'Z';
    }
    req.setGbloc((void *)p, BLOB_SIZE);
    InsertGblocRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertGblobReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the InsertGblobReq class performs the API call. This method has the following prototype:

void execute (AsApi &*api*, SimpleRsp &*rsp*)

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setGblob

The setGblob method in the InsertGblobReq class specifies the Gblob to insert. This method has the following prototype:

void setGblob (void **blob*, int *iBytes*)

blob

Specifies the Global Binary Large OBject (BLOB) to insert.

iBytes

Specifies the size (in bytes) of the global blob.

setName

The setName method in the InsertGblobReq class specifies the name of the Gblob. This method has the following prototype:

void setName (const std::string &*strGblobName*)

strGblobName

Specifies the name of the global blob.

InsertJblob

InsertJblob ties a binary large object to an existing CA Workload Automation AE job.
InsertJblob is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertJblob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertJblobReq req;
    req.setName("MyJob");
    const int BLOB_SIZE = 512;
    char p[BLOB_SIZE];
    for (int i = 0; i < BLOB_SIZE; ++i) { // Initialize memory
        p[i] = 'Z';
    }
    req.setJblob((void *)p, BLOB_SIZE);
    InsertJblobRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertJblobReq

extends

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the InsertJblobReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, InsertJblobRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an InsertJblobRsp object.

Returns

- RS_SUCCESS
- RS_FAILURE
- RS_NO_ATTRIBUTES
- RS_BAD_ATTRIBUTES

The execute status is returned in the inherited response method status().

setName

The setName method in the InsertJblobReq class specifies the job name to which the blob is tied. This method has the following prototype:

```
void setName (const std::string &strJobName)
```

strJobName

Specifies the job name to which the blob is tied.

setJblob

The setJblob method in the InsertJblobReq class specifies the Jblob. This method has the following prototype:

```
void setJblob (void *blob, int iBytes)
```

blob

Specifies the Binary Large Object (BLOB) to tie to the job.

iBytes

Specifies the size (in bytes) of the blob.

setSyntaxChecker

The setSyntaxChecker method in the InsertJblobReq class verifies the Jblob definition. The Jblob is not written to the database. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck)
```

setType

The setType method in the InsertJblobReq class specifies the type of the Jblob. This method has the following prototype:

```
bool setType(const eJobBlobType eType)
```

eType

Specifies the type of Jblob. Valid values are the following:

- JbStdin—stdin
- JbStdout—stdout
- jbStderr—stderr

Default: JbStdin

InsertJblobRsp**extends**

ApiResponse

header

AsPublicSDK.h

getNumberOfBlobVersions

The getNumberOfBlobVersions method in the InsertJblobRsp class returns the number of versions of the input blob. This method has the following prototype:

```
int getNumberOfBlobVersions ()
```

InsertJob

InsertJob creates a job in the database.

The InsertJob performs external job validation, then verifies security permissions and verifies that the caller has create permissions. The caller must also have execute permissions on the specified machine, the specified owner, the run calendar, and the exclude calendar.

InsertJob follows security checks with inter-field verification (for example, if you specify that the job is a command job, then you need to have a command field). It then inserts the job and issues future STARTJOB events if required. Finally, the API updates audit information.

InsertJob is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertJob API

```
try {
    AsApi api;
    api.openRemote(waae_port, waae_server);
    InsertJobReq req;
    req.set(KW_JOB_NAME, (std::string) "TestJob");
    req.set(KW_JOB_TYPE, (int) TYPES_CMD);
    req.set(KW_ALARM_IF_FAIL, true);
    req.set(KW_AUTO_DELETE, 5);
    req.set(KW_AUTO_HOLD, false);
    req.set(KW_COMMAND, (std::string) "SAstatusreport");
    req.set(KW_DESCRIPTION, (std::string) "Reporting after completion of all
processing.");
    req.set(KW_MACHINE, (std::string) "mozart");
    req.set(KW_N_RETRY, 2);
    InsertJobRsp rsp;
    req.execute(api, rsp);
    if(rsp.status() == RS_SUCCESS) {
        std::cout << "TestJob has been inserted." << std::endl;
    }
    else {
        std::cout << "TestJob has not been inserted." << std::endl;
        std::cout << rsp.info() << std::endl;
    }
}
```

```
        }
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    catch(ApiTimeout&) {
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(CredsNotAuthorized&) {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&) {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    api.close();
}
```

InsertJobReq

extends

ApiRequest

header

AsPublicSDK.h

add

The add method in the InsertJobReq class adds a [JobItem](#) (see page 787). This method has the following prototype:

void add (JobItem * *job_item*)

job_item

Specifies a job item object such as sap_recipients or sap_step_parms.

execute

The execute method in the InsertJobReq class performs the API call. This method has the following prototype:

```
execute (AsApi & api, InsertJobRsp & rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an InsertJobRsp object.

Returns:

- RS_SUCCESS
- RS_FAILURE
- RS_ACCESS_DENIED
- RS_EXTERNAL_VALIDATION_FAILURE
- RS_FIELD_VALIDATION_FAILURE

set

This set method in the InsertJobReq class sets the Boolean value for the specified keyword. This method has the following prototype:

```
voidset (eKeywords keyword, bool value)
```

keyword

Specifies a keyword that represents a Boolean job property.

Example: set(KW_AUTO_HOLD, false);

value

Specifies a Boolean value for this keyword.

set

This set method in the InsertJobReq class sets the double value for the specified keyword. This method has the following prototype:

```
voidset (eKeywords keyword, double value)
```

keyword

Specifies a keyword that represents a job property of type double.

value

Specifies the value for this keyword.

set

This set method in the InsertJobReq class sets the value for a keyword that represents a BLOB type. This method has the following prototype:

```
voidset (eKeywords keyword, void * value, int length)
```

keyword

Specifies a keyword that represents a valid BLOB job property.

value

Specifies the value for this keyword.

length

Specifies the length of the BLOB.

set

This set method in the InsertJobReq class sets the integer value for the specified keyword. This method has the following prototype:

```
voidset (eKeywords keyword, int value)
```

keyword

Specifies a keyword that represents an integer job property.

Example: set(KW_N_RETRYS, 2);

value

Specifies the integer value for this keyword.

set

This set method in the InsertJobReq class sets the string value for the specified keyword. This method has the following prototype:

```
voidset (eKeywords keyword, const std::string & value)
```

keyword

Specifies a keyword that represents a string job property.

Example: set(KW_JOB_NAME, (std::string) "TestJob");

value

Specifies the string value for this keyword.

setSyntaxChecker

The setSyntaxChecker method in the InsertJobReq class checks the syntax of the job definition. This method has the following prototype:

```
voidsetSyntaxChecker (bool verify)
```

verify

Specifies whether to check the syntax of the job definition. Possible values are:

TRUE

Checks the syntax of the job definition.

Note: The job definition is **not** written to the database.

FALSE

Performs the InsertJob.

Note: The job definition is written to the database.

setVerifyConditions

The setVerifyConditions method in the InsertJobReq class checks for missing conditions. This method has the following prototype:

```
voidsetVerifyConditions (bool verify)
```

verify

Specifies whether to check for missing conditions. Possible values are:

TRUE

Checks for missing conditions.

FALSE

Do not check for missing conditions.

InsertJobRsp

extends

ApiResponse

header

AsPublicSDK.h

getUndefinedConditionJob

The `getUndefinedConditionJob` method in the `InsertJobRsp` class returns an undefined condition string job. This method has the following prototype:

```
std::string getUndefinedConditionJob (int index) const  
index
```

Specifies the index of an undefined job in the list.

getUndefinedConditionJobsCount

The `getUndefinedConditionJobsCount` method in the `InsertJobRsp` class returns the number of jobs listed in the conditions that are not yet defined. This method has the following prototype:

```
int getUndefinedConditionJobsCount (void ) const
```

InsertRealMachine

`InsertRealMachine` creates/updates a real machine. This API is used by jil to create/update real machines.

The `InsertRealMachine` API verifies that the caller has create access to the named machine. It then inserts the machine and updates audit information.

`InsertRealMachine` is a CAT1 API.

Executing this API can result in the following exceptions:

- `ApiExceptionWithInfo`
- `ApiTimeout`
- `CredsNotAuthorized`
- `FailedToSetCreds`
- `InternalResponseError`

Example: Sample Code for Executing the InsertRealMachine API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertRealMachineReq req;
    req.setName("MyMachine");
    req.setType(MR_NT);
    req.setMaxLoad(25);
    req.setFactor(1.0);
    InsertRealMachineRsp rsp;
    req.execute(api, rsp);
    std::cout << "MyMachine has been inserted." << endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertRealMachineReq**extends**

ApiRequest

header

AsPublicSDK.h

execute

The execute method in the InsertRealMachineReq class performs the API call. This method has the following prototype:

```
execute (AsApi & api, InsertRealMachineRsp & rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an InsertRealMachineRsp object.

setAdministrator

The setAdministrator method in the InsertRealMachineReq class specifies the administrator user ID. This method has the following prototype:

```
void setAdministrator (const std::string & strAdministrator)
```

strAdministrator

Specifies the administrator user ID to use for provisioning the CA Workload Automation AE agent through DCA.

setAgentName

Valid for CA Workload Automation AE r11.3 only

The setAgentName method in the InsertRealMachineReq class specifies the agent name. This method has the following prototype:

```
void setAgentName (const std::string & strAgentName)
```

strAgentName

Specifies the agent name.

Note: Encryption settings must be identical for all machines defined with the same agent_name (for example, encryption_type and key_from_agent values).

setAgentOS

Valid for CA Workload Automation AE r11.3 and machine type MR_EWAA only

The setAgentOS method in the InsertRealMachineReq class specifies the operating system for the agent. This method has the following prototype:

```
void setAgentOS (eAgentOSType eAgentOS)
```

eAgentOS

Specifies the encryption type for the CA Workload Automation AE r11.3 system agent. The following are the valid enumerated agent OS types:

- AO_aix
- AO_hpx
- AO_linux
- AO_openvms
- AO_os400
- AO_solaris
- AO_tandem
- AO_windows
- AO_zos

setCharacterCode

Valid for CA Workload Automation AE r11.3 only

The setCharacterCode method in the InsertRealMachineReq class specifies the character code as either ASCII or EBCDIC. This method has the following prototype:

```
void setCharacterCode (eCharacterCode eCharCode)
```

eCharCode

Specifies the character code. The following are valid character codes:

- CC_ASCII
- CC_EBCDIC

setDescription

The `setDescription` method in the `InsertRealMachineReq` class sets the description of the machine. This method has the following prototype:

```
void setDescription (const std::string & strDescription)
```

strDescription

Specifies a description of the agent.

setEncryptionType

Valid for CA Workload Automation AE r11.3 and machine type MR_EWAA only

The `setEncryptionType` method in the `InsertRealMachineReq` class specifies the encryption type for the system agent. This method has the following prototype:

```
void setEncryptionType (eEncryptionType eEncryptType)
```

eEncryptType

Specifies the encryption type for the system agent. The following are the valid enumerated encryption types:

- `ET_NONE` - encryption disabled
- `ET_DEFAULT` - default encryption key and type
- `ET_AES` - AES 128-bit encryption

setFactor

The `setFactor` method in the `InsertRealMachineReq` class specifies the factor of the machine. This method has the following prototype:

```
void setFactor (double dFactor)
```

dFactor

Specifies a factor to multiply by a real machine's available CPU cycles to determine the relative available CPU cycles. When more than one real machine or a virtual machine is specified in the job's machine attribute, the `real_number` value determines on which machine a job should run. The factor value indicates a machine's relative processing power. The factor only applies to CA Workload Automation AEagents.

setHeartbeatAttempts

Valid for CA Workload Automation AE r11.3 only

The setHeartbeatAttempts method in the InsertRealMachineReq class specifies the heartbeat attempts. This method has the following prototype:

```
void setHeartbeatAttempts (int iHeartbeatAttempts)
```

iHeartbeatAttempts

Specifies the number of heartbeat signals the scheduler attempts before it sends an SNMP message indicating CA Workload Automation AE r11.3 inactivity.

setHeartbeatFrequency

The setHeartbeatFrequency method in the InsertRealMachineReq class specifies the heartbeat frequency. This method has the following prototype:

```
void setHeartbeatFrequency (int iHeartbeatFrequency)
```

iHeartbeatFrequency

Defines the frequency with which you want the scheduler to send the heartbeat signal. This parameter is valid only for CA Workload Automation AE r11.3.

setInsertMachine

Note: This method has been deprecated.

The setInsertMachine method in the InsertRealMachineReq class specifies whether this is an INSERT_MACHINE or UPDATE_MACHINE API call. This method has the following prototype:

```
void setInsertMachine (bool bInsert)
```

bInsert

Specifies the type of call. Possible values are:

TRUE

Performs an INSERT_MACHINE.

FALSE

Performs an UPDATE_MACHINE.

setKeyFromAgent

Valid for CA Workload Automation AE r11.3 only

Valid for a Cybernation agent on z/OS only

The setKeyFromAgent method in the InsertRealMachineReq class specifies the key from agent value. This method has the following prototype:

```
void setKeyFromAgent (const std::string & strKeyFromAgent)
```

strKeyFromAgent

Specifies the key-from-agent value. This parameter is valid only if encryption_type is specified as a value other than none or default.

setKeyToAgent

Valid for CA Workload Automation AE r11.3 only

The setKeyToAgent method in the InsertRealMachineReq class specifies the key to agent value. This method has the following prototype:

```
void setKeyToAgent (const std::string & strKeyToAgent)
```

strKeyToAgent

Specifies the key-to-agent value. This value must match the security.cryptkey parameter on the CA Workload Automation AE agent. If they do not match, the encryption fails and communication is not allowed to the CA Workload Automation AE agent.

Limits: 16 characters or 32 hexadecimal digits

setMaxLoad

Valid on CA Workload Automation AE agents only

The setMaxLoad method in the InsertRealMachineReq class specifies the maximum load of the machine. This method has the following prototype:

```
void setMaxLoad (int iMaxLoad)
```

iMaxLoad

Specifies the maximum load (in load units) that a machine can handle.

setName

The setName method in the InsertRealMachineReq class specifies the name of the machine. This method has the following prototype:

```
void setName (const std::string &strMachineName)
```

strMachineName

Specifies the machine name.

setNodeName

The setNodeName method in the InsertRealMachineReq class specifies the node name. This method has the following prototype:

```
void setNodeName (const std::string & strnodeName)
```

strnodeName

Specifies the IP address or DNS name of the machine. If strnodeName is not specified, the machine name is used for the node name.

setPortNumber

The setPortNumber method in the InsertRealMachineReq class specifies the port number. This method has the following prototype:

```
void setPortNumber (int iPortNumber)
```

iPortNumber

Specifies the IP port number CA Workload Automation AE uses to listen for traffic.

setProvision

The setProvision method in the InsertRealMachineReq class specifies whether to provision an CA Workload Automation AE agent. This method has the following prototype:

```
void setProvision (bool bProvision)
```

bProvision

Indicates, if TRUE, that the Data Center Automation (DCA) Manager must be requested to provision an CA Workload Automation AE agent on this machine.

setSyntaxChecker

The `setSyntaxChecker` method in the `InsertRealMachineReq` class verifies the machine definition. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck);
```

bCheck

Specifies whether to verify the machine definition. Possible values are:

TRUE

Checks the machine definition.

FALSE

Performs the job.

setType

The setType method in the InsertRealMachineReq class specifies the type of machine. This method has the following prototype:

```
void setType (eMachineType eType)
```

eType

Specifies the type of machine. Possible values are:

MR_NT

Specifies a Windows machine.

MR_Unix

Specifies a UNIX machine.

MR_UnixVirtual

Specifies a virtual UNIX machine.

MR_WindowsVirtual

Specifies a virtual machine consisting of ONLY Windows real machines.

MR_Connect

Specifies CA Workload Automation AE Connect.

MR_UUJMA

Specifies Universal Job Management Agent.

MR_UnixLegacy

Specifies a legacy UNIX machine.

MR_NTLegacy

Specifies a legacy NT machine.

MR_EWAA

Specifies CA Workload Automation AE System Agent.

MR_POOL

Specifies a pool of DCA managed machines.

InsertResource

InsertResource inserts a virtual resource into the CA Workload Automation AE database.
InsertResource is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertResource API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertResourceReq req;
    req.setName("MyResource");
    req.setType(VRT_Depletable);
    req.setDescription("Description of MyResource");
    req.setMachine("mach01");
    req.setAmount(20);
    InsertResourceRsp rsp;
    req.execute(api, rsp);
    std::cout << "MyResource has been inserted." << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertResourceReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the InsertResourceReq class performs the API call. This method has the following prototype:

`execute(AsApi & api, InsertResourceRsp & rsp)`

api

Specifies an AsApi object.

rsp

Specifies an InsertResourceRsp object.

setName

The setName method in the InsertResourceReq class specifies the resource name. This method has the following prototype:

`void setName(std::string str resourceName)`

str resourceName

Specifies the resource name.

setAmount

The setAmount method in the InsertResourceReq class specifies the amount of resources. This method has the following prototype:

`void setAmount(int iAmount)`

iAmount

Specifies the resource amount.

setMachine

The setMachine method in the InsertResourceReq class specifies a resource machine. This method has the following prototype:

```
void setMachine( std::string strMachineName)
```

strMachineName

Specifies the resource machine name.

setDescription

The setDescription method in the InsertResourceReq class specifies a description for the resource. This method has the following prototype:

```
void setDescription(const std::string & strDescription)
```

strDescription

Specifies the resource description.

setType

The setType method in the InsertResourceReq class specifies the resource type. This method has the following prototype:

```
void setType(eResourceType eType)
```

eType

Specifies one of the following resource types:

- VRT_Depletable
- VRT_Renewable
- VRT_Threshold

InsertResourceRsp

extends

ApiResponse

header

AsPublicSDK.h

InsertUpdateDeleteJobType

InsertUpdateDeleteJobType inserts, updates, and deletes job types in the CA Workload Automation AE database. InsertUpdateDeleteJobType is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertUpdateDeleteJobType API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertUpdateDeleteJobTypeReq req;
    req.setInsertJob(JtInsert);
    req.setJobType("1");
    req.setCommand("Notepad.exe");
    req.setDescription("Text Editor.");
    SimpleRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertUpdateDeleteJobTypeReq

extends
 ApiRequest
header
 AsPublicSDK.h

InsertUpdateDeleteJobTypeReq

This constructor has the following syntax:

```
InsertUpdateDeleteJobTypeReq(const InsertUpdateDeleteJobTypeReq & arg)
```

execute

The execute method in the InsertUpdateDeleteJobTypeReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, SimpleRsp &rsp)  
api
```

Specifies an AsApi object.

```
rsp  
Specifies a SimpleRsp object.
```

setCommand

The setCommand method in the InsertUpdateDeleteJobTypeReq class sets the command associated with the job type. This method has the following prototype:

```
void setCommand (const std::string &strCommand) const  
strCommand
```

Specifies the command associated with the job type.

setDescription

The setDescription method in the InsertUpdateDeleteJobTypeReq class sets the description associated with the job type. This method has the following prototype:

```
void setDescription (const std::string &strDescription) const  
strDescription
```

Specifies the description associated with the job type.

setInsertJob

The setInsertJob method in the InsertUpdateDeleteJobTypeReq class sets the operation type to insert. This method has the following prototype:

```
void setInsertJob (eJobTypeOperation eInsertType) const  
eInsertType
```

Specifies the operation type. Possible values are:

- JtInsert
- JtUpdate
- JtDelete

setJobType

The setJobType method in the InsertUpdateDeleteJobTypeReq class sets the job type to insert, update, or delete. This method has the following prototype:

```
void setJobType (const std::string &strType) const  
strType
```

Specifies the job type to insert.

Limits: 0 - 9

setSyntaxChecker

The setSyntaxChecker method in the InsertUpdateDeleteJobTypeReq class verifies the validity of the job definition. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck)
```

bCheck

Specifies whether to verify the validity of the job definition. Possible values are:

TRUE

Turns this option on.

FALSE

Turns this option off.

Note: The job definition is not written to the database.

InsertUpdateExternalInstance

InsertUpdateExternalInstance inserts or updates external instance entries in the CA Workload Automation AE database. InsertUpdateExternalInstance is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertUpdateExternalInstance API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertUpdateExternalInstanceReq req;
    req.setInsertInstance(TRUE);
    req.setInstanceId("TST");
    req.setInstanceType(XI_AppServer);
    req.setInstanceMachine("machineA");
    SimpleRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertUpdateExternalInstanceReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the InsertUpdateExternalInstanceReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, InsertUpdateExternalInstanceRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies an InsertUpdateExternalInstanceRsp object.

setCryptKey

The setCryptKey method in the InsertUpdateExternalInstanceReq class specifies an encryption key. This method has the following prototype:

`void setCryptKey(const std::string & strCryptKey) const`

strCryptKey

Specifies the string containing the xcrypt_key attribute.

setCryptType

The setCryptType method in the InsertUpdateExternalInstanceReq class specifies an encryption type. This method has the following prototype:

`void setCryptType(eEncryptType iCryptType) const`

iCryptType

Specifies one of the following encryption types:

- XI_None
- XI_Default
- XI_AES

setInsertInstance

The `setInsertInstance` method in the `InsertUpdateExternalInstanceReq` class specifies whether this is an insert or update external instance API call. This method has the following prototype:

```
void setInsertInstance (bool bInsert) const
```

bInsert

Specifies whether to insert or update the external instance. Possible values are:

TRUE

Performs an insert.

FALSE

Performs an update.

setInstanceId

The `setInstanceId` method in the `InsertUpdateExternalInstanceReq` class specifies the external instance ID to insert or update. This method has the following prototype:

```
void setInstanceId (const std::string &strInstanceId) const
```

strInstanceId

Specifies the ID of the external instance to insert, for example, ACE.

setInstanceMachine

The `setInstanceMachine` method in the `InsertUpdateExternalInstanceReq` class specifies an instance machine. This method has the following prototype:

```
void setInstanceMachine (const std::string &strInstanceMachine) const
```

strInstanceMachine

Specifies the string containing xmachine attribute entries, for example, machineA:9000 (Application Server Instance) or machineA (CA Workload Automation AE Connect or Unicenter Workload Instance).

setInstanceType

The `setInstanceType` method in the `InsertUpdateExternalInstanceReq` class specifies the instance type. This method has the following prototype:

```
void setInstanceType (eInstanceType eType) const  
eType
```

Specifies the type of the external instance. Valid instance types are:

- `XI_AppServer`
- `XI_Connect`
- `XI_Unicenter`

setManager

The `setManager` method in the `InsertUpdateExternalInstanceReq` class specifies an instance manager. This method has the following prototype:

```
void setManager(const std::string & strManager) const  
strManager
```

Specifies the string containing the `xmanager` attribute.

setPort

The `setPort` method in the `InsertUpdateExternalInstanceReq` class specifies the external instance port number. This method has the following prototype:

```
void setPort(int iPort) const  
iPort
```

Specifies the external instance port number.

setSyntaxChecker

The `setSyntaxChecker` method in the `InsertUpdateExternalInstanceReq` class verifies the external instance definition. The definition is not written to the database. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck) const
```

InsertUpdateJob

The InsertUpdateJob API creates or updates a job in the database. This API is used by jil for insert_job and update_job.

The InsertUpdateJob performs external job validation (if in place), then verifies security permissions. If the API is being used to update a job, it verifies that the caller has write permissions for the job. If the API is being used to insert a new job, it verifies that the caller has create permissions. The caller must also have execute permissions on the specified machine, the specified owner, the run calendar, and the exclude calendar.

InsertUpdateJob follows security checks with inter-field verification (for example, if you specify that the job is a command job, then you need to have a command field). It then inserts the job and issues future STARTJOB events if required. Finally, the API updates audit information.

InsertUpdateJob is a CAT1 API.

Example: Sample Code for Executing the InsertUpdateJob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertUpdateJobReq req;
    req.setInsertJob(TRUE);
    req.setName("MyJob");
    req.setMachine("localhost");
    req.setCommand("ls");
    InsertUpdateJobRsp rsp;
    req.execute(api, rsp);

}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
```

InsertUpdateJobReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the InsertUpdateJobReq class performs the API call. This method has the following prototype:

`execute (AsApi & api, InsertUpdateJobRsp & rsp)`

api

Specifies an AsApi object.

rsp

Specifies an InsertUpdateJobRsp object.

setInsertJob

The setInsertJob method in the InsertUpdateJobReq class specifies whether this is an INSERT_JOB or UPDATE_JOB call. This method has the following prototype:

`void setInsertJob(bool bInsert)`

bInsert

Performs an INSERT_JOB if the value is TRUE or an UPDATE_JOB if the value is FALSE.

setVerifyConditions

The setVerifyConditions method in the InsertUpdateJobReq class checks for missing conditions. This method has the following prototype:

`void setVerifyConditions(bool verify)`

verify

Specifies whether to check for missing conditions. Possible values are:

TRUE

Checks for missing conditions.

FALSE

Do not check for missing conditions.

[setSyntaxChecker](#)

The `setSyntaxChecker` method in the `InsertUpdateJobReq` class verifies the validity of job definitions. This method has the following prototype:

```
void setSyntaxChecker(bool bCheck)
```

bCheck

Specifies whether to verify the validity of the job definition. Possible values are:

TRUE

Turns this option on.

FALSE

Turns this option off.

[setAlarmIfFail](#)

The `setAlarmIfFail` method in the `InsertUpdateJobReq` class indicates whether an alarm should be posted to the Event processor if the job fails or is terminated. This method has the following prototype:

```
void setAlarmIfFail (bool bAlarmIfFail)
```

bAlarmIfFail

Turns this property on if TRUE and off if FALSE.

[setApplication](#)

The `setApplication` method in the `InsertUpdateJobReq` class indicates the job's associated application. This method has the following prototype:

```
void setApplication(const std::string & strApplication)
```

strApplication

Indicates the job's associated application.

setAutoDelete

The setAutoDelete method in the InsertUpdateJobReq class indicates whether the job should be automatically deleted after completion. This method has the following prototype:

```
void setAutoDelete(int iAutoDelete)
```

iAutoDelete

Defines the interval (in hours) after the job finishes at which CA Workload Automation AE should automatically delete the job.

- Set this value to 0 to delete the job immediately after completion.
- Set this value to -1 if you do not want to automatically delete the job.

setAutoHold

The setAutoHold method in the InsertUpdateJobReq class indicates whether to change the job state to ON_HOLD when the box it is in begins RUNNING. This method has the following prototype:

```
void setAutoHold(bool bAutoHold)
```

bAutoHold

Turns this property on if TRUE and off if FALSE.

setAvgRunTime

The setAvgRunTime method in the InsertUpdateJobReq class indicates an average runtime (in minutes) for a job that is newly submitted to the database. This method has the following prototype:

```
void setAvgRunTime(int iAvgRunTime)
```

iAvgRunTime

Indicates the average runtime in minutes.

setBoxFailure

The setBoxFailure method in the InsertUpdateJobReq class specifies the conditions to be interpreted as a box failure. This method has the following prototype:

```
void setBoxFailure(const std::string & strBoxFailure)
```

strBoxFailure

Specifies the conditions to be interpreted as a box failure.

setBoxName

The setBoxName method in the InsertUpdateJobReq class specifies the name of the box to which this job belongs. This method has the following prototype:

```
void setBoxName(const std::string & strBoxName)
```

strBoxName

Specifies the box name.

setBoxSuccess

The setBoxSuccess method in the InsertUpdateJobReq class specifies the conditions to be interpreted as a box success. This method has the following prototype:

```
void setBoxSuccess(const std::string & strBoxSuccess)
```

strBoxFailure

Specifies the conditions to be interpreted as a box success.

setBoxTerminator

The setBoxTerminator method in the InsertUpdateJobReq class indicates whether the box containing this job should be terminated if the job fails or terminates. This method has the following prototype:

```
void setBoxTerminator(bool bBoxTerminator)
```

bBoxTerminator

Turns this property on if TRUE, and off if FALSE.

setChkFiles

The setChkFiles method in the InsertUpdateJobReq class specifies the minimum amount of file space that must be available on designated file systems for the job to be started. This method has the following prototype:

```
void setChkFiles(const std::string & strChkFiles)
```

strChkFiles

Sets the minimum file space that must be available on the designated file systems for the job to start.

setCommand

The `setCommand` method in the `InsertUpdateJobReq` class specifies the job command. This method has the following prototype:

```
void setCommand(const std::string & strCommand)
```

strCommand

Specifies the job command.

setCondition

The `setCondition` method in the `InsertUpdateJobReq` class specifies the conditions for job to run, for example, "success(DB_BACKUP)", "exitcode (my_job=4)", "VALUE(TODAY)=Friday". This method has the following prototype:

```
void setCondition(const std::string & strCondition)
```

strCondition

Specifies the conditions for a job to run.

setCycleCalendar

The `setCycleCalendar` method in the `InsertUpdateJobReq` class sets the cycle calendar name. This method has the following prototype:

```
void setCycleCalendar(const std::string & strCycleCalendar)
```

strCycleCalendar

Sets the cycle calendar name.

setDateAdjust

The `setDateAdjust` method in the `InsertUpdateJobReq` class sets the date adjustment. This method has the following prototype:

```
void setDateAdjust (int iDateAdjust)
```

iDateAdjust

Sets the date adjustment.

setDateConditions

The setDateConditions method in the InsertUpdateJobReq class indicates whether there are date or time conditions for starting this job. This method has the following prototype:

```
void setDateConditions (bool bDateConditions)
```

bDateConditions

Specify true when using date or time conditions.

setDateKeywords

The setDateKeywords method in the InsertUpdateJobReq class sets the date keywords. This method has the following prototype:

```
void setDateKeywords (const std::string & strDateKeywords)
```

strDateKeywords

Sets the date keywords.

setDaysOfWeek

The setDaysOfWeek method in the InsertUpdateJobReq class indicates the days of the week when the job will be run, for example, "mo,tu,we,fr". This method has the following prototype:

```
void setDaysOfWeek (const std::string & strDaysOfWeek)
```

strDaysOfWeek

Indicates the days of the week when the job will be run.

setDescription

The setDescription method in the InsertUpdateJobReq class provides a description of the job. This method has the following prototype:

```
void setDescription (const std::string & strDescription)
```

strDescription

Provides a description of the job.

setExcludeCalendar

The `setExcludeCalendar` method in the `InsertUpdateJobReq` class indicates the name of the custom calendar to be used for determining the days of the week on which this job will not run. This method has the following prototype:

```
void setExcludeCalendar (const std::string & strExcludeCalendar)  
strExcludeCalendar
```

Specifies the name of the custom exclude calendar.

setGroup

The `setGroup` method in the `InsertUpdateJobReq` class indicates the job's associated group. This method has the following prototype:

```
void setGroup (const std::string & strGroup)  
strGroup
```

Specifies the name of the job's associated group.

setHeartbeatInterval

The `setHeartbeatInterval` method in the `InsertUpdateJobReq` class specifies the frequency (in minutes) at which this job's command is expected to issue a heart beat. This method has the following prototype:

```
void setHeartbeatInterval (int iHeartbeatInterval)  
iHeartbeatInterval
```

Specifies the heartbeat frequency interval (in minutes).

setHolidayAction

The `setHolidayAction` method in the `InsertUpdateJobReq` class specifies the action to take when the work falls on a holiday. This method has the following prototype:

```
void setHolidayAction (eCalendarAction eHolidayAction)  
eHolidayAction
```

Indicates one of the following holiday actions:

- `CaDefault`
- `CaOnly`
- `CaSchedule`
- `CaNextDay`
- `CaNextWorkDay`
- `CaPreviousWorkDay`

setHolidayCalendar

The `setHolidayCalendar` method in the `InsertUpdateJobReq` class specifies the holiday calendar to use. This method has the following prototype:

```
void setHolidayCalendar (const std::string & strHolidayCalendar)  
strHolidayCalendar
```

Specifies the name of the holiday calendar.

setJobLoad

The `setJobLoad` method in the `InsertUpdateJobReq` class specifies the relative amount of processing power the job will consume. This method has the following prototype:

```
void setJobLoad (int iJobLoad)  
iJobLoad
```

Specifies the relative amount of processing power the job will consume.

setJobTerminator

The `setJobTerminator` method in the `InsertUpdateJobReq` class specifies whether to use a KILLJOB event to terminate the job if its containing box job fails or is terminated. This method has the following prototype:

```
void setJobTerminator (bool bJobTerminator)  
bJobTerminator
```

Specify true to activate job termination.

setMachine

The setMachine method in the InsertUpdateJobReq class specifies the machine where job is performed. This method has the following prototype:

```
void setMachine (const std::string & strMachine)
```

strMachine

Specifies the machine name.

setMaxExitSuccess

The setMaxExitSuccess method in the InsertUpdateJobReq class specifies the maximum exit code with which the job can exit and still be considered a success. This method has the following prototype:

```
void setMaxExitSuccess (int iMaxExitSuccess)
```

iMaxExitSuccess

Specifies the maximum exit code with which the job can exit and still be considered a success.

setMaxRunAlarm

The setMaxRunAlarm method in the InsertUpdateJobReq class specifies the maximum runtime (in minutes) that a job should require to finish normally. This method has the following prototype:

```
void setMaxRunAlarm (int iMaxRunAlarm)
```

iMaxRunAlarm

Specifies the maximum run time alarm value.

setMinRunAlarm

The setMinRunAlarm method in the InsertUpdateJobReq class specifies the minimum runtime (in minutes) that a job should require to finish normally. This method has the following prototype:

```
void setMinRunAlarm (int iMinRunAlarm)
```

iMinRunAlarm

Specifies the minimum run time alarm value.

setName

The setName method in the InsertUpdateJobReq class specifies the job name. This method has the following prototype:

```
void setName (const std::string & strJobName)
```

strJobName

Specifies the job name.

setWeekendAction

The setWeekendAction method in the InsertUpdateJobReq class specifies the action to take when the work falls on a non-workday. This method has the following prototype:

```
void setWeekendAction (eCalendarAction eAction)
```

eAction

Specifies one of the following as weekend actions:

- CaDefault
- CaOnly
- CaNextDay
- CaNextWorkDay
- CaPreviousWorkDay

setRetries

The setRetries method in the InsertUpdateJobReq class specifies how many times, if any, the job should be restarted after exiting with a FAILURE status. This method has the following prototype:

```
void setRetries (int iRetries)
```

iRetries

Specifies the number of times a job should be restarted after exiting with a FAILURE status.

setOwner

The setOwner method in the InsertUpdateJobReq class specifies the job owner. This method has the following prototype:

```
void setOwner (const std::string & strOwner)
```

strOwner

Specifies the job owner.

setPermission

The `setPermission` method in the `InsertUpdateJobReq` class sets the job permissions. This method has the following prototype:

```
void setPermission (const std::string & strPermission)
```

strPermission

Specifies the job permission.

setPriority

The `setPriority` method in the `InsertUpdateJobReq` class specifies the queue priority of the job. This method has the following prototype:

```
void setPriority (int iPriority)
```

iPriority

Specifies the job queue priority.

setProfile

The `setProfile` method in the `InsertUpdateJobReq` class specifies a job profile that defines environment variables to be set before the specified command is executed. This method has the following prototype:

```
void setProfile (const std::string & strProfile)
```

strProfile

Specifies the job profile.

setRunCalendar

The `setRunCalendar` method in the `InsertUpdateJobReq` class indicates the name of the custom calendar to be used when determining the days of the week on which a job will run. This method has the following prototype:

```
void setRunCalendar (const std::string & strRunCalendar)
```

strRunCalendar

Specifies the run calendar.

setRunWindow

The `setRunWindow` method in the `InsertUpdateJobReq` class indicates the time span during which the job will be allowed to start. If this attribute is specified, it implies the job is eligible to run only during this time interval. This method has the following prototype:

```
void setRunWindow (const std::string &strRunWindow)
```

strRunWindow

Specifies the run window.

setStartMins

The `setStartMins` method in the `InsertUpdateJobReq` class indicates the number of minutes past the hour, every hour, on the specified days or dates, when the job will be started, for example, "20,40". This method has the following prototype:

```
void setStartMins (const std::string & strStartMins)
```

strStartMins

Indicates the number of minutes past the hour every hour, on the specified days or dates, when the job will be started.

setStartTimes

The `setStartTimes` method in the `InsertUpdateJobReq` class indicates the times of day, in 24-hour format, on the specified days or dates, when the job will be started, for example, "12:00, 13:00, 14:15". This method has the following prototype:

```
void setStartTimes (const std::string & strStartTimes)
```

strStartTimes

Indicates the times of day in which the job will be started.

setStderrFile

The `setStderrFile` method in the `InsertUpdateJobReq` class specifies the file to which the standard error file's output should be redirected. This method has the following prototype:

```
void setStderrFile (const std::string & strStderrFile)
```

strStderrFile

Specifies the standard error output file.

setStdinFile

The `setStdinFile` method in the `InsertUpdateJobReq` class specifies the file to which the standard input file for the job should be redirected. This method has the following prototype:

```
void setStdinFile (const std::string & strStdinFile)
```

strStdinFile

Specifies the standard input file.

setStdoutFile

The `setStdoutFile` method in the `InsertUpdateJobReq` class specifies the file to which the standard output file should be redirected. This method has the following prototype:

```
void setStdoutFile (const std::string & strStdoutFile)
```

strStdoutFile

Specifies the standard output file.

setTermRunTime

The `setTermRunTime` method in the `InsertUpdateJobReq` class specifies the maximum runtime (in minutes) that the job you are defining should require to finish normally. If the job runs longer than the specified time, CA Workload Automation AE terminates it. This method has the following prototype:

```
void setTermRunTime (int iTermRunTime)
```

iTermRunTime

Specifies the maximum runtime (in minutes) that a job should require to finish normally.

setTimezone

The `setTimezone` method in the `InsertUpdateJobReq` class allows you to schedule a job based on a chosen time zone. This method has the following prototype:

```
void setTimezone (const std::string & strTimezone)
```

strTimezone

Specifies the job's time zone.

setType

The setType method in the InsertUpdateJobReq class specifies the job type. This method has the following prototype:

```
void setType (char cType)
```

cType

Specifies the job type.

setWatchFile

The setWatchFile method in the InsertUpdateJobReq class specifies the full path name of the file for which this file watcher job should watch. This method has the following prototype:

```
void setWatchFile (const std::string & strWatchFile)
```

strWatchFile

Specifies the name of the watch file.

setWatchFileMinSize

The setWatchFileMinSize method in the InsertUpdateJobReq class specifies the watch file minimum size (in bytes), which determines when enough data has been written to the file to consider it complete. This method has the following prototype:

```
void setWatchFileMinSize (int iWatchFileMinSize)
```

strWatchFile

Specifies the minimum size (in bytes) of the watch file.

setWatchInterval

The setWatchInterval method in the InsertUpdateJobReq class specifies the interval (in seconds) at which the file watcher job will check for the existence and size of the watched-for file. This method has the following prototype:

```
void setWatchInterval (int iWatchInterval)
```

iWatchInterval

Specifies the watch interval in seconds.

setWorkdays

The `setWorkdays` method in the `InsertUpdateJobReq` class sets the work days string. This method has the following prototype:

```
void setWorkdays (const std::string & strWorkdays)
```

strWorkdays

Specifies the selected workdays. For example:

- To select Monday through Friday, pass the string "XXXX__" to `setWorkdays`.
- To select Monday, Wednesday and Friday, pass the string "X_X_X__" to `setWorkdays`.
- To select Monday and Friday, pass the string "X__X__" to `setWorkdays`.

setSendNotification

The `setSendNotification` method in the `InsertUpdateJobReq` class sends a notification. This method has the following prototype:

```
void setSendNotification (bool bSendNotify)
```

bSendNotify

Specify true to activate send notification.

setNotificationID

The `setNotificationID` method in the `InsertUpdateJobReq` class specifies the notification ID. This method has the following prototype:

```
void setNotificationID (const std::string & strNotificationID)
```

strNotificationID

Specifies the notification ID.

setNotificationMsg

The `setNotificationMsg` method in the `InsertUpdateJobReq` class specifies the notification msg. This method has the following prototype:

```
void setNotificationMsg (const std::string & strNotificationMsg)
```

strNotificationMsg

Specifies the notification msg.

setServiceDesk

The setServiceDesk method in the InsertUpdateJobReq class specifies the service desk flag. This method has the following prototype:

```
void setServiceDesk (bool bServiceDesk)
```

bServiceDesk

Specifies the service deck flag.

setServiceDeskDescription

The setServiceDeskDescription method in the InsertUpdateJobReq class specifies the service desk description. This method has the following prototype:

```
void setServiceDeskDescription (const std::string & strServiceDeskDesc)
```

strServiceDeskDesc

Specifies the service desk description.

setServiceDeskPriority

The setServiceDeskPriority method in the InsertUpdateJobReq class specifies the service desk priority. This method has the following prototype:

```
void setServiceDeskPriority (char cServiceDeskPri)
```

cServiceDeskPri

Specifies the priority level to assign to the service desk request. Valid values are '0' through '5'.

setServiceDeskImpact

The setServiceDeskImpact method in the InsertUpdateJobReq class specifies the service desk impact. This method has the following prototype:

```
void setServiceDeskImpact (char cServiceDeskImp)
```

cServiceDeskImp

Specifies the impact level to assign to the service desk request. Valid values are '0' through '5'.

setServiceDeskSeverity

The setServiceDeskSeverity method in the InsertUpdateJobReq class specifies the service desk severity level. This method has the following prototype:

```
void setServiceDeskSeverity (char cServiceDeskSev)
```

iServiceDeskImp

Specifies the severity level to assign to the service desk request. Valid values are '0' through '5'.

setServiceDeskAttribute

The setServiceDeskAttribute method in the InsertUpdateJobReq class specifies the service desk attribute. This method has the following prototype:

```
void setServiceDeskAttribute (const std::string & strServiceDeskAttr)
```

strServiceDeskAttr

Specifies the service desk attribute.

InsertUpdateJobRsp

extends

ApiResponse

header

AsPublicSDK.h

getUndefinedConditionJob

The getUndefinedConditionJob method in the InsertUpdateJobRsp class returns an undefined job that is listed in the condition string. This method has the following prototype:

```
std::string getUndefinedConditionJob (int i) const
```

i

Specifies the index of the undefined job list.

getUndefinedConditionJobsCount

The getUndefinedConditionJobsCount method in the InsertUpdateJobRsp class returns the number of jobs listed in conditions that are not yet defined. This method has the following prototype:

```
int getUndefinedConditionJobsCount (void) const
```

InsertVirtualMachine

InsertVirtualMachine creates a virtual machine. This API is used by JIL to insert virtual machines.

The InsertVirtualMachine API verifies that the caller has create access to the named machine. It then inserts the machine and updates audit information.

InsertVirtualMachine is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the InsertVirtualMachine API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    InsertVirtualMachineReq req("MyVirtualMachine", MR_NT);
    req.appendMachine("machine1");

    req.appendMachine("machine2");
    InsertVirtualMachineRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

InsertVirtualMachineReq

extends
 ApiRequest
header
 AsPublicSDK.h

InsertVirtualMachineReq

The InsertVirtualMachineReq constructor configures the request. This constructor has the following syntax:

(const std::string & *strMachineName*, eMachineType *eType*)

strMachineName

Specifies the name of the virtual machine.

eType

Specifies one of the following machine types:

- MR_NT—Windows virtual machine
- MR_UNIX—UNIX virtual machine

appendMachine

This AppendMachine method in the InsertVirtualMachineReq class adds a component machine to the virtual machine definition. This method has the following prototype:

void appendMachine (const std::string &*strComponentName*, int *iMaxLoad*, double *dFactor*)

strComponentName

Specifies the component (real) machine name.

iMaxLoad

Specifies the maximum load of the component machine.

dFactor

Specifies the component machine factor.

appendMachine

This appendMachine method in the InsertVirtualMachineReq class adds a component machine to the virtual machine definition. This method has the following prototype:

void appendMachine (const std::string & *strComponentName*)

strComponentName

Specifies the component machine name.

execute

The execute method in the InsertVirtualMachineReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, InsertVirtualMachineRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an InsertVirtualMachineRsp object.

setSyntaxChecker

The setSyntaxChecker method in the InsertVirtualMachineReq class verifies the machine definition. The job definition is not written to database. This method has the following prototype:

```
void setSyntaxChecker (void)
```

InsertVirtualMachineRsp

extends

ApiResponse

header

AsPublicSDK.h

InternalResponseError

The InternalResponseError class is used to relay that there was an internal error with the API response.

InternalUnexpected

The InternalUnexpected class indicates that the server returned a result that was not expected by the client libraries. This is usually caused by a version mismatch between the two.

JobItem

The JobItem class is used to specify repeating job items and retrieve job items from a CA Workload Automation AE job. For example, a SAP job may contain multiple SAP_STEP_PARMS. Each SAP_STEP_PARMS can be represented by a JobItem object.

Example: Sample Code Using the JobItem Class

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    AsApi api;
    api.openRemote(waae_port, waae_server);
    req.set(KW_JOB_NAME, (std::string) "TestJob");
    req.set(KW_JOB_TYPE, TYPES_DBPROC);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_DESCRIPTION, (std::string) "Test dbproc job definition");
    req.set(KW_CONNECT_STRING, (std::string) "jdbc.cnxt_str");
    req.set(KW_USER_ROLE, (std::string) "usr role");
    req.set(KW_SP_NAME, (std::string) "spnam");
    req.set(KW_SUCCESS_CRITERIA, (std::string) "sx_criteria");
    { // Create job item...
        autosys::JobItem jitem(pMetaData);
        jitem.setItemType(KW_SP_ARG);
        jitem.set(KW_SEQ_NO, 1);
        jitem.set(KW_IGNORE, 0);
        jitem.set(KW_NAME, (std::string) "salary");
        jitem.set(KW_ARGLTYPE, (std::string) "IN");
        jitem.set(KW_DATATYPE, (std::string) "VARCHAR");
        jitem.set(KW_VALUE, (std::string) "dbproc value");
        req.add(&jitem);
    }
    req.execute(api, rsp);
    api.close();
}
catch( ApiExceptionWithInfo& e ) {
    std::cout << "Exception thrown in createDbprocJob. Message follows: " <<
    std::endl;
    std::cout << e.what() << std::endl;
}
```

add

The add method in the JobItem class adds a job item to a job. This method has the following prototype:

```
void add (JobItem job_item)  
job_item
```

Specifies the job item to add to a job.

getAttributes

The getAttributes method in the JobItem class gets the count of returned attributes. This method has the following prototype:

```
int getAttributes ()
```

getAttributeName

The getAttributeName method in the JobItem class gets the attribute name. This method has the following prototype:

```
int getAttributeName (int index)
```

getBool

The getBool method in the JobItem class gets a bool value from a job item. This method has the following prototype:

```
int getBool (int kwNumber)  
kwNumber
```

Specifies the enumerated keyword of the job property to be retrieved.

getInt

The getInt method in the JobItem class retrieves an integer value from a job item. This method has the following prototype:

```
int getInt (int kwNumber)
```

kwNumber

Specifies the the enumerated keyword of the job property to be retrieved, for example, KW_N_RETRY..

getJobItem

The getJobItem method in the JobItem class returns a repeating job item. This method has the following prototype:

```
JobItem getJobItem (int item_type)
```

item_type

Specifies the item type.

getItemType

The getItemType method in the JobItem class gets the job item's type, for example, KW_JMX_PARAMETER. This method has the following prototype:

```
int getItemType ()
```

getString

The getString method in the JobItem class returns a string value from a job item. This method has the following prototype:

```
std::string getString (int kwNumber)
```

kwNumber

Specifies the the enumerated keyword of the job property to be retrieved, for example, KW_JOB_NAME.

getVectorOfKeywordConstants

The `getVectorOfKeywordConstants` method in the `JobItem` class gets the vector of keyword constants. This method has the following prototype:

```
std::vector<int> getVectorOfKeywordConstants ()
```

set

This set method in the `JobItem` class sets the integer value for a job item property. This method has the following prototype:

```
void set (int kwNumber, int kwValue)
```

kwNumber

Specifies the enumerated keyword, for example, `KW_N_RETRYS`.

kwValue

Specifies the value for the job item property.

set

This set method in the `JobItem` class sets the string value for a job item property. This method has the following prototype:

```
void set(int kwNumber, const std::string & kwValue)
```

kwNumber

Specifies the enumerated keyword, for example, `KW_JOB_NAME`.

kwValue

Specifies the value for the job item property, for example, `TestJob`.

set

This set method in the `JobItem` class sets the boolean value for a job property. This method has the following prototype:

```
void set(int kwNumber, bool kwValue)
```

kwNumber

Specifies the enumerated keyword, for example, `KW_ALARM_IF_FAIL`.

kwValue

Specifies the Boolean value for the job property.

[setItem Type](#)

The `setItem Type` method in the `JobItem` class sets the item type for a repeating job item. This method has the following prototype:

```
void setItemType (int item_type)
```

item_type

Specifies the job item type.

[MissingFinalResponse](#)

The `MissingFinalResponse` class is used to indicate that the client attempted to get final results from `ApiResponseSet` before all responses had been retrieved.

ModifyCycleCalendarDays

ModifyCycleCalendarDays modifies a cycle calendar in the CA Workload Automation AE database.

ModifyCycleCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the ModifyCycleCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    ModifyCycleCalendarDaysReq req;
    req.setName("CycleTest");
    req.appendCycle("07/01/2006 12:00:00", "07/15/2006 12:00:00");
    req.appendCycle("08/12/2006 10:00:00", "08/28/2006 10:30:00");
    ModifyCycleCalendarDaysRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

ModifyCycleCalendarDaysReq

extends
 ApiRequest
header
 AsPublicSDK.h

appendCycle

The appendCycle method in the ModifyCycleCalendarDaysReq class appends date ranges to the cycle calendar. This method has the following prototype:

```
void appendCycle (const std::string &strStartDate, const std::string &strEndDate)
```

strStartDate

Specifies the date that the cycle should start, using the standard format:
MM/DD/YYYY HH:MM:SS.

strEndDate

Specifies the date that the cycle should end, using the standard format:
MM/DD/YYYY HH:MM:SS.

execute

The execute method in the ModifyCycleCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, ModifyCycleCalendarDaysRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setName

The setName method in the ModifyCycleCalendarDaysReq class sets the name of the cycle calendar to modify. This method has the following prototype:

```
void setName (const std::string &strCycleCalendarName)
```

strCycleCalendarName

Specifies the name of the cycle calendar to modify.

ModifyExtendedCalendarDays

ModifyExtendedCalendarDays modifies an extended calendar definition.
ModifyExtendedCalendarDays is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the ModifyExtendedCalendarDays API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    ModifyExtendedCalendarDaysReq req;
    req.setCalendarName("MyCalendar");
    req.setWorkdays("XX_X_");
    ModifyExtendedCalendarDaysRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

ModifyExtendedCalendarDaysReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the ModifyExtendedCalendarDaysReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, SimpleRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setAdjustment

The setAdjustment method in the ModifyExtendedCalendarDaysReq class specifies the number of days before or after the condition setting. This method has the following prototype:

```
void setAdjustment (int iAdjustment) const
```

iAdjustment

Specifies a negative or positive number defining the days before or after the condition setting.

setCalendarName

The setCalendarName method in the ModifyExtendedCalendarDaysReq class specifies the name of the calendar to modify. This method has the following prototype:

```
void setCalendarName (const std::string &strCalendarName) const
```

strCalendarName

Specifies the name of the calendar to modify.

setCycleCalName

The `setCycleCalName` method in the `ModifyExtendedCalendarDaysReq` class sets the name of the cycle calendar. This method has the following prototype:

```
void setCycleCalName (const std::string &strCycleCalName) const  
strCycleCalName
```

Specifies the name of the cycle calendar to use.

setDateCon

The `setDateCon` method in the `ModifyExtendedCalendarDaysReq` class sets the condition keyword string. This method has the following prototype:

```
void setDateCon (const std::string &strDateCon) const  
strDateCon
```

Specifies the condition keyword string.

setHolidayAction

The `setHolidayAction` method in the `ModifyExtendedCalendarDaysReq` class specifies the action to take when the work falls on a holiday. This method has the following prototype:

```
void setHolidayAction (eCalendarAction eHolidayAction) const  
eHolidayAction
```

Specifies the holiday action. Valid holiday actions are:

- `CaDefault`
- `CaOnly`
- `CaSchedule`
- `CaNextDay`
- `CaNextWorkDay`
- `CaPreviousWorkDay`

setHolidayCalendar

The `setHolidayCalendar` method in the `ModifyExtendedCalendarDaysReq` class specifies the name of the holiday calendar to use. This method has the following prototype:

```
void setHolidayCalendar (const std::string &strHolidayCalendar) const  
strHolidayCalendar
```

Specifies the name of the holiday calendar

setWeekendAction

The setWeekendAction method in the ModifyExtendedCalendarDaysReq class specifies the action to take when the work falls on a non-workday. This method has the following prototype:

```
void setWeekendAction (eCalendarAction eWeekendAction) const  
eWeekendAction
```

Specifies the action to take when the work falls on a non-workday. Valid weekend actions are:

- CaDefault
- CaOnly
- CaNextDay
- CaNextWorkDay
- CaPreviousWorkDay

setWorkdays

The setWorkdays method in the ModifyExtendedCalendarDaysReq class sets the workday string. This method has the following prototype:

```
void setWorkdays (const std::string &strWorkdays) const  
strWorkdays
```

Specifies a string that represents the selected workdays. For example:

- To select Monday through Friday, pass the string XXXXX__ to setWorkdays.
- To select Monday, Wednesday and Friday, pass the string X_X_X__ to setWorkdays.
- To select Monday and Friday, pass the string X__X__ to setWorkdays.

NoResponseAvailable

The NoResponseAvailable class is used to relay that an attempt was made to retrieve a Response (ApiResponseSet::next()) but one is not available. Typically, ApiResponseSet::next() was called without first calling ApiResponseSet::hasMore().

OverrideJob

OverrideJob is used to override job properties.

OverrideJob is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the OverrideJob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    OverrideJobReq req;
    req.setJobName("MyJob");
    req.setMachine("Mozart");
    OverrideJobRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

OverrideJobReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the OverrideJobReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, OverrideJobRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an OverrideJobRsp object.

setAutoHold

The setAutoHold method in the OverrideJobReq class specifies whether to change the job state to ON_HOLD when the box it is in begins RUNNING. This method has the following prototype:

```
void setAutoHold (bool bAutoHold)
```

bAutoHold

Specifies whether to change the job state. Possible values are:

TRUE

Turns this option on.

FALSE

Turns this option off.

setCommand

The setCommand method in the OverrideJobReq class specifies the job command, for example, cd, echo hello. This method has the following prototype:

```
void setCommand (const std::string &strCommand)
```

strCommand

Specifies the job command.

setCondition

The setCondition method in the OverrideJobReq class specifies the conditions for the job to run, for example, success(DB_BACKUP), exitcode (my_job=4), VALUE(TODAY)=Friday. This method has the following prototype:

```
void setCondition (const std::string &strCondition)
```

strCondition

Specifies the conditions for the job to run.

setDateConditions

The setDateCondition method in the OverrideJobReq class specifies whether there are date or time conditions for starting the job. This method has the following prototype:

```
void setDateConditions (bool bDateConditions)
```

bDateConditions

Specifies whether there are date or time conditions for starting the job. Possible values are:

TRUE

Turns this option on.

FALSE

Turns this option off.

setDaysOfWeek

The setDaysOfWeek method in the OverrideJobReq class specifies the days of the week on which to run the job, for example, mo,tu,we,fr. This method has the following prototype:

```
void setDaysOfWeek (const std::string &strDaysOfWeek)
```

strDaysOfWeek

Specifies the days of the week on which to run the job.

setExcludeCalendar

The setExcludeCalendar method in the OverrideJobReq class specifies the name of the custom calendar to be used for determining the days of the week on which to not run this job. This method has the following prototype:

```
void setExcludeCalendar (const std::string &strExcludeCalendar)
```

strExcludeCalendar

Specifies the name of the custom exclude calendar.

setJobName

The setJobName method in the OverrideJobReq class specifies the CA Workload Automation AE job name. This method has the following prototype:

```
void setJobName (const std::string &strJobName)
```

strJobName

Specifies the job name.

setMachine

The setMachine method in the OverrideJobReq class specifies the machine where the job is performed. This method has the following prototype:

```
void setMachine (const std::string &strMachine)
```

strMachine

Specifies the machine name.

setMaxRunAlarm

The setMaxRunAlarm method in the OverrideJobReq class specifies the maximum run time (in minutes) that a job should require to finish normally. This method has the following prototype:

```
void setMaxRunAlarm (int iMaxRunAlarm)
```

iMaxRunAlarm

Specifies the maximum runtime alarm value.

setMinRunAlarm

The setMinRunAlarm method in the OverrideJobReq class specifies the minimum run time (in minutes) that a job should require to finish normally. This method has the following prototype:

```
void setMinRunAlarm (int iMinRunAlarm)
```

iMinRunAlarm

Specifies the minimum runtime alarm value.

setProfile

The setProfile method in the OverrideJobReq class specifies a job profile that defines environment variables to be set before the specified command is executed. This method has the following prototype:

```
void setProfile (const std::string &strProfile)
```

strProfile

Specifies the job profile.

setRetries

The setRetries method in the OverrideJobReq class specifies how many times, if any, the job restart should be attempted after exiting with a FAILURE status. This method has the following prototype:

```
void setRetries (int iRetries)
```

iRetries

Specifies the number of times a job restart should be attempted after exiting with a FAILURE status.

setRunCalendar

The setRunCalendar method in the OverrideJobReq class specifies the name of the custom calendar to be used when determining the days of the week on which to run a job. This method has the following prototype:

```
void setRunCalendar (const std::string &strRunCalendar)
```

strRunCalendar

Specifies the run calendar.

setRunWindow

The setRunWindow method in the OverrideJobReq class specifies the time span during which the job is allowed to start. If this attribute is specified, it implies the job is allowed to start only during this interval. This method has the following prototype:

```
void setRunWindow (const std::string &strRunWindow)
```

strRunWindow

Specifies the run window.

setStartMins

The setStartMins method in the OverrideJobReq class specifies the number of minutes past the hour, every hour, to start the job on the specified days. This method has the following prototype:

```
void setStartMins (const std::string &strStartMins)
```

strStartMins

Specifies the number of minutes past the hour in which to start the job.

setStartTimes

The setStartTimes method in the OverrideJobReq class specifies the times of day, in 24-hour format, to start the job on the specified days or dates. This method has the following prototype:

```
void setStartTimes (const std::string &strStartTimes)
```

strStartTimes

Specifies the times of day to start the job.

setStderrFile

The setStderrFile method in the OverrideJobReq class specifies the file to which the standard error file output should be redirected. This method has the following prototype:

```
void setStderrFile (const std::string &strStderrFile)
```

strStderrFile

Specifies the standard error output file.

setStdinFile

The setStdinFile method in the OverrideJobReq class specifies the file to which the standard input file for the job should be redirected. This method has the following prototype:

```
void setStdinFile (const std::string &strStdinFile)
```

strStdinFile

Specifies the standard input file.

[setStdoutFile](#)

The `setStdoutFile` method in the `OverrideJobReq` class specifies the file to which the standard output file should be redirected. This method has the following prototype:

```
void setStdoutFile (const std::string &strStdoutFile)
```

strStdoutFile

Specifies the standard output file.

[setSyntaxChecker](#)

The `setSyntaxChecker` method in the `OverrideJobReq` class verifies the overrides. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck)
```

bCheck

Specifies whether to verify the overrides. Possible values are:

TRUE

Turns this option on.

FALSE

Turns this option off.

Note: The job definition is not written to the database.

[setTermRunTime](#)

The `setTermRunTime` method in the `OverrideJobReq` class specifies the maximum run time (in minutes) that a job should require to finish normally. This method has the following prototype:

```
void setTermRunTime (int iTermRunTime)
```

iTermRunTime

Specifies the maximum run time (in minutes) for a job to finish.

[setWatchFile](#)

The `setWatchFile` method in the `OverrideJobReq` class specifies the full path name of the file for which this file watcher job should watch. This method has the following prototype:

```
void setWatchFile (const std::string &strWatchFile)
```

strWatchFile

Specifies the name of the watch file.

setWatchFileMinSize

The setWatchFileMinSize method in the OverrideJobReq class specifies the watch file minimum size (in bytes), which determines when enough data has been written to the file to consider it complete. This method has the following prototype:

```
void setWatchFileMinSize (long long iWatchFileMinSize)
```

iWatchFileMinSize

Specifies the watch file minimum size (in bytes).

setWatchInterval

The setWatchInterval method in the OverrideJobReq class specifies the interval (in seconds) at which the file watcher job checks for the existence and size of the watched-for file. This method has the following prototype:

```
void setWatchInterval (int iWatchInterval)
```

iWatchInterval

Specifies the watch interval (in seconds).

OverrideJobRsp

extends

ApiResponse

header

AsPublicSDK.h

getOverrideNumber

The getOverrideNumber method in the OverrideJobRsp class returns the number of the override just set. This method has the following prototype:

```
int getOverrideNumber (void) const
```

PingApi

PingApi pings the CA Workload Automation AE server. PingApi is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the PingApi API

```
AsApi api;
try {
    api.openRemote(port_, server_);
    PingApiReq req;
    PingApiRsp rsp;
    req.execute(api, rsp);
    std::cout << "\nServerVersion: " << rsp.getServerVersion() << std::endl;
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

PingApiReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the PingApiReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, PingApiRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a PingApiRsp object.

PingApiRsp

extends
 ApiResponse
header
 AsPublicSDK.h

getServerVersion

The getServerVersion method in the PingApiRsp class has the following prototype:

`const int getServerVersion ()`

RegenerateExtendedCalendar

RegenerateExtendedCalendar regenerates an extended calendar in the CA Workload Automation AE database. RegenerateExtendedCalendar is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the RegenerateExtendedCalendar API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    RegenerateExtendedCalendarReq req;
    req.setName("MyExtCal");
    RegenerateExtendedCalendarRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

RegenerateExtendedCalendarReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the RegenerateExtendedCalendarReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, SimpleRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setName

The setName method in the RegenerateExtendedCalendarReq class sets the name of extended calendar to be regenerated. This method has the following prototype:

`void setName (const std::string &name)`

name

Specifies the extended calendar name.

RemoveOverride

RemoveOverride deletes an override that has been set for a job. This API is used by JIL to execute "override_job: delete". This API does not support wildcards.

Given a job name, RemoveOverride verifies that the caller has write access to the job. It then removes the override and sends or deletes future events if the date or time fields are affected.

RemoveOverride is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the RemoveOverride API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    RemoveOverrideReq req;
    req.setJobName("MyJob");
    RemoveOverrideRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

RemoveOverrideReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the RemoveOverrideReq class performs the API call. This method has the following prototype:

```
void execute (AsApi &api, RemoveOverrideRsp &rsp)
```

api

Specifies an AsApi object.

rsp

Specifies a RemoveOverrideRsp object.

setJobName

The setJobName method in the RemoveOverrideReq class specifies the CA Workload Automation AE job name for which to remove an override. This method has the following prototype:

```
void setJobName (const std::string &strJobName)
```

strJobName

Specifies the name of the job for which to remove the override.

setSyntaxChecker

The setSyntaxChecker method in the RemoveOverrideReq class verifies the remove override. The remove override is not performed. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck)
```

bCheck

Specifies whether to verify the remove override. Possible values are:

TRUE

Turns this option on.

FALSE

Turns this option off.

ResponseDequeueFailure

ResponseDequeueFailure is used to indicate that an internal failure occurred when trying to retrieve a response from the API.

SapArcParmItem

SapArcParmItem is the container class for specifying SAP archive parameters in job definitions. This JobItem subclass directly maps to the arc_parms jil keyword.

Example: Sample Code for Executing the SapArcParmItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_SAPDA);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_ARC_OBJ_VARIANT, (std::string) "myVariant");
    req.set(KW_ARC_OBJ_NAME, (std::string) "myArcObjName");
    req.set(KW_SAP_CLIENT, (std::string) "23");
    req.set(KW_SAP_LANG, (std::string) "EN");
    req.set(KW_SAP_RFC_DEST, (std::string) "myRFCDest");
    req.set(KW_SAP_TARGET_SYS, (std::string) "MyTargSys");
    // Set up the arc_parms part of the job definition and
    // attach it to the base job definition.
    SapArcParmItem sapi;
    sapi.set(KW_ARC_FORMAT, std::string("myFormat"));
    sapi.set(KW_ARC_PRINTER, std::string("PRTR"));
    sapi.set(KW_ARC_CLIENT, 235);
    sapi.set(KW_ARC_CONNECT, std::string("myConnect"));
    sapi.set(KW_ARC_DOC_CLASS, std::string("myDocClass"));
    sapi.set(KW_ARC_DOC_TYPE, std::string("myDocType"));
    sapi.set(KW_ARC_HOST_LINK, std::string("myHostLink"));
    sapi.set(KW_ARC_INFO, std::string("INF"));
    sapi.set(KW_ARC_OBJ_TYPE, std::string("myObjType"));
    sapi.set(KW_ARC_PATH, std::string("myPath"));
    sapi.set(KW_ARC_PROTOCOL, std::string("myProto"));
    sapi.set(KW_ARC_REPORT, std::string("myReport"));
    sapi.set(KW_ARC_SERVICE, std::string("myService"));
    sapi.set(KW_ARC_STORAGE, std::string("my"));
    sapi.set(KW_ARC_TEXT, std::string("myText"));
    sapi.set(KW_ARC_VERSION, std::string("Vers"));
    sapi.set(KW_ARC_USERID, std::string("myUserId"));
    sapi.set(KW_ARC_DATE, std::string("20101201"));
    req.add(&sapi);
}
```

```

SapPrintParmItem spp;
    spp.set(KW_PRT_ARC_MODE, EN_SAPARCMODE_BOTH);
    spp.set(KW_SAP_BANNER,EN_YND_N);
    spp.set(KW_BANNER_PAGE, "n");
    spp.set(KW_EXPIRATION, 9);
    spp.set(KW_NUM_COLUMNS,3);
    spp.set(KW_COPIES, 2);
    spp.set(KW_DEPT, std::string("myDept"));
    spp.set(KW_DEST, std::string("myDest"));
    spp.set(KW_DATASET, std::string("myDataset"));
    req.add(&spp);
    req.execute(api, rsp);
}

catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_ARC_OBJ_VARIANT, KW_ARC_PARMS};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
        rset.next(rsp);
        for (int i=0; i < rsp.getAttributes(); i++) {

            // Display keywords that are of type string.
            if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                    rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }
        }
    }
}

```

```
// Display job items.  
else if( rsp.getAttributeType(i) == AtJobItem ) {  
    autosys::JobItem myJobitem = rsp.getJobItem(i);  
    std::vector<int> keywords =  
        myJobitem.getVectorOfKeywordConstants();  
    for( int j=0; j < myJobitem.getAttributes(); j++ ) {  
        // Only displaying jobitem keywords that are of type string.  
        std::cout << "Jobitem string attribute [" <<  
            rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<  
            myJobitem.getString( (int)keywords[j] ) << "]" <<  
            std::endl;  
    }  
}  
}  
}  
}  
}  
}  
}  
catch(ApiExceptionWithInfo& e) {  
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<  
    std::endl;  
}
```

SapExtTableItem

The SapExtTableItem class is the container class for specifying SAP ext_table parameters in job definitions.

Example: Sample Code for Executing the SapExtTableItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_SAPBWIP);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_JOB_CLASS, std::string("myClass"));
    req.set(KW_SAP_CLIENT, (std::string) "235");
    req.set(KW_SAP_INFO_PACK, (std::string) "mySapInfoPak");
    req.set(KW_ARC_OBJ_NAME, (std::string) "myArcObjName");

    // Set up the sap_ext_table part of the job definition and
    // attach it to the base job definition.
    SapExtTableItem seti;
    seti.set(KW_FIELD_NAME, (std::string) "myField");
    seti.set(KW_HIGH, (std::string) "high");
    seti.set(KW_LOW, (std::string) "low");
    seti.set(KW_OBJECT_NAME, (std::string) "myObjectName");
    seti.set(KW_SIGN, EN_SAPDETAILSIGNTYPE_I);
    seti.set(KW_OPERATION, EN_SAPDETAILOPTYPE_EQ);
    req.add(&seti);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_JOB_CLASS, KW_SAP_CLIENT,
KW_SAP_EXT_TABLE};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);
```

```
GetJobsWithFilterRspSet rset;
req.execute(api, rset);
while( rset.hasMore() ) {
    GetJobsWithFilterRsp rsp;
    rset.next(rsp);
    for (int i=0; i < rsp.getAttributes(); i++) {

        // Display keywords that are of type string.
        if( rsp.getAttributeType(i) == AtString ) {
            std::cout << "String attribute [" <<
            rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
            rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
        }

        // Display job items.
        else if( rsp.getAttributeType(i) == AtJobItem ) {
            autosys::JobItem myJobitem = rsp.getJobItem(i);
            std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
            for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                // Only displaying jobitem keywords that are of type string.
                std::cout << "Jobitem string attribute [" <<
                rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                myJobitem.getString( (int)keywords[j] ) << "]" <<
            std::endl;
            }
        }
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
std::endl;
}
```

SapPrintParmItem

The SapPrintParmItem class is the container class for specifying SAP print parameters in job definitions.

Example: Sample Code for Executing the SapPrintParmItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_SAPDA);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_JOB_CLASS, std::string("myClass"));
    req.set(KW_SAP_CLIENT, (std::string) "235");
    req.set(KW_ARC_OBJ_VARIANT, (std::string) "myVariant");

    // Set up the sap_print_parms part of the job definition and
    // attach it to the base job definition.
    SapPrintParmItem spp;
    spp.set(KW_BANNER_PAGE, "y");
    spp.set(KW_COPIES, 2);
    spp.set(KW_DATASET, std::string("myData"));
    spp.set(KW_DEPT, std::string("myDepartment"));
    spp.set(KW_DEST, std::string("LP01"));
    spp.set(KW_EXPIRATION, 9);
    spp.set(KW_FOOTER, "n");
    spp.set(KW_FORMAT, std::string("X_65_132"));
    spp.set(KW_HOST_PAGE, EN_YND_N);
    spp.set(KW_NEW_SPOOL, "y");
    spp.set(KW_NUM_COLUMNS, 132);
    spp.set(KW_NUM_LINES, 65);
    spp.set(KW_PRT_ARC_MODE, EN_SAPARCMODE_PRINT);
    spp.set(KW_PRINT_IMM, "y");
    spp.set(KW_PRINT_PRIORITY, 1);
    spp.set(KW_RECIPIENT_NAME, std::string("myRecipient"));
    spp.set(KW_RELEASE, "y");
    spp.set(KW_REQ_TYPE, std::string("myRequestType"));
    spp.set(KW_SAP_BANNER, EN_YND_N);
    spp.set(KW_SPOOL_NAME, std::string("mySpoolName"));
    spp.set(KW_TITLE, std::string("myTitle"));
    req.add(&spp);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.
```

```
// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_JOB_CLASS, KW_SAP_CLIENT,
KW_ARC_OBJ_VARIANT, KW_SAP_PRINT_PARMS};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) / 
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
        rset.next(rsp);
        for (int i=0; i < rsp.getAttributes(); i++) {

            // Display keywords that are of type string.
            if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                    rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }

            // Display job items.
            else if( rsp.getAttributeType(i) == AtJobItem ) {
                autosys::JobItem myJobitem = rsp.getJobItem(i);
                std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
                for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                    // Only displaying jobitem keywords that are of type string.
                    std::cout << "Jobitem string attribute [" <<
rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                        myJobitem.getString( (int)keywords[j] ) << "]" <<
                std::endl;
                }
            }
        }
    }
    catch(ApiExceptionWithInfo& e) {
        std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```

SapRecipientItem

The SapRecipientItem class is the container class for specifying SAP recipient parameters in job definitions.

Example: Sample Code for Executing the SapRecipientItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_SAP);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_JOB_CLASS, std::string("myClass"));
    req.set(KW_SAP_CLIENT, (std::string) "235");
    req.set(KW_SAP_EVENT_ID, (std::string) "myEventID");
    req.set(KW_SAP_JOB_NAME, (std::string) "mySapJob");

    // Set up the sap_recipient part of the job definition and
    // attach it to the base job definition.
    SapRecipientItem sr;
    sr.set(KW_RCP_TYPE, EN_SAPRECIPTYPE_INT);
    sr.set(KW_RECIPIENT, std::string("myAddress@ca.com"));
    req.add(&sr);

    // Set up the sap_step_parms of the job definition and
    // attach it to the base job definition.
    SapStepParmItem ssp;
    ssp.set(KW_ABAP_LANG, (std::string) "EN");
    ssp.set(KW_ABAP_NAME, (std::string) "RSLG0000");
    req.add(&ssp);

    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_JOB_CLASS, KW_SAP_CLIENT,
KW_SAP_EVENT_ID, KW_SAP_RECIPIENTS};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");
```

```
try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
        rset.next(rsp);
        for (int i=0; i < rsp.getAttributes(); i++) {

            // Display keywords that are of type string.
            if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }

            // Display job items.
            else if( rsp.getAttributeType(i) == AtJobItem ) {
                autosys::JobItem myJobitem = rsp.getJobItem(i);
                std::vector<int> keywords =
                myJobitem.getVectorOfKeywordConstants();
                for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                    // Only displaying jobitem keywords that are of type string.
                    std::cout << "Jobitem string attribute [" <<
                    rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                    myJobitem.getString( (int)keywords[j] ) << "]" <<
                    std::endl;
                }
            }
        }
    }
    catch(ApiExceptionWithInfo& e) {
        std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
        std::endl;
    }
}
```

SapStepParmItem

The SapStepParmItem class is the container class for specifying SAP step parameters in job definitions.

Example: Sample Code for Executing the SapStepParmItem API

```

InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_SAPBWIP);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_JOB_CLASS, std::string("myClass"));
    req.set(KW_SAP_CLIENT, (std::string) "235");
    req.set(KW_SAP_EVENT_ID, (std::string) "myEventID");
    req.set(KW_SAP_JOB_NAME, (std::string) "mySapJob");

    SapStepParmItem sspi;
    sspi.set(KW_ABAP_LANG, (std::string) "EN");
    sspi.set(KW_ABAP_NAME, (std::string) "RSLG0000");

    req.add(sspi);
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_JOB_CLASS, KW_SAP_CLIENT,
KW_SAP_EVENT_ID, KW_SAP_STEP_PARM};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) / 
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;

```

```
rset.next(rsp);
for (int i=0; i < rsp.getAttributes(); i++) {

    // Display keywords that are of type string.
    if( rsp.getAttributeType(i) == AtString ) {
        std::cout << "String attribute [" <<
        rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
            rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
    }

    // Display job items.
    else if( rsp.getAttributeType(i) == AtJobItem ) {
        autosys::JobItem myJobitem = rsp.getJobItem(i);
        std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
        for( int j=0; j < myJobitem.getAttributes(); j++ ) {
            // Only displaying jobitem keywords that are of type string.
            std::cout << "Jobitem string attribute [" <<
            rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                myJobitem.getString( (int)keywords[j] ) << "]" <<
            std::endl;
        }
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```

SendEvt

SendEvt sends an event for processing by the scheduler.

SendEvt checks the validity of the requested event, and checks if the user has the appropriate permissions for the event. Permissions are required as follows:

- Create access is required to add a new global variable.
- Write access is required to change the value of an existing global variable or to send the CHANGE_PRIORITY event.
- Execute access is required to send the following events: CHANGE_STATUS, COMMENT, FORCE_STARTJOB, JOB_OFF_HOLD, JOB_OFF_ICE, JOB_ON_HOLD, JOB_ON_ICE, KILLJOB, SEND_SIGNAL, SET_GLOBAL, and STARTJOB.
- Machine write access is required to send a MACH_OFFLINE or MACH_ONLINE event. In the default product security mode, the user must be an EXEC superuser.
- When DELETEJOB is the requested event, the user must have delete access to the specified job. For CHANGE_PRIORITY, the user must have write access to the job. For all other job-related events, the user must have execute access to the job.
- When STOP_DEMON is the requested event, the user must have permission to execute a STOP_DEMON.
- For SET_GLOBAL, the user must have execute permissions for the specified global variable. If the variable is new, the user must also have create access.
- When the value is set to DELETE, the user must also have delete permissions.

When SendEvt has validated the requested event and the user's access to the specified job, the event is committed to the database and audit information is updated.

The code interface enforces which attributes must be set for which events. Setting up an event to send consists of calling a specific event request such as setKillJobRequest and calling the generic setCommonValues method. setCommonValues sets attributes that apply to all event types.

SendEvt is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException

- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the SendEvt API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    SendEvtReq req;
    SendEvtRspSet rset;

    req.setStartJob("TestJob");
    req.execute(api, rset);
    while (rset.hasMore()) {
        SendEvtRsp rsp;
        rset.next(rsp);
        if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
        std::cout << "ObjectName: " << rsp.getObjectName() << std::endl;
    }

    catch(ApiTimeout&) {
        std::cerr << "ApiTimeout thrown." << std::endl;
    }
    catch(AsExternalValidationException&) {
        std::cerr << "AsExternalValidationException thrown." << std::endl;
    }
    catch(AsFieldValidationException&) {
        std::cerr << "AsFieldValidationException thrown." << std::endl;
    }
    catch(AsGeneralErrorException&) {
        std::cerr << "AsGeneralErrorException thrown." << std::endl;
    }
    catch(AsObjectDoesNotExistException&) {
        std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
    }
    catch(AsSecurityException&) {
        std::cerr << "AsSecurityException thrown." << std::endl;
    }
    catch(AsUnknownErrorException&) {
```

```
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(Cat2StartFailure&) {
        std::cerr << "AsUnknownErrorException thrown." << std::endl;
    }
    catch(CredsNotAuthorized&) {
        std::cerr << "CredsNotAuthorized thrown." << std::endl;
    }
    catch(FailedToSetCreds&) {
        std::cerr << "FailedToSetCreds thrown." << std::endl;
    }
    catch(InternalResponseError&) {
        std::cerr << "InternalResponseError thrown." << std::endl;
    }
    catch(InternalUnexpected&) {
        std::cerr << "InternalUnexpected thrown." << std::endl;
    }
    catch(MissingFinalResponse&) {
        std::cerr << "MissingFinalResponse thrown." << std::endl;
    }
    catch(NoResponseAvailable&) {
        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure&) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}
```

SendEvtReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the SendEvtReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, SendEventRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a SendEventRsp object.

setAlarm

The setAlarm method in the SendEvtReq class configures an ALARM event. This method has the following prototype:

`void setAlarm (int iAlarmID)`

iAlarmID

Specifies the alarm ID.

setChangePriority

The setChangePriority method in the SendEvtReq class configures a CHANGE_PRIORITY event. This method has the following prototype:

`void setChangePriority (const std::string &strJob)`

strJob

Specifies the job for which to change the priority.

setChangeStatus

The setChangeStatus method in the SendEvtReq class configures a CHANGE_STATUS event. This method has the following prototype:

```
void setChangeStatus (const std::string &strJob, int iStatus)
```

strJob

Specifies the job for which to change the status.

iStatus

Specifies the new status value.

setComment

The setComment method in the SendEvtReq class configures a COMMENT event. The setCommonValues method sets the comment. This method has the following prototype:

```
void setComment ()
```

setCommonValues

The setCommonValues method in the SendEvtReq class specifies common values for a send event request. This method has the following prototype:

```
void setCommonValues (const std::string &strComment, int iPriority, int iMaxSendTrys, int iJobQueuePriority, time_t sTime)
```

strComment

Specifies a free-form text comment to place in the CA Workload Automation AE log.
Pass NULL if no comment.

iPriority

Specifies the priority of the event.

Limits: 1-10. A priority 1 event executes prior to a priority 10 event.

iMaxSendTrys

Specifies the maximum number of times to retry sending the event.

iJobQueuePriority

Specifies the job queue priority. This value is used with load balancing.

sTime

Specifies the time at which to execute the event. Pass a -1 value to any of the setCommonValues int parameters to take default value.

setDeleteJob

The setDeleteJob method in the SendEvtReq class configures a DELETEJOB event. This method has the following prototype:

```
void setDeleteJob (const std::string &strJob)
```

strJob

Specifies the job to be deleted.

setForceStartJob

The setForceStartJob method in the SendEvtReq class configures a FORCE_STARTJOB event. This method has the following prototype:

```
void setForceStartJob (const std::string &strJob)
```

strJob

Specifies the name of the job to force start.

setJobOffIce

The setJobOffIce method in the SendEvtReq class configures a JOB_OFF_ICE event. This method has the following prototype:

```
void setJobOffIce (const std::string &strJob)
```

strJob

Specifies the job to be placed off ice.

setJobOnHold

The setJobOnHold method in the SendEvtReq class configures a JOB_ON_HOLD event. This method has the following prototype:

```
void setJobOnHold (const std::string &strJob)
```

strJob

Specifies the job to be placed on hold.

setJobOnIce

The setJobOnIce method in the SendEvtReq class configures a JOB_ON_ICE event. This method has the following prototype:

```
void setJobOnIce (const std::string &strJob)
```

strJob

Specifies the job to be placed on ice.

setJobOffHold

The setJobOffHold method in the SendEvtReq class configures a JOB_OFF_HOLD event. This method has the following prototype:

```
void setJobOffHold (const std::string &strJob)
```

strJob

Specifies the job to be placed off hold.

setKillJob

The setKillJob method in the SendEvtReq class configures a KILLJOB event. This method has the following prototype:

```
void setKillJob (const std::string &strJob, const std::vector< int > signals, int num_signals)
```

strJob

Specifies the name of the job to kill.

signals

Specifies an array of signal values which can be used to kill jobs on UNIX Remote Agents.

num_signals

Specifies the number of signals.

Default: 0

setMachOffline

The setMachOffline method in the SendEvtReq class takes the specified machine offline. This method has the following prototype:

```
void setMachOffline (const std::string &strMachine)
```

strMachine

Specifies the machine to be taken offline.

setMachOnline

The setMachOnline method in the SendEvtReq class takes the specified machine online. This method has the following prototype:

```
void setMachOnline (const std::string &strMachine)
```

strMachine

Specifies the machine to be taken online.

setReleaseResource

This setReleaseResource method in the SendEvtReq class configures a RELEASE_RESOURCE event. This method has the following prototype:

```
setReleaseResource (const std::string & strJob)
```

strJob

Specifies the job whose resources are to be released.

setReplyResponse

This setReplyResponse method in the SendEvtReq class configures a REPLY_RESPONSE event. This method has the following prototype:

```
setReplyResponse (const std::string & strJob, const std::string & strReply)
```

strJob

Specifies the job associated with the response.

strReply

Specifies the reply data for the job.

setSendSignal

The setSendSignal method in the SendEvtReq class configures a SEND_SIGNAL event. This method has the following prototype:

```
void setSendSignal (const std::string &strJob, const std::vector< int > signals, int num_signals)
```

strJob

Specifies the job to receive the signals.

signals

Specifies an array of signal values to send.

num_signals

Specifies the number of signals to send.

setSetGlobal

The setSetGlobal method in the SendEvtReq class configures a SET_GLOBAL event. This method has the following prototype:

```
void setSetGlobal (const std::string &strGlobalName, const std::string &strGlobalValue)
```

strGlobalName

Specifies the name of the global variable.

strGlobalValue

Specifies the global variables value. A value of DELETE deletes the existing global variable.

setStartJob

The setStartJob method in the SendEvtReq class configures a STARTJOB event. This method has the following prototype:

```
void setStartJob (const std::string &strJob)
```

strJob

Specifies the job to start.

setStopDemon

The setStopDemon method in the SendEvtReq class configures a STOP_DEMON event. This method has the following prototype:

```
void setStopDemon ()
```

SendEvtRsp

extends

ApiResponse

header

AsPublicSDK.h

getObjectName

The getObjectName method in the SendEvtRsp class returns the requested object name. This method has the following prototype:

```
std::string getObjectName () const
```

StoredProcedureArgumentItem

The StoredProcedureArgumentItem class is the container class for specifying stored procedure arguments in job definitions. This JobItem subclass directly maps to the sp_arg jil keyword.

Example: Sample Code for Executing the StoredProcedureArgumentItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_DBPROC);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_JOB_CLASS, (std::string) "myClass");
    req.set(KW_SP_NAME, (std::string) "myProcName");

    // Set up the sp_arg of the job definition and
    // attach it to the base job definition.
    StoredProcedureArgumentItem spa;
    spa.set(KW_ARGLTYPE, EN_DBARGTYPE_INOUT);
    spa.set(KW_DATATYPE, EN_DBDATATYPE_VARCHAR);
    spa.set(KW_IGNORE, true);
    spa.set(KW_NAME, (std::string) "myName");
    spa.set(KW_VALUE, (std::string) "myValue");
    req.add(&spa);

    // Set up a second sp_arg for the the job definition and
    // attach it to the base job definition.
    StoredProcedureArgumentItem spa2;
    spa2.set(KW_ARGLTYPE, EN_DBARGTYPE_IN);
    spa2.set(KW_DATATYPE, EN_DBDATATYPE_SMALLINT);
    spa2.set(KW_IGNORE, true);
    spa2.set(KW_NAME, (std::string) "myName2");
    spa2.set(KW_VALUE, (std::string) "myValue2");
    req.add(&spa2);

    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_NAME, KW_VALUE, KW_SP_ARG};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );
```

```
// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
        GetJobsWithFilterRsp rsp;
        rset.next(rsp);
        for (int i=0; i < rsp.getAttributes(); i++) {

            // Display keywords that are of type string.
            if( rsp.getAttributeType(i) == AtString ) {
                std::cout << "String attribute [" <<
                rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
            }

            // Display job items.
            else if( rsp.getAttributeType(i) == AtJobItem ) {
                autosys::JobItem myJobitem = rsp.getJobItem(i);
                std::vector<int> keywords =
                    myJobitem.getVectorOfKeywordConstants();
                for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                    // Only displaying jobitem keywords that are of type string.
                    std::cout << "Jobitem string attribute [" <<
                    rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
                    myJobitem.getString( (int)keywords[j] ) << "]" <<
                    std::endl;
                }
            }
        }
    }
}

catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```

UpdateJob

UpdateJob updates a job in the CA Workload Automation AE database.

The UpdateJob performs external job validation (if in place), then verifies security permissions. The API verifies that the caller has write permissions for the job.

UpdateJob follows security checks with inter-field verification (for example, if the job is a command job, then you need to have a command field). It then updates the job and issues future STARTJOB events if required. Finally, the API updates audit information.

UpdateJob is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the UpdateJob API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    UpdateJobReq req;
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_MACHINE, (std::string) "mach01");
    UpdateJobRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
```

```

}
api.close();

```

UpdateJobReq

extends
 ApiRequest
header
 AsPublicSDK.h

add

The add method in the UpdateJobReq class adds a [JobItem](#) (see page 787). This method has the following prototype:

```
void add (JobItem * job_item)  

job_item
```

Specifies a job item object such as sap_recipients or sap_step_parms.

execute

The execute method in the UpdateJobReq class performs the API call. This method has the following prototype:

```
execute (AsApi & api, UpdateJobRsp & rsp)  

api
```

Specifies an AsApi object.

```
rsp
```

Specifies an UpdateJobRsp object.

set

This set method in the UpdateJobReq class sets the Boolean value for the specified keyword. This method has the following prototype:

```
void set (eKeywords keyword, bool value)  

keyword
```

Specifies a keyword that represents a Boolean job property.

```
value
```

Specifies the Boolean value for this keyword.

set

This set method in the `UpdateJobReq` class sets the double value for the specified keyword. This method has the following prototype:

```
void set (eKeywords keyword, double value)
```

keyword

Specifies a keyword that represents a job property of type double.

value

Specifies the value for this keyword.

set

This set method in the `UpdateJobReq` class sets the value for a keyword that represents a BLOB type. This method has the following prototype:

```
void set (eKeywords keyword, void * value, int length)
```

keyword

Specifies a keyword that represents a valid BLOB job property.

value

Specifies the value for this keyword.

length

Specifies the length of the BLOB.

set

This set method in the `UpdateJobReq` class sets the integer value for the specified keyword. This method has the following prototype:

```
void set (eKeywords keyword, int value)
```

keyword

Specifies a keyword that represents an integer job property.

value

Specifies the integer value for this keyword.

set

The `setNull` method in the `UpdateJobReq` class nulls out a keyword's value. This method has the following prototype:

`setNull (eKeywords keyword)`

keyword

Specifies the keyword whose value is to be nullified.

setNull

This `set` method in the `UpdateJobReq` class sets the string value for the specified keyword. This method has the following prototype:

`set (eKeywords keyword, const std::string & value)`

keyword

Specifies a keyword that represents a string job property.

value

Specifies the string value for this keyword.

setSyntaxChecker

The `setSyntaxChecker` method in the `UpdateJobReq` class checks the syntax of the job definition updates. This method has the following prototype:

`void setSyntaxChecker (bool verify)`

verify

Specifies whether to check the syntax of the job definition updates. Possible values are:

TRUE

Checks the syntax of the job definition updates.

Note: The job definition updates are **not** written to the database.

FALSE

Performs the job update.

Note: The job definition updates are written to the database.

setVerifyConditions

The setVerifyConditions method in the UpdateJobReq class checks for missing conditions. This method has the following prototype:

```
void setVerifyConditions (bool verify)
```

verify

Specifies whether to check for missing conditions. Possible values are:

TRUE

Checks for missing conditions.

FALSE

Does not check for missing conditions.

UpdateJobRsp

extends

ApiResponse

header

AsPublicSDK.h

getUndefinedConditionJob

The getUndefinedConditionJob method in the UpdateJobRsp class returns an undefined condition string job. This method has the following prototype:

```
std::string getUndefinedConditionJob(int index) const
```

index

Specifies the index of the undefined job list.

getUndefinedConditionJobsCount

The getUndefinedConditionJobsCount method in the UpdateJobRsp class returns the number of jobs listed in the conditions that are not yet defined. This method has the following prototype:

```
int getUndefinedConditionJobsCount(void) const
```

UnsendEvt

UnsendEvt cancels an event if the event has not yet been processed.

If you do not define a date with a call to the setEventTime(Date timeOfEvent) method, the product cancels all events that satisfy the specified parameters.

When multiple future events exist, use setEventTime(Date timeOfEvent) method to cancel a specific event.

The product does not delete canceled events from the database, it changes their status. This is equivalent to executing "sendevent -U".

UnsendEvt is a CAT2 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- AsExternalValidationException
- AsFieldValidationException
- AsGeneralErrorException
- AsObjectDoesNotExistException
- AsSecurityException
- AsUnknownErrorException
- Cat2StartFailure
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError
- InternalUnexpected
- MissingFinalResponse
- NoResponseAvailable
- ResponseDequeueFailure

Example: Sample Code for Executing the UnsendEvt API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    UnsendEvtReq req;
    req.setSetGlobal("TODAY", "Monday");
    UnsendEvtRspSet rset;
```

```
req.execute(api, rset);
while (rset.hasMore()) {
    UnsendEvtRsp rsp;
    rset.next(rsp);
    if (rsp.hasInfo()) std::cout << rsp.info() << std::endl;
    std::cout << "EventID : " << rsp.getEventID() << std::endl;
    std::cout << "EventTime: " << rsp.getEventTime() << std::endl;
}
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(AsExternalValidationException&) {
    std::cerr << "AsExternalValidationException thrown." << std::endl;
}
catch(AsFieldValidationException&) {
    std::cerr << "AsFieldValidationException thrown." << std::endl;
}
catch(AsGeneralErrorException&) {
    std::cerr << "AsGeneralErrorException thrown." << std::endl;
}
catch(AsObjectDoesNotExistException&) {
    std::cerr << "AsObjectDoesNotExistException thrown." << std::endl;
}
catch(AsSecurityException&) {
    std::cerr << "AsSecurityException thrown." << std::endl;
}
catch(AsUnknownErrorException&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(Cat2StartFailure&) {
    std::cerr << "AsUnknownErrorException thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
catch(InternalUnexpected&) {
    std::cerr << "InternalUnexpected thrown." << std::endl;
}
catch(MissingFinalResponse&) {
    std::cerr << "MissingFinalResponse thrown." << std::endl;
}
catch(NoResponseAvailable&) {
```

```

        std::cerr << "NoResponseAvailable thrown." << std::endl;
    }
    catch(ResponseDequeueFailure& e) {
        std::cerr << "ResponseDequeueFailure thrown." << std::endl;
    }
    catch(ApiExceptionWithInfo& e) {
        std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
        std::cerr << e.what() << std::endl;
    }
    api.close();
}

```

UnsendEvtReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the UnsendEvtReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, SimpleRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setAlarm

The setAlarm method in the UnsendEvtReq class configures an ALARM event. This method has the following prototype:

`void setAlarm (int iAlarmID)`

iAlarmID

Specifies the alarm ID.

[setChangePriority](#)

The `setChangePriority` method in the `UnsendEvtReq` class configures a `CHANGE_PRIORITY` event. This method has the following prototype:

```
void setChangePriority (const std::string &strJob)  
strJob
```

Specifies the name of the job for which to change the priority.

[setChangeStatus](#)

The `setChangeStatus` method in the `UnsendEvtReq` class configures a `CHANGE_STATUS` event. This method has the following prototype:

```
void setChangeStatus (const std::string &strJob, int iStatus)  
strJob
```

Specifies the name of the job for which to change the status.

iStatus

Specifies the new status value.

[setComment](#)

The `setComment` method in the `UnsendEvtReq` class configures a `COMMENT` event. Comment is set using `setCommonValues()`. This method has the following prototype:

```
void setComment ()
```

[setDeleteJob](#)

The `setDeleteJob` method in the `UnsendEvtReq` class configures a `DELETEJOB` event. This method has the following prototype:

```
void setDeleteJob (const std::string &strJob)  
strJob
```

Specifies the job to be deleted.

[setEventTime](#)

The `setEventTime` method in the `UnsendEvtReq` class sets the time for the specified event. This method has the following prototype:

```
void setEventTime (const time_t timeOfEvent)  
timeOfEvent
```

Specifies the time of the specified event.

setForceStartJob

The setForceStartJob method in the UnsendEvtReq class configures a FORCE_STARTJOB event. This method has the following prototype:

```
void setForceStartJob (const std::string &strJob)  
strJob
```

Specifies the name of the job to force start.

setJobOffHold

The setJobOffHold method in the UnsendEvtReq class configures a JOB_OFF_HOLD event. This method has the following prototype:

```
void setJobOffHold (const std::string &strJob)  
strJob
```

Specifies the name of the job to place off hold.

setJobOffIce

The setJobOffIce method in the UnsendEvtReq class configures a JOB_OFF_ICE event. This method has the following prototype:

```
void setJobOffIce (const std::string &strJob)  
strJob
```

Specifies the name of the job to place off ice.

setJobOnHold

The setJobOnHold method in the UnsendEvtReq class configures a JOB_ON_HOLD event. This method has the following prototype:

```
void setJobOnHold (const std::string &strJob)  
strJob
```

Specifies the name of the job to place on hold.

setJobOnIce

The setJobOnIce method in the UnsendEvtReq class configures a JOB_ON_ICE event. This method has the following prototype:

```
void setJobOnIce (const std::string &strJob)  
strJob
```

Specifies the name of the job to place on ice.

[setKillJob](#)

The `setKillJob` method in the `UnsendEvtReq` class configures a KILLJOB event. This method has the following prototype:

```
void setKillJob (const std::string &strJob, const std::vector< int > signals, int  
num_signals)
```

strJob

Specifies the name of the job to kill.

signals

Specifies an array of signal values which can be used to kill jobs on UNIX Remote Agents.

num_signals

Specifies the number of signals.

Default: 0

[setMachOffline](#)

The `setMachOffline` method in the `UnsendEvtReq` class takes the specified machine offline. This method has the following prototype:

```
void setMachOffline (const std::string &strMachine)
```

strMachine

Specifies the machine to take offline.

[setMachOnline](#)

The `setMachOnline` method in the `UnsendEvtReq` class takes the specified machine online. This method has the following prototype:

```
void setMachOnline (const std::string &strMachine)
```

strMachine

Specifies the machine to take online.

setSendSignal

The setSendSignal method in the UnsendEvtReq class configures a SEND_SIGNAL event. This method has the following prototype:

```
void setSendSignal (const std::string &strJob, const std::vector< int > signals, int num_signals)
```

strJob

Specifies the job to receive the signals.

signals

Specifies an array of signal values to send.

num_signals

Specifies the number of signals to send.

setSetGlobal

The setSetGlobal method in the UnsendEvtReq class configures a SET_GLOBAL event. This method has the following prototype:

```
void setSetGlobal (const std::string &strGlobalName, const std::string &strGlobalValue)
```

strGlobalName

Specifies the name of the global variable.

strGlobalValue

Specifies the global variables value. A value of DELETE deletes the existing global variable.

setStartJob

The setStartJob method in the UnsendEvtReq class configures a STARTJOB event. This method has the following prototype:

```
void setStartJob (const std::string &strJob)
```

strJob

Specifies the job to start.

setStopDemon

The setStopDemon method in the UnsendEvtReq class configures a STOP_DEMON event. This method has the following prototype:

```
void setStopDemon ()
```

UnsendEvtRsp

extends
ApiResponse
header
AsPublicSDK.h

getEventID

The getEventID method in the UnsendEvtRsp class returns the requested event ID. This method has the following prototype:

```
std::string getEventID () const
```

getEventTime

The getEventTime method in the UnsendEvtRsp class returns the requested event time. This method has the following prototype:

```
int getEventTime () const
```

UpdateRealMachine

UpdateRealMachine creates/updates a real machine. This API is used by JIL to create real machines.

The UpdateRealMachine API verifies that the caller has create access to the named machine. It then updates the machine and updates audit information.

UpdateRealMachine is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the UpdateRealMachine API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    UpdateRealMachineReq req;
    req.setName("MyMachine");
    req.setMaxLoad(25);
    req.setFactor(1.0);
    UpdateRealMachineRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

UpdateRealMachineReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the UpdateRealMachineReq class performs the API call. This method has the following prototype:

`execute (AsApi &api, UpdateRealMachineRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies an UpdateRealMachineRsp object.

setSyntaxChecker

The setSyntaxChecker method in the UpdateRealMachineReq class verifies the machine definition. This method has the following prototype:

`void setSyntaxChecker (bool bCheck)`

bCheck

Indicates syntax checking is on when set to TRUE and off when FALSE.

setName

The setName method in the UpdateRealMachineReq class specifies the machine name. This method has the following prototype:

`void setName (const std::string & strMachineName)`

strMachineName

Specifies the machine name.

setMaxLoad

The setMaxLoad method in the UpdateRealMachineReq class specifies the maximum load of the machine. This method has the following prototype:

`void setMaxLoad (int iMaxLoad)`

iMaxLoad

Specifies the maximum load (in load units) that a machine can reasonably handle. Load units are arbitrary values, the range of which is user-defined. The parameter applies only to CA Workload Automation AE agents.

setFactor

The setFactor method in the UpdateRealMachineReq class specifies the machine factor. This method has the following prototype:

```
void setFactor (double dFactor)
```

dFactor

Defines a factor to multiply by a real machine's available CPU cycles to determine the "relative available CPU cycles." When more than one real machine or a virtual machine is specified in the job's machine attribute, the real_number value determines on which machine a job should run. The factor value indicates a machine's relative processing power. This factor applies only to CA Workload Automation AE agents.

setDescription

The setDescription method in the UpdateRealMachineReq class specifies the machine description. This method has the following prototype:

```
void setDescription (const std::string & strDescription)
```

strDescription

Specifies the agent description.

setnodeName

The setnodeName method in the UpdateRealMachineReq class specifies the node name. This method has the following prototype:

```
void setnodeName (const std::string & strnodeName)
```

strnodeName

Specifies the IP address or DNS name of the machine. If this value is not specified, the machine name is used as the node name.

setPortNumber

The setPortNumber method in the UpdateRealMachineReq class specifies the port number. This method has the following prototype:

```
void setPortNumber (int iPortNumber)
```

iPortNumber

Specifies the IP port number EWA r11.3 uses to listen for traffic.

[setCharacterCode](#)

The `setCharacterCode` method in the `UpdateRealMachineReq` class specifies the character code. This method has the following prototype:

```
void setCharacterCode (eCharacterCode eCharCode)
```

eCharacterCode

Specifies the ASCII or EBCDIC character code. Valid character codes are `CC_ASCII` `CC_EBCDIC`. This is valid only for EWA r11.3.

[setKeyToAgent](#)

The `setKeyToAgent` method in the `UpdateRealMachineReq` class specifies the key to agent value. This method has the following prototype:

```
void setKeyToAgent (const std::string & strKeyToAgent)
```

strKeyToAgent

Specifies the key-to-agent string. The `security.cryptkey` parameter on the EWA r11.3 agent should match this encryption key. If these values do not match, encryption will fail and communication will not be allowed to the EWA r11.3 agent. A maximum of 16 characters or 32 hexadecimal digits may be specified. This is valid only for EWA r11.3.

[setKeyFromAgent](#)

The `setKeyFromAgent` method in the `UpdateRealMachineReq` class specifies the key-from-agent value. This method has the following prototype:

```
void setKeyFromAgent(const std::string & strKeyFromAgent)
```

strKeyFromAgent

Specifies a key-from-agent string for a Cybernation agent on z/OS. This is valid only for a EWA r11.3 and only if `encryption_type` is other than “none” or “default”.

[setAgentName](#)

The `setAgentName` method in the `UpdateRealMachineReq` class specifies the agent name. This method has the following prototype:

```
void setAgentName (const std::string & strAgentName)
```

strAgentName

Specifies the agent instance name. Encryption settings must be identical for all machines defined with the same “`agent_name`”. This is valid only for EWA r11.3.

[setHeartbeatFrequency](#)

The setHeartbeatFrequency method in the UpdateRealMachineReq class specifies the heartbeat frequency. This method has the following prototype:

```
void setHeartbeatFrequency (int iHeartbeatFrequency)
```

iHeartbeatFrequency

Specifies the frequency with which you want the scheduler to send the heartbeat signal. This is valid only for EWA r11.3.

setHeartbeatAttempts

The setHeartbeatAttempts method in the UpdateRealMachineReq class specifies the heartbeat attempts. This method has the following prototype:

```
void setHeartbeatAttempts (int iHeartbeatAttempts)
```

iHeartbeatAttempts

Specifies the number of heartbeat signals the scheduler attempts before it sends an SNMP message indicating EWA r11.3 inactivity. This is valid only for EWA r11.3.

setEncryptionType

The setEncryptionType method in the UpdateRealMachineReq class specifies the encryption type. This method has the following prototype:

```
void setEncryptionType (eEncryptionType eEncryptType)
```

eEncryptType

Specifies the encryption type for the EWA r11.3 system agent. This is valid for EWA r11.3 and machine type MR_EWAA only. The following are the valid enumerated encryption types:

- ET_NONE—encryption disabled
- ET_DEFAULT—default encryption key and type
- ET_AES—AES 128-bit encryption

setAgentOS

The setAgentOS method in the UpdateRealMachineReq class specifies the agent operating system. This method has the following prototype:

```
void setAgentOS (eAgentOSType eAgentOS)
```

eAgentOS

Specifies the agent operating system. This is valid only for EWA r11.3 and only machine type MR_EWAA. The following are the valid enumerated agent OS types:

- AO_aix
- AO_hpx

- AO_linux
- AO_openvms
- AO_os400
- AO_solaris
- AO_tandem
- AO_windows
- AO_zos

[setProvision](#)

The `setProvision` method in the `UpdateRealMachineReq` class specifies whether to Provision an EWA agent. This method has the following prototype:

```
void setProvision (bool bProvision)
```

bProvision

Indicates whether the Data Center Automation (DCA) Manager should be requested to provision an EWA agent on this machine.

[setAdministrator](#)

The `setAdministrator` method in the `UpdateRealMachineReq` class specifies the administrator ID. This method has the following prototype:

```
void setAdministrator (const std::string & strAdministrator)
```

strAdministrator

Specifies the administrator user ID to use for provisioning the EWA agent through DCA.

UpdateResource

UpdateResource updates a virtual resource. UpdateResource is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the UpdateResource API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    UpdateResourceReq req;
    req.setName("MyResource");
    req.setDescription("New description of MyResource");
    req.setMachine("mach01");
    req.setAmount(20);
    UpdateResourceRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

UpdateResourceReq

extends
 ApiRequest
header
 AsPublicSDK.h

execute

The execute method in the UpdateResourceReq class performs the API call. This method has the following prototype:

`execute(AsApi & api, UpdateResourceRsp & rsp)`

api

Specifies an AsApi object.

rsp

Specifies an UpdateResourceRsp object.

setName

The setName method in the UpdateResourceReq class specifies the resource name. This method has the following prototype:

`void setName(const std::string & str resourceName)`

str resourceName

Specifies the resource name.

setMachine

The setMachine method in the UpdateResourceReq class specifies a resource machine. This method has the following prototype:

`void setMachine(std::string str Machine)`

str Machine

Specifies a machine ID for the resource.

setAmount

This setAmount method in the UpdateResourceReq class specifies the resource amount. This method has the following prototype:

```
void setAmount(int iAmount)
```

iAmount

Indicates the number of resources.

setAmount

This setAmount method in the UpdateResourceReq class specifies the resource amount. This method has the following prototype:

```
void setAmount(const std::string & strAmount)
```

strAmount

Specifies the resource amount. This overloaded version of setAmount lets you increment or decrement the resource amount using plus or minus prefixes. For example, setAmount("+3") would increment the resource amount by 3.

setDescription

The setDescription method in the UpdateResourceReq class specifies a resource description. This method has the following prototype:

```
void setDescription(const std::string & strDescription)
```

strDescription

Specifies the resource description.

UpdateVirtualMachine

UpdateVirtualMachine creates a virtual machine. This API is used by JIL to insert virtual machines. The UpdateVirtualMachine API verifies if the caller has Create access to the specified machine. It then updates the machine and updates audit information.

UpdateVirtualMachine is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the UpdateVirtualMachine API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    UpdateVirtualMachineReq req("MyVirtualMachine");
    req.appendMachine("machine1");
    req.appendMachine("machine2");
    UpdateVirtualMachineRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

UpdateVirtualMachineReq

extends
 ApiRequest
header
 AsPublicSDK.h

UpdateVirtualMachineReq

The UpdateVirtualMachineReq constructor configures the request. This constructor has the following syntax:

`UpdateVirtualMachineReq (const std::string & strMachineName)`

strMachineName

Specifies the virtual machine name.

appendMachine

This appendMachine method in the UpdateVirtualMachineReq class adds a component machine to the virtual machine definition. This method has the following prototype:

`void appendMachine (const std::string & strComponentName, const int max_load, const double factor)`

strComponentName

Specifies the component (real) machine name.

max_load

Specifies the component (real) machine name max_load.

factor

Specifies the component (real) machine name factor.

appendMachine

This appendMachine method in the UpdateVirtualMachineReq class adds a component machine to the virtual machine definition. This method has the following prototype:

`void appendMachine (const std::string & strComponentName)`

strComponentName

Specifies the component (real) machine name.

execute

The execute method in the UpdateVirtualMachineReq class performs the API call. This method has the following prototype:

```
execute (AsApi & api, UpdateVirtualMachineRsp & rsp)
```

api

Specifies an AsApi object.

rsp

Specifies an UpdateVirtualMachineRsp object.

[setSyntaxChecker](#)

The setSyntaxChecker method in the UpdateVirtualMachineReq class verifies the machine definition. This method has the following prototype:

```
void setSyntaxChecker (bool bCheck)
```

bCheck

Passes TRUE to perform only a syntax check.

[UpdateVirtualMachineRsp](#)

extends

ApiResponse

header

AsPublicSDK.h

[setResponse](#)

The setResponse method in the UpdateVirtualMachineRsp class returns the count of component names following an update. This method has the following prototype:

```
setResponse (Response * rsp) [virtual]
```

rsp

Specifies the index into the real machine list.

UserUpdateAlarm

UserUpdateAlarm allows users to update the specified alarm to closed, acknowledged, or open. The user field in the alarm table is set to the * user as indicated in the request. Alarms are not secured but the user that updates the alarm is tracked.

UserUpdateAlarm is a CAT1 API.

Executing this API can result in the following exceptions:

- ApiExceptionWithInfo
- ApiTimeout
- CredsNotAuthorized
- FailedToSetCreds
- InternalResponseError

Example: Sample Code for Executing the UserUpdateAlarm API

```
AsApi api;
try {
    api.openRemote(waae_port, waae_server);
    UserUpdateAlarmReq req;
    req.setEoid("ACE000010k00");
    req.setResponse("The is the response comment");
    req.setState(AS_ALARM_CLOSED);
    UserUpdateAlarmRsp rsp;
    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cerr << "ApiExceptionWithInfo thrown." << std::endl;
    std::cerr << e.what() << std::endl;
}
catch(ApiTimeout&) {
    std::cerr << "ApiTimeout thrown." << std::endl;
}
catch(CredsNotAuthorized&) {
    std::cerr << "CredsNotAuthorized thrown." << std::endl;
}
catch(FailedToSetCreds&) {
    std::cerr << "FailedToSetCreds thrown." << std::endl;
}
catch(InternalResponseError&) {
    std::cerr << "InternalResponseError thrown." << std::endl;
}
api.close();
```

UserUpdateAlarmReq

extends

 ApiRequest

header

 AsPublicSDK.h

execute

The execute method in the UserUpdateAlarmReq class performs the API call. This method has the following prototype:

`void execute (AsApi &api, SimpleRsp &rsp)`

api

Specifies an AsApi object.

rsp

Specifies a SimpleRsp object.

setEoid

The setEoid method in the UserUpdateAlarmReq class sets the eoid that identifies the alarm. This method has the following prototype:

`void setEoid (const std::string &eoid)`

eoid

Specifies the eoid that identifies the alarm.

setResponse

The setResponse method in the UserUpdateAlarmReq class specifies the response that is a free form user comment about the alarm action. This method has the following prototype:

`setResponse (const std::string & response)`

response

Specifies the response.

setState

The `setState` method in the `UserUpdateAlarmReq` class sets the updated state of the alarm. This method has the following prototype:

```
void setState (eAlarmState state)
```

state

Specifies the updated state of the alarm. Possible values are:

- `AS_ALARM_OPEN`
- `AS_ALARM_ACKNOWLEDGED`
- `AS_ALARM_CLOSED`

WebParameterItem

The WebParameterItem class is the container class for specifying web parameters in job definitions.

Example: Sample Code for Executing the WebParameterItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_WBSVC);
    req.set(KW_MACHINE, (std::string) "localhost");
    req.set(KW_JOB_CLASS, (std::string) "myClass");
    req.set(KW_ENDPOINT_URL, (std::string) "myEndPoint");
    req.set(KW_TARGET_NAMESPACE, (std::string) "myNamespace");
    req.set(KW_WSDL_OPERATION, (std::string) "myOperation");
    req.set(KW_WSDL_URL, (std::string) "myUrl");

    // Set up the web_parameter of the job definition and
    // attach it to the base job definition.
    WebParameterItem wp;
    wp.set(KW_WEB_PARAMETER, (std::string) "parm1=p1");
    wp.set(KW_WEB_PARAMETER, (std::string) "parm2=p2");
    req.add(&wp);

    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_ENDPOINT_URL, KW_TARGET_NAMESPACE,
KW_WEB_PARAMETER};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
```

```
req.execute(api, rset);
while( rset.hasMore() ) {
    GetJobsWithFilterRsp rsp;
    rset.next(rsp);
    for (int i=0; i < rsp.getAttributes(); i++) {

        // Display keywords that are of type string.
        if( rsp.getAttributeType(i) == AtString ) {
            std::cout << "String attribute [" <<
            rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
                rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
        }

        // Display job items.
        else if( rsp.getAttributeType(i) == AtJobItem ) {
            autosys::JobItem myJobitem = rsp.getJobItem(i);
            std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
            for( int j=0; j < myJobitem.getAttributes(); j++ ) {
                // Only displaying jobitem keywords that are of type string.
                std::cout << "Jobitem string attribute [" <<
                rsp.getAttributeNameString( (eKeywords) keywords[j]) << "] has value [" <<
                    myJobitem.getString( (int)keywords[j] ) << "]" <<
                std::endl;
            }
        }
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
std::endl;
}
```

ZOsConditionCodeItem

The ZOsConditionCodeItem class is the container class for specifying condition code arguments in job definitions.

Example: Sample Code for Executing the ZOsConditionCodeItem API

```
InsertJobReq req;
InsertJobRsp rsp;
try {
    req.set(KW_JOB_NAME, (std::string) "myjob");
    req.set(KW_JOB_TYPE, (int) TYPES_WBSVC);
    req.set(KW_MACHINE, (std::string) "localhost");

    // Set up the condition_code of the job definition and
    // attach it to the base job definition.
    ZOsConditionCodeItem zcc;
    zcc.set(KW_RC, std::string("9"));
    zcc.set(KW_RESULT, EN_ZOSRESULT_SUCCESS);
    zcc.set(KW_ACTION, EN_ZOSACTION_STOP);
    zcc.set(KW_PROC_STEP, (std::string) "PROC");
    zcc.set(KW_PROGRAM, (std::string) "PROGRAM");
    zcc.set(KW_STEP_NAME, (std::string) "STEP1");
    req.add(&zcc);

    req.execute(api, rsp);
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}

// Now retrieve the job that was inserted.

// Select keywords to be returned from GetJobsWithFilter.
int myKeywords[] = {KW_JOB_NAME, KW_MACHINE, KW_CONDITION_CODE};
std::vector<int> attributes( myKeywords, myKeywords + sizeof(myKeywords) /
sizeof(int) );

// Create filter to select only jobs that begin with "my".
FilterString* name_filter = new FilterString(J_JobName, "my%");

try {
    GetJobsWithFilterReq req;
    req.setAttributes(attributes);
    req.setFilter(name_filter);

    GetJobsWithFilterRspSet rset;
    req.execute(api, rset);
    while( rset.hasMore() ) {
```

```
GetJobsWithFilterRsp rsp;
rset.next(rsp);
for (int i=0; i < rsp.getAttributes(); i++) {

    // Display keywords that are of type string.
    if( rsp.getAttributeType(i) == AtString ) {
        std::cout << "String attribute [" <<
        rsp.getAttributeNameString(rsp.getAttributeName(i)) << "] has value [" <<
        rsp.getString(rsp.getAttributeName(i)) << "]" << std::endl;
    }

    // Display job items.
    else if( rsp.getAttributeType(i) == AtJobItem ) {
        autosys::JobItem myJobitem = rsp.getJobItem(i);
        std::vector<int> keywords =
myJobitem.getVectorOfKeywordConstants();
        for( int j=0; j < myJobitem.getAttributes(); j++ ) {
            // Only displaying jobitem keywords that are of type string.
            std::cout << "Jobitem string attribute [" <<
            rsp.getAttributeNameString( (eKeywords) keywords[j] ) << "] has value [" <<
            myJobitem.getString( (int)keywords[j] ) << "]" <<
            std::endl;
        }
    }
}
catch(ApiExceptionWithInfo& e) {
    std::cout << "ApiExceptionWithInfo thrown. Message follows: " << e.what() <<
    std::endl;
}
```


Index

A

 ApiException • 39
 ApiRequest • 26, 55
 ApiResponse • 56
 ApiResponseSet • 57
 application
 building (C++) • 43
 building (Java) • 30
 deployment • 20
 application servers
 defined • 21
AsConstants • 68
AsException/AsRunTimeException (Java) • 26
AsMessage • 90
Authorization • 23

C

 C++ SDK
 using • 37
 calendar APIs, common
 DeleteEntireCalendarDate (C++) • 541
 DeleteEntireCalendarDate (Java) • 117
 GetCalendarNames (C++) • 587
 GetCalendarNames (Java) • 170
cat1 API
 executing (C++) • 40
 executing (Java) • 27
cat2 API
 executing (C++) • 41
 executing (Java) • 29
cycle calendar APIs
 DeleteCycleCalendarDays (C++) • 539
 DeleteCycleCalendarDays (Java) • 114
 GetCycleCalendarDays (C++) • 590
 GetCycleCalendarDays (Java) • 172
 InsertCycleCalendarDays (C++) • 730
 InsertCycleCalendarDays (Java) • 730
 ModifyCycleCalendarDays (C++) • 792
 ModifyCycleCalendarDays (Java) • 404

D

 Delete Job (C++) • 552
Deployment • 20

E

 event APIs
 SendEvt (C++) • 826
 SendEvt (Java) • 470
 UnsendEvt (C++) • 841
 UnsendEvt (Java) • 484
 exception APIs
 AsBadAttributesException • 63
 AsBadFilterFieldsException • 65
 AsErrorException • 75
 AsException • 76
 AsExternalValidationException • 78
 AsFieldValidationException • 79
 AsGeneralErrorException • 81
 AsInitializationException • 82
 AsMachineDeletionException • 88
 AsNoAttributesException • 90
 AsObjectDoesNotExistException • 92
 AsRuntimeException • 94
 AsSecurityException • 95
 AsTimeoutException • 96
 AsUnknownErrorException • 98
 extended calendar APIs
 DeleteExtendedCalendarDays (C++) • 543
 DeleteExtendedCalendarDays (Java) • 119
 GetExtendedCalendarDays (C++) • 594
 GetExtendedCalendarDays (Java) • 174
 GetGeneratedExtendedCalendarDays (C++) • 607
 GetGeneratedExtendedCalendarDays (Java) • 183
 InsertExtendedCalendarDays (C++) • 732
 InsertExtendedCalendarDays (Java) • 329
 ModifyExtendedCalendarDays (C++) • 794
 ModifyExtendedCalendarDays (Java) • 407
 RegenerateExtendedCalendar (C++) • 809
 RegenerateExtendedCalendar (Java) • 430
 external instance APIs
 DeleteExternalInstance (C++) • 545
 DeleteExternalInstance (Java) • 121
 GetExternalInstances (C++) • 598
 GetExternalInstances (Java) • 177
 InsertUpdateExternalInstance (C++) • 762
 InsertUpdateExternalInstance (Java) • 357

F

filtering • 34

filters

- JobFilterAnd (Java) • 223
- JobFilterBool (Java) • 254
- JobFilterChar (Java) • 258
- JobFilterDate (Java) • 225
- JobFilterInt (Java) • 228
- JobFilterOr (Java) • 224
- JobFilterString (Java) • 235

G

GetAdapterJobStats (C++) • 572

GetAdapterJobStats (Java) • 165

GetGlobals (C++) • 618

GetGlobals (Java) • 187

GetIntCodes (C++) • 622

GetIntCodes (Java) • 189

GetObjectNames (C++) • 687

GetObjectNames (Java) • 283

GetSchedulerLog (C++) • 707

GetSchedulerLog (Java) • 294

GetStatistics (C++) • 710

GetStatistics (Java) • 296

GetUniqueNames (C++) • 720

GetUniqueNames (Java) • 302

global binary large object (BLOB) APIs

- DeleteGblob (C++) • 547

- DeleteGblob (Java) • 122

- FetchGblob (C++) • 564

- FetchGblob (Java) • 148

- GetGblobList (C++) • 603

- GetGblobList (Java) • 180

- InsertGblob (C++) • 736

- InsertGblob (Java) • 333

J

job binary large object (BLOB) APIs

- DeleteJblob (C++) • 549

- DeleteJblob (Java) • 124

- FetchJblob (C++) • 567

- FetchJblob (Java) • 151

- InsertJblob (C++) • 738

- InsertJblob (Java) • 336

job box APIs

- DeleteBox (C++) • 530

- DeleteBox (Java) • 107

job definition APIs

- DeleteJob (C++) • 552

- DeleteJob (Java) • 127

- GetGlobalConditionsForJob (C++) • 613

- GetJobsWithFilter (C++) • 642

- GetJobsWithFilter (Java) • 203

- InsertUpdateJob (Java) • 360

- JobProperty • 387

- OverrideJob (C++) • 799

- OverrideJob (Java) • 419

- RemoveOverride (C++) • 811

- RemoveOverride (Java) • 432

Job Definition APIs, DeleteJob • 127

job status reporting APIs

- GetJobLog (C++) • 626

- GetJobLog (Java) • 191

- GetJobRunsWithFilter (C++) • 629

- GetJobRunsWithFilter (Java) • 194

- GetJobStatus (C++) • 640

- GetJobStatus (Java) • 201

job type APIs

- GetJobTypesDetail (C++) • 658

- GetJobTypesDetail (Java) • 272

- InsertUpdateDeleteJobType (C++) • 759

- InsertUpdateDeleteJobType (Java) • 354

L

logging • 33

M

machine APIs

- DeleteMachine (C++) • 556

- DeleteMachine (Java) • 130

- GetMachDefs (C++) • 665

- GetMachDefs (Java) • 274

- GetMachJobs (C++) • 672

- GetMachJobs (Java) • 277

- GetMachRuns (C++) • 677

- GetMachRuns (Java) • 280

- InsertRealMachine (C++) • 746

- InsertRealMachine (Java) • 344

- InsertVirtualMachine (C++) • 784

- InsertVirtualMachine (Java) • 368

P

PingApi (C++) • 806

PingApi (Java) • 425

R

Request/Response Protocol • 21

requests

- ApiRequest • 55
 - Cat1Request (Java) • 100
 - Cat2Request (Java) • 101
- responses
- AbstractResponse (Java) • 45
 - ApiResponse • 56
 - ApiResponseSet • 57

S

SDK localization • 23

standard calendar APIs

- DeleteCalendarDate (C++) • 536
- DeleteCalendarDate (Java) • 112
- GetCalendarDays (C++) • 583
- GetCalendarDays (Java) • 168
- InsertCalendarDate (C++) • 726
- InsertCalendarDate (Java) • 324
- ModifyCalendarDate (Java) • 401

U

user

- impersonating (C++) • 42
 - impersonating (Java) • 30
- UserUpdateAlarm (C++) • 860
- UserUpdateAlarm (Java) • 505