Axeda® Glossary

Version 6.5
October 2012

Axeda Corporation
25 Forbes Boulevard, Suite 3
Foxborough, MA 02035
USA
(508) 337-9200

Document Number: Glossary/6.5/1012

Chapter 1
Using the Glossary

This section provides the following information:

About the Glossary explains the intended audience of this document and summarizes the contents of each chapter contained in this guide.

Getting Help describes where to find answers to your questions about Axeda® products.

Documentation Feedback explains how you can help improve Axeda documentation.

About the Glossary

This section describes the intended audience for the Glossary and presents a summary of the contents of this guide.

Audience

This Glossary of terms is intended for all users of Axeda products. It includes brief explanations of each product as well as technical terms used to describe the products in the documentation. For users not familiar with HTTP and SSL terminology, it also includes definitions of some of the standard terms used in these protocols.

Contents

This booklet contains this introductory section and the glossary, organized in alphabetical order. The explanations of Axeda products are located in the A section, under Axeda products and then presented in alphabetical order for the product names. If you are viewing this document online, you can use the hyperlinks at the top of the glossary to navigate to a letter section. In addition, the cross references (See Also's) are hyperlinks to the definitions of the referenced term. A "Back to Top" hyperlink takes you from the end of a letter section to the beginning of the glossary:

Getting Help

If you still have questions after reading the documentation for your Axeda product, you can contact Axeda at http://help.axeda.com for help or more information.

Axeda Developer Connection

The Axeda Platform is open to the worldwide M2M developer community at the Axeda Developer Connection: http://developer.axeda.com. This site includes lots of sample business applications, examples of building applications, code snippets, and more. Registration is free. See http://developer.axeda.com for complete information.

Documentation Feedback

As part of our ongoing efforts to produce effective documentation, Axeda asks that you send us any comments, additions or corrections for the documentation. Your feedback is very important to us and we will use it to improve our products and services.

Please send your comments to documentation@axeda.com. Thank you for your help.

Axeda Glossary

This glossary is a reference for all Axeda Platform components and for elements of the Axeda Applications. It is organized in alphabetical order. The terms associated with a main topic are described in sub-entries of a main entry. This glossary is long, so to help you navigate, the following links are provided. Cross-references throughout this document are hyperlinks and shown *in this font and color*. Click a letter to go directly to the start of the section:

A B C D E F G H I J L M N O P Q R S T U V W

A

Access Rights (Axeda Policy Server)

A required component of filters for permissions in Axeda® Policy Server, access rights define how an Axeda Gateway or Axeda Connector Agent addresses an action sent from the Axeda® Platform. Each permission to perform an action is defined with an access right, which may be specific to the policy for the asset or inherited from a policy for a parent. For example, a permission can be set to Always Ask for Approval in a policy of a parent, and then (if not locked in the parent policy), set to Never Allow in a child policy.

When the Agent receives an action request from the Axeda Platform (for example, the Platform wants to register a script on that asset), the Agent refers to its known policy to determine how to handle or not handle the action.

The possible access rights for permissions are as follows:

- Always Allow - the Agent can execute these permissions without asking for approval or sending the action information to Axeda® Policy Server. To see which actions of Always allow rights were performed on an asset, you need to refer to the log file.
- Ask for Approval - the Agent forwards the action and its parameters to Policy Server for approval, as well as a status message to the Axeda Platform. When Policy Server receives the action, it sends an e-mail to the address specified for the policy and then stores the action request in the Pending Requests queue. The action request will remain shown in the Pending Request page until it is approve or denied, or it times out. (If timed out, the action is denied and needs to be re-requested, if desired. A message is logged in the audit log for Policy Server.

  If approved or denied, the action request is removed from the Pending Requests page. A message regarding the approval or denial is written to the audit log. Policy Server sends its response (accept or deny) to the asset. The asset sends another status message to the Axeda Platform to identify whether the action request was approved or denied. If the action request was approved, the asset then processes the action.
- Never Allow - the Agent will not execute these permissions and will send information for these requests to Policy Server only when Never Allow actions are requested from the Axeda Platform. To see which asset-initiated actions of Never Allow rights were denied on an asset, you need to refer to the log file for the Axeda Agent that is running on the asset (for example, the log file for Axeda® Gateway is xGate.log).
- Reset to Parent (does not appear when setting permissions in the Global policy, or for permissions that are not available in the policy of the parent) - Policy Server will change the current access right to that set in the policy of the parent. If this permission is not available in the policy of the parent, this access right is not available for selection.

Note: *The Ask for Approval access right is not available for some permissions where setting this access right could cause the Agents to send too many requests for approval because the actions are frequently attempted. For example, the Set Data Items action supports only the Never Allow and Always Allow access rights. It does not support Ask for Approval.*

For more information about Axeda Policy Server, refer to the *Axeda® Policy Server Administration Guide*. Additional information about Policy Server can be found in the following entries in this glossary: *Action (Axeda Policy Server)*, *Locking Permissions (Policy Server)*, *Permissions (Policy Server)*, *Policy (Policy Server)*, *Privileges (Policy Server)*, *Profiles (Policy Server)*, and *Role (Policy Server)*.

Acknowledge Alarm

In the Axeda Service application or in a display served from the Axeda Connector Web Server, acknowledging an alarm is the act of an operator clicking a button or link in a display or in a page of the Axeda Service application to tell the Platform that he/she has seen the report of an alarm. *See Also: Alarm State*

Action (Axeda Applications)

In the context of the Axeda Platform and Axeda Applications, an action is an operation performed as the result of a condition evaluating to true, such as the value of a data item being greater than a specified number. For example, if a temperature value is over 80, then the action could be to generate an alarm.

The Axeda Platform provides multiple ways to detect conditions and trigger actions based on the results of evaluating the condition. Using the Axeda Configuration application, you can create Expression Rules and State Machines that evaluate conditions and run actions based on the results. See also *Actions for Expression Rules* .

You can also create separate actions in the Axeda Configuration application and then run them from Asset dashboards in the Axeda Service application. These actions are sometimes referred to as “legacy” actions since they were the original actions for the Platform.

Using Axeda Builder, you can create a project in which you associate actions with triggers in logic schemas, data loggers, or alarm loggers.

For more information, refer to the online help for the Axeda Applications and Axeda Builder and to the *Axeda® Platform Web Services Developer’s Reference Guide*.

Actions for Expression Rules

Actions identify the operations you want Axeda Enterprise to perform after evaluating the conditions for (IF) an expression rule. You can configure actions in the Then and Else expressions of an expression rule. The actions you can configure for an expression rule vary based on the type of trigger selected for the expression rule.

Actions are the functions supported for a trigger type. To configure an action for an expression, you add the action and then specify your information. For example, you may define the action SendDataItem() for a Then expression, and specify that you want the Platform to set the value for data item “dataItemABC” at the asset.

Action (Axeda Policy Server)

In the Axeda Policy Server context, an action is any activity that Axeda Gateway or Axeda Connector can perform, such as sending data item values, files, or alarms to the Axeda Platform, registering scripts on the asset, or transferring files between the asset and the Platform. Actions can be initiated in two ways:

- From the Axeda Platform — Actions the Platform is attempting to perform on an asset may be manually or automatically initiated. For example, a service technician wants to restart the asset immediately to fix a problem and sends a Restart action to the Axeda Gateway or Connector Agent that is running on the asset. Alternatively, an expression rule evaluates an alarm event from the Gateway or Connector Agent and if the condition is true, the Axeda Platform sends a Restart action to the Agent.
- From the Agent configuration — Actions that the Axeda Gateway or Connector Agent is attempting to perform on the asset as part of the project configuration of that Agent. For example, the project may be configured to send its data item values to the Axeda Platform every hour or upload a log file based on an alarm.

Actions can be managed and audited by Axeda Policy Server. Axeda Policy Server contains entries for all known actions, based on the actions supported by the connected Axeda Gateway and Axeda Connector Agents. If an Agent registers with support for a new action, that action appears in Axeda Policy Server with the default access right of Ask for Approval.

Using the Policy component of the Policy Server application, you can create and edit permissions for actions. Each policy will contain at least one permission for an action: the default permission (which is modifiable). You can create more permissions for an action. For example, your policy can specify an action of Package that has an Ask for Approval permission for any package; an Always Allow permission for a Package of name Maintenance, version 1.4.35, as well as for a Package of name Trouble, version 9.9.59.4; and a Never Allow permission for a Package of name Unsafe, version 3.2.1.

For more information about using Axeda Policy Server, refer to the *Axeda® Policy Server Administration Guide*. Additional information about Policy Server can be found in the following entries in this glossary: *Access Rights (Axeda Policy Server)*, *Locking Permissions (Policy Server)*, *Permissions (Policy Server)*, *Policy (Policy Server)*, *Privileges (Policy Server)*, *Profiles (Policy Server)*, *Role (Policy Server)*

ActiveMQ

The Axeda Platform ActiveMQ Integration solution makes it possible for the Axeda Platform to send messages and relevant data about any event that occurs there. Customers can subscribe to receive these messages and build custom workflows and new business processes. Messages can be published to ActiveMQ through expression rules that are executed by the Platform. Messages can also be published automatically to a subscription queue, using the *Event Subscription Service (ESS)*. For information specific to the Event Subscription Service, refer to the *Axeda® Platform Event Subscription Service Integration Guide*.

Ad Hoc Report (Axeda Report)

An Ad Hoc report is a query type of report, created using Query Studio. Typically, Ad Hoc reports serve the purpose of generating a report for a specific purpose, using limited formatting and requesting a limited amount of information to display. Ad Hoc reports query the database for the information requested and present it in the format configured in Query Studio. Your organization must have the appropriate license for Query Studio for Ad Hoc reports to be available. In addition, you must be a Named User for Query Studio to be able to create new Ad Hoc reports or edit existing ones. For details about Query Studio, refer to the online help available within that tool. *See also Axeda® Report and Axeda® Dashboard (Axeda® Service Intelligence)* and *Report*.

Agent-side Authentication

The Axeda Gateway and Axeda Connector Agents support local authentication servers, such as a RADIUS server. For organizations that monitor assets and require third parties (for example, the manufacturers of the monitored assets) to go through their local authentication server before accessing assets from a remote location (for example, the service center of the manufacturer). When this authentication is enabled, users requesting Remote Sessions must enter credentials specific to the customer site before they can access assets.

This authentication covers not only Remote Sessions but also execution of actions. Once the requested credentials are passed to the Axeda Gateway or Connector Agent prior to executing an action, the Agent passes the credentials to the locally operated, dedicated authentication provided to perform authentication. After receiving the results of the authentication, the Agent grants or denies the requested action.

Using the Axeda Deployment Utility, you can enable agent-side authentication and configure the parameters that the Agent needs to communicate with the authentication server. For information about configuring agent-side authentication, refer to the *Axeda® Deployment Utility User's Guide* or to the online help for the utility.

Alarm

An event, caused by a condition, such as the value of a data item or the results of a calculation, deviating from a defined value or range of values. That the event is an *alarm* indicates that something has gone wrong. For example, it can indicate that an error has occurred, a switch must be reset, a temperature must be adjusted, or even that a machine must be shut down and then repaired.

Alarm Definition (AxedaApplications)

In the AxedaApplications, an Alarm Definition assigns a name to a set of properties for an alarm. Alarm definitions are associated with models in the Platform. Alarm definitions can be created manually using the Axeda Configuration application or automatically when a new instance of an alarm is detected by the Platform. From the Axeda Configuration and Axeda Service applications, you can enable or disable and suppress or un-suppress each alarm definition for a model. Note that you must have the appropriate privileges to be able to perform these operations.

In addition to the name, the set of properties for an alarm definition include:

- Description – Additional information about an alarm, either entered when the alarm definition was created through the Axeda Configuration application or provided with a new instance of an alarm.

  For an Axeda Gateway or Axeda Connector Agent, the description provided with a new instance of an alarm depends on the main category of alarm (analog, digital, custom, or dynamic):

  - Analog Alarm — If the alarm is a Value alarm, the Description is the value entered for the Value Level (LoLo, Low, HiHi, or High) in Axeda® Builder. If this alarm is a Deviation alarm, the Description is the information entered for the Minor Deviation or Major Deviation in Builder.
  - Digital Alarm — The Description is the text entered for the Message in the configuration of the Digital Alarm Style in Builder.
  - Custom alarms — The Description is the text entered for the Message in the configuration of the Custom Alarm Style in Builder.
  - *Dynamic Alarms* — The Description contains whatever the driver sent for a message.

  For an Axeda IDM Agent, the Description contains the four components of an IDM Agent fault (alarm): code, abstract, description, and detail. The following example shows how they would appear here:

  ```
  ErrorCode=<fault error code>
  Abstract=<fault abstract>
  Description=<fault description>
  Detail=<fault detail>
  ```

  If expression rules have been written for the IDM Agents to take this alarm information and generate new alarms, using the information, then the Description contains only the information provided in the Description component of the fault.

- Attributes — indicates whether or not the attributes of the alarm will be stored in the database.
- *Alarm Disposition* — the current disposition setting for an alarm definition for a model.

Alarm Disposition

The disposition of an alarm refers to how the alarm is handled when it is first detected by the Axeda Platform. Alarm disposition is configurable at both the model and asset levels. The possible dispositions follow:

- Enabled and Unsuppressed — New instances of the alarm are processed normally - their state is set to "Started", they are stored in the Platform, and they are available for evaluation by Expression Rules.
- Disabled — All new instances of the alarm are discarded. These alarm instances are not stored in the Platform nor are they visible in the Axeda Service application. Since they never enter the Platform, rules that are configured to run when these alarms are detected do not run.
- Suppressed — All new instances of the alarm are set to "Closed" upon receipt, and they are visible only in the Historical Alarms page. These alarm instances do not appear in the Alarms module on the Asset dashboards nor do they appear in the Current Alarms page. Like Disabled alarms, Suppressed alarms are not available for evaluation in Expression Rules.

  Note: *With the proper privileges, a Service Technician can suppress alarms for an asset from the Alarms module of the Asset dashboard or from the Alarm Definitions page for the asset in the Service application. If an alarm that was enabled and unsuppressed at the model level has been suppressed at the asset level, the icon appears next to the alarm name. Changing the disposition at the model level while the override is still on does NOT affect the disposition at the asset level. You must remove the override to change the disposition at the asset level.*

See also: Alarm State

Alarm Filter

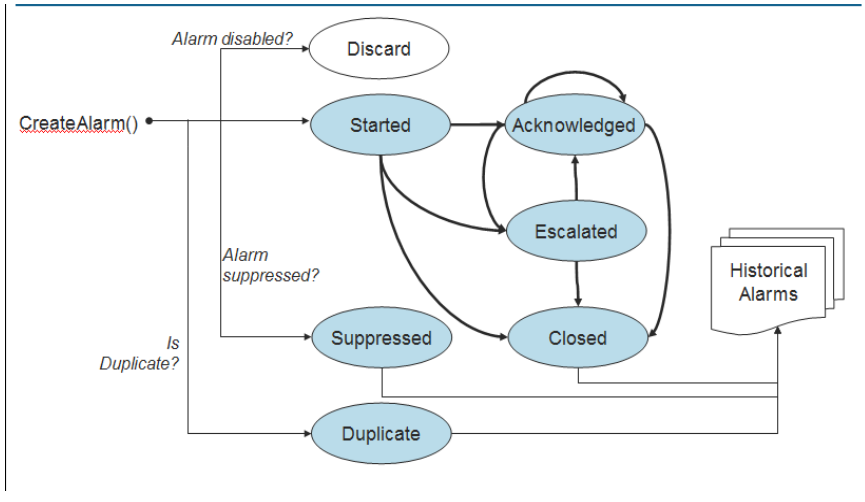
In Axeda Builder projects, a mask for including and excluding alarm events in alarm windows (Web visualization only), sound objects, and triggering events for loggers and logic schemas. For example, you can define an alarm filter that includes digital alarms only, select which digital data to include in the filter, and then apply that filter to a sound object. The sound object will play only when a digital alarm event for the selected digital data occurs.

Alarm Logger

The configuration of a log file or a collection of alarm events that will be sent to the Axeda Platform as a history of alarms generated by a project. There are multiple ways to configure the alarm logger to send the data to the Axeda Platform. To configure alarm loggers, use the **Alarm Logger Configuration** dialog box in Axeda Builder.

Alarm State

In the AxedaApplications, the “state” of an alarm describes the status of an alarm. Alarm states can show the progress of problem resolution. For example, the initial state of a new alarm is Started. Once a service technician is ready to work on the problem he or she can set the state to Acknowledged. If the problem requires additional assistance from a development engineer, the technician can set the state to Escalated and notify that engineer. The engineer can indicate that he or she is working on it by setting the state to Acknowledged again. Once the issue is resolved, then the state can be set to Closed. The following figure shows the possible state changes for a Started alarm within the lifecycle of an alarm:



As you can see in this figure, the lifecycle of an alarm has more steps than changes between states. For more details about the lifecycle of alarms, refer to the topic, *Lifecycle of an Alarm*, in the online help for the AxedaApplications.

Alarm Style

In Axeda Builder, an Alarm Style defines a template for the generation of an alarm, based on predefined values or ranges of values for data items. Axeda Builder allows you to configure three types of Alarm Styles; two of these styles are based on the type of data item: Analog Alarm Styles and Digital Alarm Styles. The third type is the Custom Alarm Style. *See Also Alarm Filter*.

Analog Data Item

A data item that returns a floating point (double) value. Note that counters are always analog.

Analytics Queries (Axeda Usage)

The Usage application supports three types of user queries against the Platform.

- A Simple search contains a limited number of filters and is not intended for complex data queries.
- An Advanced search provides for a more advanced search for usage data that can include the use of wildcards, and manufacturer and asset name filters.
- A Target search contains filters you can define to search for usage data that fell outside a specified range. For example, you can search the Platform to determine whether an organization is using an asset more or less than his contract allows, which could indicate that he may need a different model or a different contract.

Note: *Note: All usage queries are performed in GMT (Greenwich Mean Time) to ensure that the all usage queries will return consistent results and usage data, regardless of the time zone selected by the actual Applications user who runs the query.*

Asset (“device”)

In the world of business, an asset is generally anything that a company owns that is of value. Assets are often categorized as tangible and intangible. It is the tangible assets that the Axeda® Platform and the Axeda Applications can help you manage. Tangible assets are also called Property, Plant, and Equipment (PPE) or fixed assets. The concept of fixed in the business or accounting sense is that they are purchased for long-term, continuing use by the business. These fixed assets can be stationary (installed in one location and rarely moved) or mobile (consistently moving from one location to another). In addition, a stationary asset could become mobile, and vice versa. In the Platform, stationary and mobile assets can coexist and be managed as required.

In general in the Platformenvironment, assets are any Property, Plant, and Equipment that you want the Platformto monitor. For the Platformto monitor an asset, communication between the Platform and the asset must exist. It does not have to be constant communication; the timing of the communication can be set up to suit the needs of your organization. In addition, assets can communicate with the Platformover a variety of communication media, including wireless networks.

Your assets might be intelligent devices, capable of running an Axeda® Gateway Agent, an Axeda® Connector Agent, or an Axeda® IDM Agent (formerly Questra® Service Agent) that serves as the medium that enables communications with the Platform. Axeda Gateway and Axeda Connector Agents can also run scripts, process software management packages, and upload and download files. All Axeda Agents can handle information for remote desktop sessions with the assets where they are running.

Your assets might be Property, Plant, and Equipment (PPE) that have tracking devices, such as an Edge device. In this scenario, the tracking devices communicate directly with the Platform over wireless networks. The PPE and the tracking devices are treated as separate assets in the Platform. However, you can associate assets with one another. By associating tracking assets with the assets where they are installed, you enable the Platformto make data and alarms sent in by tracking assets available in the Asset dashboards of the PPE being tracked. Each asset is defined within a model. Some or all assets of a model can be organized in a single asset group. User groups (and their users) are granted privileges to selected asset groups.

An asset can connect directly to the Platform (standalone), or connect through an Axeda® Gateway asset (gateway-managed assets). You can define an asset as a gateway and configure it to manage the assets that are connected to it. Axeda Gateway must be installed on each gateway asset. and must be configured to manage the assets connected to it.

Asset or Model Association

The ability to associate assets or models with each other allows you to take information from multiple agents or wireless devices (sources) and build a purely virtual "*Composite Asset* " (target). These associations include connecting the data items sent by the source assets with data items configured for the target Composite Assets. To create asset or model associations, you can use either the Axeda PlatformWeb services (refer to the *Axeda® Platform Web Services Developer's Reference Guide*) or the Axeda Configuration application.

A Composite Asset is available in the Axeda Applications in the same ways as any other asset. You can see information about it in an Asset dashboard (Axeda Serviceapplication). You can run reports against it, and you can execute expression rules against it.

For the purposes of the rest of this definition, the assets running an Axeda Agent and the wireless tracking devices are referred to as the "source assets" and the assets they are monitoring (including Composite Assets) are referred to as the "target assets".

Three association situations can exist:

- No association at all. In this situation, data, alarms, and files uploaded by a source asset appear in the Asset dashboard for the source asset and are stored with the source asset in the Platform. The data, alarms, and uploaded files are not visible in the Asset dashboard of the target asset.
- Asset association only, where a single source asset is associated with a single target asset (or with itself). In this situation, the alarms and mobile locations appear in the Asset dashboard for the target asset and do not affect the source asset. Uploaded files appear in the Asset dashboard for both the source and target assets.

  Whether or not a source model is associated with itself, if an asset is associated with itself, the data items are displayed in the Asset dashboard for the source asset. If the source model is associated with itself, then the Platform uses the configured associations of data items for the models and also for the assets.
- Model association, where a model of source asset is associated with a model of target asset (or with itself). Model associations require data item associations. At least one data item from the source asset model must be associated with a data item of the target asset model. For example suppose that the Input_1 data item of the source asset model is associated with the Temp data item of the target asset model. With these associations, alarms and mobile locations are forwarded to the target asset and do not affect the source asset. The Input_1 data item from the source asset is forwarded to the Temp data item of the target asset; all other data items are discarded. If you want to see Input_1 in the Asset dashboard of the source asset, you need to associate the source asset or the source model with itself.

  If you want to see data items, and alarms, and mobile locations in the Asset dashboards of both the source and target assets, you need to associate the source asset with itself or the model of the source asset with itself.

One to Many, Many to One Associations

A single source asset or model can be associated with multiple target assets or models. The reverse is also true: a single target asset or model can be associated with multiple source assets or models. For example, source asset ModelABC can be associated with target models Container123, Container456, and Container789. In addition, target model Container123 can be associated with source asset ModelABC and source asset ModelXYZ.

A many-to-one association can result in a virtual, *Composite Asset* .

Asset Condition

An asset condition is a user-defined entity that is comprised at minimum of a name, a severity, and an image. Asset conditions can also be associated with events, alarms, and data expressions. Such associations are optional but strongly recommended as the way to respond to alarms, data expressions, or the two asset missing events (Axeda Platform to asset connection missing and Gateway to managed asset connection missing). Refer to the online help for the Axeda Applications for more information.

Asset Group (Axeda Configuration)

An Asset Group is a set of assets or devices that an organization wants to associate with one another for business purposes. Each Asset Group can be comprised of assets only, of another Asset Group only, or a combination of assets and one other Asset Group. Although an asset group can be assigned to only one other asset group (its "*Parent Asset Group*"), a single asset can be assigned to multiple Asset Groups. An asset group within another Asset Group is considered a "child" of that "parent" Asset Group.

When the Platform is first installed, the *Root Asset Group* and the *Default Model Group* are created by default. The Default Model Group has the Root Asset Group as its parent. Both of these groups can contain multiple Asset Groups and assets. They are the only asset groups that can contain more than one asset group. Neither group can be deleted.

Each time a model is created (whether automatically or manually), the Platform creates a default asset group for the model, using the word "Default" and the name of the model as the name of the asset group. All of these asset groups are created in the Default Model Group. None of these groups can be deleted or modified.

Assets are assigned to Asset Groups, and users are assigned to user groups. Asset Groups are assigned to user groups such that only users who are assigned to the assigned user groups can view, modify, and manage the assets in the related Asset Groups.

You can create Asset Groups in the Create an asset group page of the Axeda® Administration application. When the Axeda Platform is configured to allow it, the system automatically creates models, assets, and default model asset groups for all assets that register for the first time without having been added previously to the Platform. The assets can also be assigned to other asset groups, either manually or through the execution of dynamic group definitions.

For information about security for Asset Groups, refer to the topic, *Security for Objects in* Axeda® Connected Product Management Applications , in the online help for the Axeda Applications. For information about security for these objects when Delegated Administration is enabled, refer to the help topic, *Security for Objects in Delegated Administration Units*. When viewing asset groups in a Platform with Delegated Administration enabled, the names of asset groups may have prefixes; for details, refer to the help topic, *Names of Objects in DA Units*.

Asset Group (Axeda Policy Server)

In the Policy Server environment, an asset group is an organizational unit within the hierarchy that exists to facilitate applying the same policy to many assets. When an Axeda Agent running on an asset connects to Policy Server for the first time, its registration message contains a serial number, a model, and a list of supported actions. Policy Server automatically creates an asset group for each model upon registration, using the models as the identifiers (names) for these asset groups. It assigns the asset that is registering to the asset group created. Users with appropriate privileges can change the assignments of assets to asset groups. Refer to the online help for Policy Server for more information.

See also Permissions (Policy Server) File Upload Manager, *Inheritance (Policy Server)*, *Notification (Policy Server).*

Asset State / State Monitoring

An asset state is a user-defined entity that is set or reset by the value of a data item. Typically, an asset state reflects the operational status of an asset or of one of its components. For example, if Axeda Gateway is monitoring a printer, it could be configured to set the asset state Out of Toner when it receives a value of 0 from the printer for the data item, Toner Level. A State Monitor is a group of mutually exclusive asset states. You configure monitors and asset states for a project using Axeda Builder. Refer to user guides for the Axeda Agents and to the online help for Builder for details on asset state monitoring.

Audit Log (Policy Server)

In the Policy Server environment, audit logs are text files (*.txt) stored in the *audit* subdirectory of the Policy Server installation directory. A new audit log is created for each day, starting at midnight and then closing at 23:59 (11:59 PM). An example audit log file name is APM_Audit_2012_01_24.txt.

The types of activities recorded in the audit log include user access (log in, log out), configuration (changes to permissions, for example), and asset communications (request for approval of an action, for example). These types of activities are referred to as audit categories. By default, all of the categories are enabled. If you do not want the audit log to maintain entries for one of these categories, you can disable the category. Refer to the online help for Policy Server for details.

Axeda® Agents

An Axeda Agent is an embedded application that can gather information from assets, communicate that data to the Axeda Platform, and be configured to react to the data. The current Axeda Agents are the Axeda Connector Agent, Axeda Gateway Agent, the Embedded Toolkit, and the Axeda IDM Agents (formerly Questra® Service Agents). Axeda Connector supports EDD and OPC drivers, and can run a single project for multiple assets of the same model. Axeda Gateway supports EDD, OPC, and SNMP drivers, and can run a master project that contains multiple Connector projects for different models of assets (including a gateway model). For complete details about these Axeda Agents, refer to the *Axeda® Connector User's Guide* and the *Axeda® Gateway User's Guide*. For complete details about support for the IDM Agents, refer to the topic, *Axeda® IDM Agents*, in the online help for the Axeda Applications.

Axeda® Artisan

Artisan is a developer tool that can be used for developing Axeda *Custom Object* Scripts and Rich Internet Applications (RIAs). Artisan provides a project structure for keeping Custom Object scripts and RIA code together in a cohesive project. You can lay out a project in a file system structure that is natural for SCM tools like Subversion and GIT, so developers can maintain source code control over their projects. You can also edit Axeda Custom Objects for Rules or Scripto in an advanced, modern IDE with Groovy language support and the Axeda SDK classes available for quick documentation lookup and code completion.

Axeda Artisan also provides one-command deployment, so that a project's scripts and RIAs can be uploaded to the Axeda Platform in a single command, without needing to log in to the Axeda Applications.

> **Note:** *The Artisan developer tool is only available on the Axeda® Developer Connection website (http://developer.axeda.com/)*

Axeda® Builder

Axeda Builder is a Web-enabled, graphical configuration tool that you can use to create and test projects specific to your organization and processes. Using the Axeda Connector and Axeda Gateway Agents, and the Axeda Enterprise Server, you can publish interactive process data to the Internet or a local intranet.

Axeda Builder enables you to create projects that collect data from a wide array of devices and controllers, and then publish that information for viewing on-site or across the Web, all with a consistent look, and without the need for specialized client software. For project displays that will be served from an Axeda Connector device, a supported Web browser is all an operator needs to view and interact with the project. For project displays that will be served from the Axeda Enterprise Server, operators also require access to the Axeda Applications.

Using Axeda Builder, you configure projects with all the information needed to collect and display process data. You can create a wide range of projects, from basic projects containing simple graphical objects and animations, to advanced projects configured with scripts, alarms, custom handlers, business logic, and more.

For more information on Axeda Builder, see the *Axeda® Builder User's Guide*.

Axeda® Codec Server

The Axeda AnyDevice Codec Server (ACS) is a component of the Axeda Platform that bridges the gap between customer devices in the field and the Platform. The primary goal of the ACS is to offload device message encoding and decoding from the Axeda Enterprise Server. For example, the Danlaw Data Logger ACS processes the messages sent by the Danlaw Data Logger devices by means of a Codec implemented specifically for the protocol used by these devices.

Axeda® ConnectedConfiguration

Axeda ConnectedConfiguration is a separately licensed, Connected Product Management feature that enables the Axeda Platform to store, manage, and act upon asset configuration information. Each time an asset's configuration is set, or changes, the new configuration can be processed by the Axeda Platform for validation via a new type of expression rule (Validation Rules), standard expression rule execution, and storage in a configuration management database.

Axeda ConnectedConfiguration allows you to use asset configuration information to determine the current settings for an asset, including its hardware and software configuration, the make and model of its hardware, and more. Any configuration information for an asset can be captured in its asset configuration and stored to the Axeda Platform. Once there, this asset metadata can be used in expression rules, Custom Objects, or Custom Applications. For more details on Axeda ConnectedConfiguration, see the Axeda Applications online help topic, *Axeda® ConnectedConfiguration.*

Axeda® Connected Product Management Applications

The entries that follow list and describe each of the Axeda Connected Product ManagementApplications (abbreviated "Axeda Applications" in the rest of this document), in alphabetical order.

Axeda Administration

Axeda Administration enables configuration of security, including access to assets and the pages of the Applications. For example, administrators can create user groups and assign them privileges to the Applications so that they can see what they need to in order to do their jobs. This application also provides access to system administration tools, such as the Platform's audit log and archive policy. Only users with appropriate privileges can access the Administration application and its various pages.

When you first start your Platform, you use Axeda Administration to configure user authentication, privileges, and security within the applications. Only users and user groups configured in the directory service for your Platform can access the applications.

Axeda® Case Management

The Axeda Case Management application enables service organizations to collaborate with partners and manage and track all activities related to troubleshooting asset issues. Users can view the actions (who, what, when, and results) performed that led to resolutions. Using Case Management, users can diagnose and repair assets daily, use powerful reporting and dashboard capabilities to create and view critical KPIs for every aspect of their organization, measure and track call center metrics, and review asset population metrics by customer, region, and support engineer. For details, refer to the online help for the application.

Axeda Case Management and its associated functionality are licensed separately from the base system. For more information about licensing, go to *http://help.axeda.com*.

Axeda® Configuration

The Axeda Configuration application provides asset management tools that you can use to define business rules that evaluate conditions and based on those evaluations, can convert asset data into actions. For example, you can define business rules for proactive fault detection, consumable restocking, user notification, and usage-based billing. Business rules are comprised of conditions to evaluate, and one or more associated operations to perform on the related asset or within the Platform. For information about the types of rules available, refer to *Expression Rules (Axeda Applications)*, *State Machines*, *Rule Timers*, and *Threshold Rules*.

The asset management tools of this application allow you to enable, disable, suppress, and unsuppress alarms for models, associate models or assets with one another (to form a *Composite Asset*), configure the modules you want to display in the Asset dashboard for assets of a selected model, and manage the organizations, regions, locations, and other information associated with models and assets. For details refer to the online help for the application.

Axeda® Preventive Maintenance

Axeda Preventive Maintenance is a Web-based application that enables users to view the status of maintenance procedures for assets. Using the Axeda Configuration application, users create *maintenance items* that associate one or more data items with counters that measure time or cycle counts or time and cycle of product usage. Service personnel can view charts of maintenance data and visually compare the current counts to configured reminders and targets. They can also view a list of assets currently due for maintenance. For more information, go to *http://help.axeda.com*.

Axeda® Report and Axeda® Dashboard (Axeda® Service Intelligence)

Using Axeda Service Intelligence, users can build and deliver reports and dashboards regarding asset performance, service-level agreements, compliance, and operational metrics. This Axeda bundle includes the Axeda Report and Axeda Dashboard applications. Axeda Service Intelligence provides support for building and delivering reports, ad-hoc query reports, and dashboards that clearly articulate the success metrics of Axeda Applications while delivering reports to customers that clearly show the value of remote service.

For more information, go to *http://help.axeda.com*. *See also Ad Hoc Report (Axeda Report)*, *Query Studio*, and *Report Studio*.

Axeda® Service

The Axeda Service Application provides tools and views that enable you to monitor your assets and systems, proactively avert problems, identify missing assets, track user and asset activity, diagnose conditions, and fix problems on remote assets and systems. The Home page of the application enables you to search for assets and view the current status of their connection to the Axeda Platform. The name of each asset shown in the home page is a link to its Asset dashboard, where you can view information from and about the asset in a variety of configurable modules. You can also take actions from the modules, such as setting a data item, restarting the asset, or starting a remote session with the asset. For complete details, refer to the online help for this application.

Axeda Software Management

Axeda Software Management enables efficient, secure, reliable and cost-effective mass distribution of software modules, such as application and operating systems updates and upgrades, patches, and documentation. Using the Software Management application, project and asset administrators configure the transfer of software and related files to and from the assets monitored by the Axeda Platform. Assets that are running Axeda Gateway and Axeda Connector Agents that have been configured with Software Management support can upload files to and download files from the Platform through Axeda Software Management.

Axeda® Transport

Axeda Transport enables service organizations to connect assets that are offline during normal operations with an Axeda Platform. Service technicians can connect laptop computers to the Axeda Platform and download archives containing data and/or commands from the Platform for the assets they need to service. Once on site and connected to the assets, they can download the data and/or commands to the asset and also upload any data, files, and responses to the server commands. The Axeda Gateway or Axeda Connector Agents running on the laptop computers create an archive with the information from the assets, which the service technicians can later upload to the Platform. Through Axeda Transport, these remote assets can receive software patches, upload their latest diagnostic information, receive feature upgrades, and participate in troubleshooting sessions with the service technicians.

Axeda® Usage

Axeda Usage is designed to support an organization's ability to track the frequency and volume of use that their assets are experiencing in the field. The Axeda Platform supports the ability of manufacturers to service their customer, by providing current and historical data about the state of the asset. To this core competency, the Usage application adds the ability to abstract that "raw" asset data into a set of higher-level, business-oriented data. Instead of presenting data such as the value of a data item called "counter-1," the application abstracts that data into useful information, such as "black-and-white copies made per day." This data can be filtered and grouped by asset, by organization, by location, and so on. The pages of the Usage application show the frequency or volume for particular usage items for particular assets. The pages enable you to flexibly filter and aggregate usage information, (optionally) display billing estimate information, graph, export, and print the data. Filtering enables you to refine the list of data to display. You can filter information by organization, location, model, usage item, and asset group. Filtering limits the list to show only those assets that match the filter criteria. Aggregating enables you to combine related items. For example, rather than showing individual usage information for all assets, you might want to see the total usage for assets at a particular location. Aggregating combines rows of data. Data such as amount of toner remaining, or level of reagent remaining, do not represent usage items. These are consumables, which are not addressed in this release. In addition, SLA monitoring is not addressed in this release.

Axeda® Deployment Utility

Designed for Service Technicians who are deploying assets in the field, this utility enables the technicians to set or edit asset and project information when installing the asset. For example, they can set the asset's date and time, identify the model number and serial number, and adjust settings for communications between the Axeda Agent and Axeda Platform. The technicians can run this utility on the asset that is running the Axeda Agent, or they can run it on a non-agent asset and connect to the agent asset through a serial port or an Ethernet (TCP/IP) connection. All settings configured using the Deployment Utility are saved in a special configuration file on the agent asset called EDeployConfig.xml (Axeda Connector) or xgDeployConfig.xml (Gateway). *As long as this file exists, these changes override any settings in the project configuration files, even if you download a new set of project files.* You can restore default project settings by deleting this configuration file.

Axeda® Enterprise Server

The Axeda Enterprise Server is an application server that serves as the destination for communications with assets running Axeda Gateway, Axeda Connector, or Axeda IDM agents, M2M assets running agents developed with the Axeda Wireless Protocol (AWP), or M2M assets connecting through the Axeda Codec Server. The Enterprise server can store incoming data and run business rules in response to the data. Further, the Enterprise Server can send requests to assets to perform actions, to upgrade the software on the asset, to send data back to the server, or to upload files to the server. Requests come in to the Enterprise Server through a Web Application Server; the Enterprise Server can access data stored in the associated database for retrieval, updates, or other processing. The Axeda Applications enable customers to manage the Enterprise Server as well as monitor and troubleshoot their assets. For more information, refer to the *Axeda® Enterprise Server Installation and Maintenance Guide* for your platform.

Axeda® Platform

The Axeda Platform is a complete M2M data integration and application development platform with infrastructure delivered as a cloud-based service. With the highest levels of scalability and security as well as powerful development tools and flexible APIs, you can quickly build and deliver custom M2M applications for the most demanding requirements and integrate M2M data into your key enterprise applications and systems.

Axeda® Policy Server

Axeda Policy Server is an application server with a web-browser client that enables customers of OEMs to prevent unwanted access to their networks. Customers define policies for their assets, and the Axeda Agents retrieve the policies from the Policy Server when they first register with Policy Server. The policy consists of an action that can be performed (for example,

upload a file) and one or more permissions. Each permission specifically defines the access right to the action and, if appropriate for the action, the scope of the permission (for example, the file or files affected by the permission).

The Policy Server product provides a Web-based application for managing policies, asset groups, pending requests, and remote sessions for assets. The application also provides an Administration component for setting up access to the application components. *See also Policy (Policy Server)*, *Permissions (Policy Server)*, *Notification (Policy Server)*, *File Upload Manager, Privileges (Policy Server)*, *Profiles (Policy Server)*, and *Role (Policy Server)*.

Axeda Policy Server and its associated functionality are licensed separately from the base Axeda Platform. For more information, go to *http://help.axeda.com*.

Axeda® Secure Messaging Protocol (ASMP)

The Axeda Secure Messaging Protocol (ASMP) provides a mechanism for secure message exchange between managed assets and Axeda Platform. ASMP is suitable for use as a security layer for the *Axeda® Wireless Protocol (AWP)*, and it provides security services including message authentication, message confidentiality, and replay attack detection.

ASMP is intended primarily for use in communications over wireless networks. It is a compact protocol that minimizes bandwidth usage and does not assume any particular network transport mechanism. The protocol can be used on resource-constrained devices.

Axeda® Wireless Protocol (AWP)

The Axeda Wireless Protocol (AWP) is a messaging protocol used by wireless assets (such as M2M devices) or agents to communicate with Axeda Platform. AWP is a language-, transport-, and operating system-independent application-layer protocol, intended for use where an expressive messaging protocol is needed, while remaining compact and efficient over-the-wire. AWP also aims to provide security provisions, so that messaging between assets and Axeda Platform may authenticate the asset and address other security considerations.

The core of the AWP specification is an Abstract Syntax Notation (ASN.1), which fully describes the data structures and semantics used. For complete details about the Axeda Wireless Protocol and the use of ASN.1, refer to *Axeda® Wireless Protocol Technical Reference Guide*.

Axeda provides a Java-based AWP Toolkit and a C-based AWP Toolkit to enable you to develop applications for your wireless assets to be managed by Axeda Platform.

B

Backup Agent

The Backup Agent feature enables users to use a minimal project and a second instance of Axeda Connector as a backup for another Axeda Connector or for Axeda Gateway (“primary” agent). The backup agent is useful in situations where the primary agent has been deactivated or has become disabled (and therefore is not running). Users with appropriate privileges to the Axeda Service application (and access to the asset) can switch to the backup agent to diagnose and repair a problem with the primary agent. The backup agent is expected to remain transparent until it is needed; when backup agent capability is enabled, users can switch between the primary and backup agents from the Asset dashboard for the related asset (Axeda Service application). For more information, refer to the user guides for these Agents, *Axeda® Gateway User’s Guide* and *Axeda® Connector User’s Guide*.

Binding (Data Item)

The relationship established in a project between data items and data sources (such as OPC or EDD drivers). At runtime, the Data Source Manager of Axeda Gateway or Axeda Connector determines to which data source provider a request from the Tag component should be sent based on the binding.

Block Group (Data Items)

In the context of Axeda Builder, a Block Group is a set of data items defined as one unit (block), to be transmitted as one stream by an Axeda Gateway or Axeda Connector Agent, rather than data item-by-data item.

Bounding Geometry

The bounding geometry of a set of points is the smallest polygon that encloses all of those points. To visualize a bounding geometry, imagine an elastic band stretched open to surround a set of pins that mark locations on a map. The shape of this band represents the bounding geometry of the locations.

C

Codec

In the Axeda Platform environment, a codec is the translator for communications between the Axeda Platform and wireless (and some wired) assets. The codec runs in the Axeda Codec Server (ACS). When the ACS receives messages from assets, it uses the codec to translate ("decode") the messages into a format that the Platform can process. Before messages are sent from the ACS to the assets, the codec is used to translate the messages into a format that the assets can process ("encodes"). For each model of asset that will communicate with the Platform through an ACS, you need to create a *Model Profile* and specify the codec and its parameters.

Communication Timeout (Seconds)

When a response is not received from a remote node, this parameter sets the amount of time (in seconds) before the node is disconnected.

Component

In the context of Axeda Gateway and Axeda Connector, a component is a configurable functional unit that provides a specific feature. For example, the Tag component handles data items received from the data source configured for the agent. When creating a project in Axeda Builder, you choose the components for the agent to load at runtime and for components like the Tag component, you specify configuration details for the component.

Composite Asset

One of the most interesting applications of asset associations is the result of a Many to One association, between the many components of a machine and the machine itself. In this case, the components run Axeda Agents and communicate with the Axeda Platform. Using the Axeda Platform Web services, you can associate the components with a model of the machine, creating a virtual *Composite Asset* in the Platform. The Composite Asset is available in the Axeda Applications just like any other asset. You can view information about it in its Asset dashboard, run reports against it, and execute expression rules against it.

Consecutive / Non-Consecutive Execution (Actions)

Actions can be run consecutively or non-consecutively for an expression rule. If an expression rule is defined with Consecutive actions, the actions are executed each time the expression evaluates to true. If an expression rule is not defined with Consecutive actions, actions are run only the rule first evaluates to true and then they are not executed again until the rule evaluates to false and then to true again.

Important! *For the Reevaluate() action to work, you must enable the consecutive flag.*

Contact (Axeda Configuration)

A person or other entity that should be called (or otherwise notified) regarding an asset. When creating contacts, you define the means by which those individuals can be reached, including physical addresses, phone numbers, fax numbers, and e-mail addresses.

You can associate multiple contacts with each asset. It is intended that contacts represent different relationships with an asset. For example an asset can have three contacts: an “owner”, a “bill-to”, and a “ship-to”. A contact is defined as a “contact type” for an asset. For example, contact Bill Smith may be selected as the “Leasing Agent” contact type for an asset. In addition, a single contact can have multiple relationships with the same asset. For example, one contact can have both a “bill-to” and “ship-to” relationship with an asset.

Counter

A data item that can be used in projects for Axeda Gateway and Axeda Connector Agents to track instances of events, such as data changes or alarms. Actions available for counters are Increment Counter (by 1), Decrement Counter (by 1), Reset Counter (to initial value), and Write Data Item (writes a specified value to the counter). A counter’s initial value (typically 0) is set by the project developer at design time. See also *Data Items*.

Custom Application

See *Extensions (Custom Applications)*.

Custom Object

A custom object provides a shortcut for customizing Axeda Platform rules and actions for your environment. Based on the Groovy scripting language and capable of using the Axeda SDK and Web Services, the custom object feature provides a simple way to implement custom Java code for Axeda Platform. To configure custom objects, use the Axeda Configuration application. For complete details, refer to the online help for that application.

D

Data Accumulator

The Data Accumulator is an Axeda API (`com.axeda.platform.sdk.v1.services.data.DataAccumulatorService`) that provides the ability to write data to a file store (“the accumulator”) in the Axeda Platform, retrieve the data from the accumulator, and delete the accumulator when finished. This service is a no-frills bulk data API. The data written to the accumulator will never be evaluated by the Platform, so it can’t be configured for rule expressions or viewing in the UI or in reports. The data will be written to the accumulator and then read out when accessed via the API.

When data is written to a defined accumulator, the Platform will create the accumulator if it does not already exist and write trip data, appending any existing data that may exist already for the trip. Using this API, your aggregated data, as a whole, can be accessed and FTP'd to an off-site server, reducing the number of trips and enhancing data integrity.

Data Item Format

In the context of Axeda Builder and the Axeda Gateway and Connector Agents, analog data items can have the following *formats*: Unsigned 16, Signed 16, BCD, Float, Signed 32, Unsigned 32. For digital data items, the format is always Digital. For string data items, the format is always String.

Data Item Group (Axeda Builder)

For organizational purposes in Axeda Builder, you can assemble data items for a project into sets, called *data item groups*. For example, your project has multiple analog data items and multiple digital data items for a variety of assets. You could organize the analog data items into groups, by asset type. Similarly, you could organize the digital data items into groups by asset type. You can create either individual data items or data item groups first.

Data Items

A data item is a data point that may be associated with a data source. In the context of Axeda Gateway and Axeda Connector Agents (and Axeda Builder) data items are the configurations of data points that enable theAgent to send and receive data. These Agents can read and store data from external data sources, such as EDD, EEDDSNMPDriver, or OPC, or from other components in the same project.

To send changes in data item values to the Platform from an Axeda Gateway or Axeda Connector Agent, you need to configure a Data Logger for the project in Axeda Builder and set it to trigger on the change in the data item value and then to send the data to the Enterprise Server. In addition, the data items must be selected for viewing (Axeda Configuration application, View and manage models, UI Preferences action).

Note: *The documentation for the Axeda IDM Agents uses the terms Property and Property Reading when referring to data items. A Property is the pairing of an attribute and a value, while a Property Reading refers to the actual value.*

Data item group (Axeda Applications)

A collection of data items, created as a way to manage privileges to data items. Data item groups are assigned to user groups, effectively allowing access to the data items in a group to those users who belong to the assigned user groups. If a user group is not yet associated with any specific data item group, all users in that group can access all data items for the assets to which they have access. Once you assign a data item group to a user group, all users in that user group can see only those data items.

A data item group can contain data items from one or more assets for a single model. One data item can belong to multiple data item groups. Each data item group can be assigned to multiple user groups. Each user group can be associated with multiple data item groups.

Data Logger (Axeda Builder)

A Data Logger is the configuration of one or more files (log files) or of a set of data items to be sent to the Axeda Platform. In Axeda Builder, you can select from multiple ways to send the data to the Axeda Platform. To configure log files or data loggers (also known as *data sets*), use the Logger component of Axeda Builder. Refer to the online help available from the Data Logger dialog box for details.

Data Object

A data object is an object that can be read or written to. Three types of data objects exist in the Axeda Gateway and Axeda Connector Agents: data items (including system data items), expressions, and counters.

Data Source

The Dynamic Link Library (*.dll) or Shared Object (*.so) that retrieves data from an asset and passes it to the Axeda Gateway or Axeda Connector Agent running on the asset. For Axeda Connector, the data sources supported are EDD and OPC For Gateway, the data sources supported are EDD, OPC, and the custom EDDSNMP driver for SNMP-enabled assets. You configure the data source for an agent using Axeda Builder.

Data Synchronization

The ability of an Axeda Gateway or Axeda Connector Agent to receive a block of data items from the device driver and publish all data changes as a block to each subscriber (subscribers are the other components of the Agent, such as the Logger component). The data is synchronized as a single block, from driver acquisition through processing of the data by the subscribing components.

The two components that require synchronization are Expression and Logger. The Expression component requires synchronization when evaluating multiple data items in an expression. The Logger needs to send synchronized blocks of data through the EnterpriseProxy component of the Agent to the Axeda Platform. The Axeda Platform has algorithms that need to be calculated with multiple variables and requires those variables to be delivered on change as a single package.

Deadband

In the context of Axeda Gateway and Axeda Connector Agents and Axeda Builder, the deadband is the percentage by which the value of a data item must change before the Platform will recognize the new value.

Default Model Group

The Default Model Group is one of two default asset groups created during installation of the Axeda Platform and the Axeda Applications. The other default asset group is the Root Asset Group, which is the immediate parent of the Default Model Group. Neither of these asset groups can be deleted. If desired, you can change their names and descriptions from the View and manage asset groups page (Axeda Administration application).

The Default Model Group is the Parent Asset Group for all asset groups created by the Platform when new assets register with the Platform. When the Platform creates the model in the database, it also creates a default asset group for that model. The default model asset groups are also created by the Platform when you manually add a model to the Platform.

> **Note:** *Automatic creation of models must be enabled for the model and model asset group to be created.*

The Root Asset Group and the Default Model Group are unique in the Platformin that they can contain multiple asset groups. No other asset group can contain more than one asset group. *See also Root Asset Group*.

Delegated Admin Units

A "Delegated Admin Unit" (DA Unit) is an object that you can use to represent a client organization within the Axeda Applications. Use Delegated Admin Units when you want each of your client organizations to have their own delegated administrator, their own set of users and user groups, and their own assets and asset groups to monitor. The users, user groups, assets, and asset groups of a DA Unit are visible only to the users associated with the unit and to the Platform administrator of the Axeda Applications.

A Delegated Admin Unit is a complete unit within Axeda Applications that can be managed separately from any other. No user in a DA Unit can access the information and assets of another DA Unit, unless the other DA Unit is a sub-DA Unit. In that case, the Delegated Administrator of the parent unit can access the information and assets of a sub-unit.

It is important to note that within the Delegated Admin Unit, user group security works in the same way as it does if you were using the Platform without DA Units. That is, user groups inherit the privileges of their parent user groups. For example, suppose you have three nested user groups, each associated with a different asset group in the hierarchy. The users who are members of the lowest user group in the hierarchy have access to the assets of all three asset groups, simply because they are members of the user group at each level of the hierarchy. This inheritance behavior is a feature of LDAP directory servers. Delegated Administration is added as a layer on top of the security currently provided by the Platform, giving customers who require it a way to create mini Axeda Applications environments within the same Platform.

Delegated Administration

In the Axeda Applications environment you can extend the security model to include users and assets that are managed by separate organizations. You can do this through Partner Logins and through Delegated Administration. Delegated Administration consists of creating and managing separate, self-contained security units, called *Delegated Admin Units*, and associating them with asset groups. It also requires that you add an administrator user, called the *Delegated Administrator*.

Delegated Administration requires a separate, secondary LDAP directory service that you need to set up before installing the Platform software. For details on configuring that directory service, refer to the *Axeda® Enterprise Server Installation and Maintenance Guide* for your platform.

Delegated Administrator

The Delegated Administrator is responsible for adding sub-Delegated Admin Units, users, and user groups for the unit and for assigning privileges to assets and to pages of the Axeda Applications to user groups. No other Delegated Administrator can see the users, user groups, assets, and asset groups of another DA Unit that is outside of his/her unit. In the hierarchy, a sub-DA Unit has as its parent the DA Unit associated with the Delegated Administrator who created the sub-DA Unit.

Using the Axeda Administration application, the Platform Administrator creates Delegated Admin Units and their Delegated Administrators. In addition, the Platform Administrator creates the primary Parent Asset Group for each Delegated Admin Unit. In the background, the Platform creates a user group, using the Delegated Admin Unit name and associates that user group with the Parent Asset Group. When you assign privileges to the Delegated Admin Unit, those privileges are automatically assigned to the Delegated Admin Unit user group (and vice versa). Within the Delegated Administration structure, the Delegated Administrator can add users, user groups, assets, and asset groups to the Platform. The Delegated Administrator can also create and modify sub-Delegated Admin Units, if required.

The Delegated Administrator has a subset of the privileges of the Platform Administrator. It is from this subset that the Delegated Administrator can assign privileges to the user groups of that Delegated Admin Unit.

Deviation Alarms

In the context of the Axeda Gateway and Axeda Connector Agents, deviation alarms are notifications of a change in the value of a data item such that the value is greater than or less than the allowed range.

Using Axeda Builder, you can specify a range that constitutes a Major Deviation and a range that constitutes a Minor Deviation.

Digital Data Item

A digital data item is a data point whose value is either 0 or 1 (Off or On). The type of data item depends on the data type of the data point being gathered from an asset. Other types of data items include analog (Double) and String.

Display (Axeda Builder)

In the context of Axeda Builder and the Asset dashboard of the Axeda Service application, a display is a graphical user interface for Axeda Connector or Axeda Gateway activities, developed using the Axeda Builder application. You can add one or multiple displays to a

Builder project. In addition, you can specify which display should appear first when the project runs and the display is served at the asset running the project, using the Web Server component of Axeda Connector.

Project developers can add objects to displays, such as buttons, alarm windows, and trend windows, so that technicians can monitor assets by running the project. A project can have one or more displays. Axeda Builder automatically saves displays in JSP format for use with the Axeda Applications and in HTML, JPEG, and DAT formats for use with the Web Visualization applet that accompanies Axeda Connector.

You can upload the JSP displays directly to the Axeda Platform from Axeda Builder. You must be a member of the Administrator group on the Axeda Platform to be able to perform this upload. After the displays have been uploaded and any related tasks performed, technicians can view the displays using the Axeda Service application.

Dynamic Alarms

Alarms generated either by an Event-Driven Driver (EDD) or a project script. To generate dynamic alarms, an EDD driver must be configured as a data source in the agent project, and its driver configuration must be set to *Polling*, rather than *Event driven*. Dynamic alarms are not configured in Axeda Builder, and they are not assigned to project data. However, you can view dynamic alarms in alarm windows (Web visualization only) and alarm loggers, and you can trigger loggers, logic schemas, and sound objects based on dynamic alarms. To perform any of those activities exclusively for dynamic alarms, define an alarm filter for dynamic alarms only.

The EEDDSNMP driver installed with Axeda Gateway does *not* generate dynamic alarms.

Dynamic Group Definition

A Dynamic Group definition is a configuration that ultimately creates a hierarchy of asset groups within a selected *Parent Asset Group*. The hierarchy is based on an ordered set of properties. These properties must be defined for models in the Platform for them to be available for creating or editing dynamic groups.

The order of the properties in the dynamic group definition determines which dynamically created asset group is the parent of another dynamically created asset group. The value for the top-level property in the dynamic group definition will create an asset group that is a parent of the group created by the value of the second-level property, and so on.

Creation of dynamic groups requires that assets are provisioned with values for properties in one of the following ways: through the Platform, through a provisioning file created for the agent running on the asset, or through the Axeda Platform Web services. For details about

provisioning assets for dynamic group definitions, refer to the topic in the online help for the Axeda Applications, called, “Provisioning Assets with Property Values for Dynamic Group Definitions.”

When dynamic group definitions exist and values are assigned to the properties for assets, an asset is assigned to the lowest asset group in the hierarchy created by the dynamic group definition. That said, the visibility of any asset depends on the assignments of asset groups to user groups. Users who are Administrators can see all asset groups and all assets. Non-administrative users who are members of a user group that has been assigned to an asset group that is at the lowest level of the hierarchy can see only the assets assigned to that lowest-level asset group. Non-administrative users who are members of a user group that has been assigned to an asset group in the middle of the hierarchy can see the assets assigned to that middle-level group AND all the assets of asset groups that have that middle-level group as its parent.

Dynamic Timers

Created from the Axeda Service application for scripts, dynamic timers are similar in functionality to the timers configured in Axeda Builder for logic schemas. Dynamic timers can be enabled, disabled, associated with scripts, and disassociated from scripts. The Axeda Gateway and Axeda Connector Agents handle these timers separately from the timers configured in a project.

E

eMessage

eMessage is the Servlet that determines how the Axeda Platform processes HTTP messages received from an Axeda Agent. It is configured as the default Servlet in the Enterprise Server Definition dialog box in Axeda Builder, and it should not be modified.

E-mail Attributes (Axeda Builder and Axeda Agents)

E-mail attributes provide a way to generate content dynamically for an e-mail triggered by an alarm or other type of event. Each attribute follows the syntax convention: `E42:`*`category`*`.`*`attribute`*. For example, the category Event has an attribute called ModelNumber. The Active Content list in the E-mail Styles dialog box shows only the attribute names, not the categories.

E-mail Properties (Axeda Builder)

E-mail properties are settings that are used for all e-mail styles. For example, the Sender address is an e-mail property used for all e-mail styles in the project. Note that if you are using POP3 for authentication, the Sender address must be valid; otherwise, the POP3 server will not send the e-mail message. Axeda Builder tests the value typed in this field for the following format: *Sender_name@xxx.yyy.zzz*. For example, Axeda_Gateway@axeda.com is acceptable. POP3 will accept addresses with one or more dots.

E-mail Styles (Axeda Builder and Axeda Agents)

E-mail styles are templates for e-mail messages that you want to send as a result of data changes or alarms at runtime. In the template, you can enter the recipient's address, a subject, and the body of the message. You can use e-mail attributes to pull additional information into the body of the message. You assign the template a name and the mail server to use. If you want, you can enter a description of the style, which is displayed in the list of E-mail styles in Axeda Builder.

Enterprise Queue

In the context of the Axeda Gateway and Axeda Connector Agents, the Enterprise queue is the collection of XML messages waiting to be sent to the Axeda Platform. The XML messages can include data updates, files, e-mail messages, and alarms. When you select the option for on-demand streaming of data or alarm loggers to the Axeda Platform, the agent checks this queue and, if it is not full, streams the requested data to the Axeda Platform through this queue.

Entity

Used in discussions of network protocols and proxy servers, the term "entity" refers to the data that accompanies an HTTP Request or Response. For example, an HTTP POST method sent with a Request might contain user input to a form.

Error recovery timeout (Seconds)

The error recovery timeout is the number of seconds after an error occurs during which an Axeda Gateway or Axeda Connector Agent attempts to recover before timing out.

Event-Driven Driver (EDD)

In the context of the Axeda Gateway and Axeda Connector agents, the Event-Driven Driver is a data driver designed to use data block callbacks to notify the client of data changes. A custom version of this driver, EEDDSNMPDriver, ships with Axeda Gateway to support SNMP-enabled assets.

Event Subscription Service (ESS)

The Axeda Event Subscription Service allows a remote application or business system to subscribe temporarily to receive asynchronous, low-latency updates regarding the event status of an asset that is managed by the Axeda Platform. This service provides a flexible way for a client application to receive asset information without requiring the configuration of multiple expression rules that trigger PublishObject actions.

The Event Subscription Service can be used by Rich Internet Applications (RIAs). The RIA can use the ESS to create a queue through which the Axeda Platform will send changes to the asset, allowing the application to keep the display current. When a user is displaying or interacting with a particular asset, it is convenient for any changes in the asset to be displayed immediately.

ESS relies on the installation and configuration of an *ActiveMQ* server, as well as other configuration settings, to work with the Axeda Platform. For more information on ESS, see the *Axeda® Platform Event Subscription Service Integration Guide*.

Expression (Axeda Builder)

In the context of Axeda Builder and the Axeda Gateway and Connector Agents, an expression is a data object that is defined by a text string containing data item names, arithmetic, and/or Boolean operators. An event occurs when an expression evaluates to a non-zero value.

Data you can use in expressions include data items, other expressions, timers, and counters. All data items, expressions, timers, and counters in a project have unique names that you choose and that you can use within an expression. An expression is defined by a text string containing names of data items, counters, and other expressions, numeric values (floating point), and operators or functions. An expression does not maintain any state and is not queued. An expression must contain at least one data item name. A data item can be read or written and is a source for events.

Expression Rules (Axeda Applications)

Expression rules provide a way to evaluate events, data, and other conditions and, based on the results, perform operations (referred to as “Actions”). Expression rules can be created using the Axeda Configuration Application or the Axeda Platform Web Services. Expression Rules contain programmatic If-Then-Else definitions. You can use the many predefined functions and actions to create a customized set of expression rules and logic specific to your business requirements. Each rule can be saved and reused or modified and saved as a new expression rule. For more information, refer to the online help for the Axeda Configuration application.

Expressions Component (Axeda Agents)

Axeda Connector and Axeda Gateway have Expressions components that subscribe to incoming data and evaluate expressions when the values of incoming data change from zero to a non-zero value (note that if the value changes from a non-zero value to another non-zero value, the trigger does not occur unless you choose to trigger with every evaluation). Expressions trigger actions based on the results of their evaluation.

You create and modify expressions from the **Expressions** window (and the Expressions Definition dialog box) of the Axeda Builder application. Axeda Builder displays a list of the expressions currently configured for a project. For each expression, the list shows the name you assigned to the expression, the description of the expression, the names of alarm styles associated with the expression, and the actual expression itself.

Extended Tabs (Custom Applications)

When you upload a custom application to the Axeda Platform, you have the option to select the Extended Tab display mode. This allows the custom application to be launched as one of up to five customer-defined tabs in the Axeda Console. You can specify the name of the tab, where it should be displayed in the tab grouping, and any privileges that are required for a user to view the extended tab. See the Axeda Applications *View and Manage Custom Applications* online help topic for more details.

Extended UI Modules

Extended UI Modules can display the content from Axeda Platform custom objects or custom applications. In the Axeda Applications, the term "module" applies to the objects used to display asset data or tools in the Asset dashboard.

Axeda Platform provides a way for you to add new "custom "modules to the Asset dashboard. The content shown in an Extended UI Module is the result of the custom object or custom application defined for that module. This must be a custom object that was created in the Configuration application, Custom Object wizard, or a custom application that was created in the Administration application. See the *Extended UI modules definition* topic of the Axeda Applications online help for more details.

Extensions (Custom Applications)

Extensions in the context of Axeda Applications are standalone applications and web applications that can be hosted by the Axeda Platform as custom applications. Using Axeda Platform developer tools, you can create business applications for your use and that of your customers. Once an application is complete, you can upload it for hosting by the Platform. For example, you might create a web-based data analysis tool that enables customers to compare certain data from their assets to that of another customer. Once you upload the application to

the Platform, you can provide the URL for the application to your customers. Note that the security for your application is entirely up to your application. The Platform does NOT provide any authentication for custom applications.

Custom web applications can be uploaded in a ZIP file that includes SWF files (Adobe Flash/ Adobe Flex) or an archive of static web resource files that can be served from the web server that is used for the Axeda Applications.

External Credentials

The Axeda Platform allows you to store encrypted external account credentials that can then be indirectly accessed via tokens in integration scripts. Passwords do not need to be embedded in scripts, and secure integrations with third party software can be made. See the Axeda Applications online help topics for *External Credentials* for more details.

F

Faults (Axeda IDM Agents)

The faults and alarms of IDM Agents have four components: code, abstract, description, and detail. The Axeda Applications alarms have only one field where the information from these components can be displayed, Description. This field does NOT appear in the current and historical alarms pages of the Axeda Service application. To view this field, click the name of the alarm to display the pop-up window containing all the alarm details (Update Alarm notes). When IDM Agents submit a fault, the four components are combined into the one Description field, as follows:

```
ErrorCode=<fault error code>
Abstract=<fault abstract>
Description=<fault description>
Detail=<fault detail>
```

To use the IDM Agent alarms in an expression rule, use the Alarm namespace and description symbol. Refer to the online help for the Axeda Configuration application for details.

File Upload Manager

The file manager (sometimes referred to as the file upload system) allows files from external systems to be programmatically uploaded into the Axeda Platform for use in content distribution or other script-based processing. The files may be uploaded with meta data including security restrictions, tags, classification, and availability and expiration dates. File manager is an API-only feature, with a full range of file lifecycle features to load, manage and remove files from the Axeda Platform.

Filters (Axeda Policy Server)

In the Policy Server environment, a filter is a set of restrictions for a permission. For every permission, you can apply one or more filters, and a filter can be applied to one or more permissions, in one or more policies. Filters for permissions allow you to:

- Maintain a static list of permissions, each with a default access right.
- Explicitly allow a user access to an action but deny access to everyone else by default.
- Explicitly deny a user access to an action but by default ask for approval for everyone else.
- Create a time window (for example, called "Maintenance Window") to allow or ask for approval when users access the asset during the time specified, and deny at any other time.

- Assign multiple filters to a permission to set up a complex set of allow, ask, deny rules. For example, the filter list for a permission such as Access SSH Remote Session could read:
  - Always allow ‘Acme’ user from 1 PM - 3 PM on Saturdays and Sundays
  - Ask for approval when ‘Partner' user requests an action on an asset
  - Always allow everyone during the time window
  - Deny in every other case

When creating filters, you must assign the filter a name that is unique in the Policy Server and an access right (Always Allow, Ask for Approval, or Never Allow). In addition, if you want to restrict a permission to certain users at certain times, you can add time windows and expressions. Time windows allow you to specify periods of time on a one-shot basis or at recurring intervals.

Expressions can consist of variables, values, and operators:

- For operators, you can use the = sign (equals) and the AND operator.
- For variables, you can specify the userId and the domain name of the Axeda Enterprise Server (enterpriseId) from which the Agent received the action request.

  Values for variables can contain the asterisk (*) wildcard character to represent zero or more characters.

  Note: *Group and other Boolean operators such as OR and NOT are not supported.*

See also Permissions (Policy Server).

G

Gallery (Component Gallery of Axeda Builder)

The Gallery is a library of graphical objects that accompanies Axeda Builder. Project developers can select images from the library or add their own images to the Gallery. The objects can be configured in displays that are served by the Web Server component of the Axeda Connector Agent to reflect data changes at runtime, using dynamic actions.

Gateway

Models defined as “Gateways” contain assets that manage the communications of other non-gateway assets. Assets operating as gateways are configured and installed with the Axeda® Gateway agent.

Gateway assets are physically connected to one or more intelligent assets. These assets must be configured in the same manner as stand-alone assets installed with Axeda® Connector. The Gateway asset manages the communications and data reads and writes between the Platform and the connected assets.

The Axeda Applications distinguish the different asset types within the Model definition. All assets of a Gateway-type model operate as Gateways; all assets of a non-Gateway-type model operate as either standalone assets or are connected to a Gateway-type model.

If you create a model of asset that is managed by a gateway model, you can specify parameters to pass to those assets. These parameters are useful in provisioning the assets for connection to the gateway. When an asset that the gateway needs to manage comes on line, the gateway passes the parameters to that asset. The asset sets these parameters and connects to the gateway for Axeda Platform operations.

> **Note:** *Some Gateways are configured with SNMP and set to automatically discover new assets. You cannot add, modify, or delete the Platform configuration of an asset "found" by a gateway with SNMP.*

Geocoding / Reverse Geocoding

Geocoding is the process of using geographic information, such as street addresses or postal codes, to determine geographic coordinates (latitude and longitude pairs). In addition, the Platform can find misspellings or other mistakes in addresses.

Reverse geocoding is the process of using geographic coordinates (a latitude and longitude pair) to determine a street address and/or postal code. The map engine of the Axeda Platform performs this operation when it receives the location data item from a tracking asset associated with an asset. The location data item is used to display a placemark on the map that points to the location and displays a balloon containing the street, city, state (U.S.) or province (Canada or Mexico), and postal code information.

Geofence

A geofence is a virtual fence that defines a geographical area. You can use a geofence as a means to trigger an event when an asset enters or exits the area. Using the Axeda Platform Web services or the Axeda Configuration application, you can find, specify, or map a location using

the various geofence functions in Expression Rules. If an asset moves inside or outside the designated area, an event can be triggered. For example, the movement might trigger e-mail messages sent to specified users about the movement.

In various applications, the shape of a geofence can be:

- Round (defined using point and radius)
- Rectangular
- Polygon
- Route

You can base a geofence on the latitude and longitude coordinates or on a street address. You can define them on the map in the Asset dashboard or through a form. In addition, the current location can be compared in a geofence.

If a tracking asset can run geofence rules, the Expression Rules engine allows the creation of geofences that can be downloaded to the tracking asset. Depending on the capabilities of the tracking asset, the geofences that can be downloaded and run may be limited to a single shape (typically, a center and radius).

Groovy

Groovy is a dynamic language for the Java Virtual Machine. This language is embedded into the Axeda Platform, to be used for scripting custom logic that calls the Axeda Platform SDK or external cloud services.

Groovy scripts can be run by Expression Rules - either as a function or an action. A script function usually gets some data or condition, or computes something like a deviation. Script actions do something as a result of a condition. Actions typically use the Axeda SDK, Cloud Services, or the Message Queue to integrate with other systems. Writing a Groovy script allows you to retrieve information from the Axeda Platform using a number of pre-defined implicit objects.

Groovy can also implement the body of a Web service method through the use of *Scripto*. Scripto allows you to write code and call that code by name through a REST Web service. This allows a client application to call a set of customized Web services that return exactly the information and format needed by the application.

H

“Heartbeat”

See *Ping rate (“heartbeat”)*.

I

Inheritance (Policy Server)

Within Axeda Policy Server is a hierarchy of asset groups, starting at the top with the Global asset group. The policy of this asset group is inherited by all asset groups that are created automatically when Axeda Gateway or Axeda Connector Agents register with Policy Server. Alternatively, you can create your own asset groups and set up their policies as needed for your organization. Once the Agents register with Policy Server, you can assign your manually created asset groups as the parents of the automatically created asset groups. This assignment makes it possible for the automatically created asset groups to inherit the policies of the manually created asset groups. For example, if you have assets to which access should be highly restricted, you can create an asset group and set its policy up appropriately. You may also have assets that your vendor can service remotely, if access is allowed during certain time periods. You can create an asset group for those assets and create filters for the permissions to allow the vendor access at predetermined time periods. *See also: Asset Group (Axeda Policy Server), File Upload Manager*, *Permissions (Policy Server)*.

Integration Queue

Integration Queue is a *Message Broker Service* provided by the Axeda Platform that sets up a permanent queue between a messaging client and the Axeda message broker. A typical use for Integration Queue is to integrate external enterprise applications with the Axeda Platform. The Axeda Platform provides a named queue to which the enterprise application can connect. Custom Objects or Expression Rules running on the Axeda Platform can then send messages to the enterprise application reliably and asynchronously. For example, a simple Expression Rule can react to a condition by placing a message in the integration queue as follows:

```
IF Alarm.severity > 100 THEN PublishObject()
```

A message is then placed in the queue describing the event, and another application may subscribe to these messages and react accordingly.

IP Address

A unique identifier for a machine on a network, an IP address consists of multiple digits, separated by periods. An example of an IP address would be 172.123.6.120; each set of digits between the periods can contain up to three integers. The sections of the address indicate the location of the machine on the network and depend on the network implementation.

J

Java Secure Channel (JSch)

Java Secure Channel (JSch) is a pure Java implementation of SSH2 that can be used to connect to an sshd server, and use port forwarding, file transfer, etc. The Axeda Platform includes the Java Secure Channel (JSch) library, enabling you to use various secure communications such as Secure FTP (SFTP) or Secure Copy (SCP) to securely transfer files when authoring Axeda Custom Objects on the Axeda Platform.

Job (Axeda Report)

Similar to a batch, a Job is a collection of reports, report views, and/or other jobs created for purposes of scheduling. All the reports, report views, and/or other jobs belonging to a job share the same schedule. When a scheduled job runs, all the items in the job are run and distributed, based on their configurations.

JSP Generator

The Axeda Builder application can save the displays of a project such that they can be used as an application with the Axeda Platform. You can configure the conversion using the Platform Web Pages Settings dialog box, available in Axeda Builder.

L

Linear Scaling

In the context of Axeda Builder and the Axeda Gateway and Axeda Connector Agents, linear scaling describes the output value for a data item, and it is calculated using linear interpolation between the endpoints.

Locale

A locale is a set of characteristics, typically identified with a country, that specifies the language, date and time format, and currency in which to show information. For example, if an Axeda Applications user specifies a locale of English (United Kingdom), when this user logs into the AxedaApplications, the pages show information in English and the pages show the time and date in the format most commonly used in the United Kingdom.

Location (mobile assets)

In the world of mobile assets, a Location is a point. The point consists of the latitude and longitude coordinates of the asset at a given point in time. The Location is similar to a data item in that logic can be applied to it (for example, for creating a geofence). Since logic can be applied to it, the Location is stored separately from data items for an asset.

A fundamental feature of wireless tracking assets is to report the location using absolute reference (for example, GPS coordinates) or using relative data, such as an offset within a building. The tracking assets post the location data to the Platform in the same way as they post other data.

Stationary (or "installed") locations are treated as different types of entities from mobile locations in the Axeda Platform. Assets running Axeda Agents are an example of assets that are typically associated with stationary locations. Assets that have tracking assets installed and the tracking assets themselves are examples of assets that are typically associated with a mobile location. These mobile assets may also have a stationary location.

Notes:
Mobile locations can also be set through the Axeda Platform SDK or Web services. You can find the current mobile location as well as any previous (historical) mobile locations using the Axeda Platform SDK or Web services. Note that the Location Web server supports stationary locations only

The Axeda Platform Web services and SDK express latitude and longitude using the World Geodetic System (WSG 84) coordinate system. This coordinate system is the one used by the Global Positioning System (GPS). For more information, refer to http://earth-info.nga.mil/ GandG/publications/tr8350.2/tr8350_2.html.

Location (stationary assets)

A Location for a stationary asset is comprised of a name and address and is associated with the asset in the database. You can associate multiple locations with a region. Once a location is associated with a region, assets associated with that location are automatically associated with that region, enabling you to search the applications by region to find associated assets. For example, all locations associated with the Massachusetts region are also associated with the Northeast, USA, and North America regions.

You can create and configure locations within the Axeda Configuration application. For assets that are running Axeda Agents, you can use the Axeda Deployment Utility to provide location information to the Axeda Platform.

Locking Permissions (Policy Server)

As part of configuring permissions for a policy, you can select to *lock* the settings at the current level, for the policies of the current asset group. Locking a permission prevents its settings from being overwritten at a higher or lower level of inheritance.

You can lock permissions at the global or model levels. You cannot lock at the asset level as no other policies would inherit from the asset level settings. When locked, the permission cannot be changed for an asset group that inherits that permission. If a permission has been overwritten for a child asset group and subsequently the permission is locked for the parent group, the permission is reset to that of the parent group for the child asset group.

All lower levels that inherit a permission will apply the permission settings as defined in the locked-at level. All higher levels that have the same permission can modify its settings, but the settings will not be inherited by the policy in which the permission is locked. *See also File Upload Manager*, *Permissions (Policy Server)*.

Logger (Axeda Gateway and Axeda Connector Agents)

The Logger component of the Axeda Agents uses the configuration information entered in the Data Logger and Alarm Logger configuration dialog boxes of Axeda Builder. The Logger component collects data values and writes them to one or more log files and/or to the Axeda Platform, depending on how you configure the logger. In addition, it collects alarms and writes them to one or more log files and/or to the Axeda Platform. *See also Alarm Logger* and *Data Logger (Axeda Builder)*.

Logger Component (Axeda Builder)

The Logger component of Axeda Builder allows you to configure the sending of alarms and data to the Axeda Platform. You can select whether to log to one or more files and send files to the Axeda Platform at a scheduled time or to stream the collected data to the Platform on demand. When saved to files, data is stored in HTML or Comma Separated Variable (CSV) format for easy file manipulation. You can select whether to use ASCII CSV or Unicode (UTF-16) CSV.

Logic Schema

In the context of Axeda Builder and the Axeda Gateway and Axeda Connector Agents, a logic schema associates a trigger with one or more sequential actions. At runtime, when the triggers occur, the associated actions run.

Logic Schema Component

In Axeda Builder, the Logic Schema component lets you define and manage the associations between triggers and actions. The Logic Schemas component of Axeda Gateway and Axeda Connector Agents manages triggers and invokes their associated actions. For example, when an alarm occurs, the Logic Schema processes the alarm (trigger) by invoking the action associated with that trigger, such as sending a report of the alarm to the Axeda Platform (action).

M

M2M (Machine to Machine)

With Axeda, M2M is all about monitoring and controlling assets and devices that are remotely deployed. M2M solutions require the status of the environment or device to be made available to a remote server-based application. The Axeda Platform allows you to build and deploy enterprise-grade applications to manage this information for connected products, both wired and wireless.

M2M applications can span many vertical markets and industries. Some examples of M2M applications include:

- **Vehicle Telematics and Fleet Management** - Monitor and track the location, movements, status, and behavior of a vehicle or fleet of vehicles
- **Home Energy Monitoring** - Smart energy sensors and plugs providing homeowners with remote control and cost-saving suggestions
- **Smart Television and Entertainment Delivery** - Integrated set-top box providing in-view interaction with other devices – review your voicemails while watching a movie, chat with your Facebook friends, etc.
- **Family Location Awareness** - Set geofences on teenagers, apply curfews, locate family members in real time, with vehicle speed and condition
- **Supply Chain Optimization** - Combine status at key inspection points with logistics and present distribution managers with an interactive, real-time control panel
- **Telemedicine** - Self-monitoring/testing, telecardiology, teleradiology

Maintenance item

A data item configured for use by the Axeda Preventive Maintenance application to determine if and when an asset is due for maintenance. The application processes maintenance items based on the type you select when configuring them. The three types of maintenance items are defined as follows:

- Incremental – An incremental maintenance item is a delta value; it contains the difference in the value received from the previous value received for the data item. For example, if the first value received is 1 and the second value is 5, the incremental maintenance item contains the value 4. If this type of maintenance item is associated with a counter, a reset of the counter to zero causes the value of the item to be set to zero. For example, suppose a counter represents the number of machine cycles and is reset to zero when the machine is powered off. The counter returns the values as follows:

  1, 2, 3, 4
  Off (reset to 0, -4)
  1, 2, 3
  Off (reset to 0)
  1, 2, 3, 4, 5

  The value of an incremental maintenance item associated with this counter is calculated as follows:

  4 – 4 (the reset to zero) + 3 – 3 + 5, for a total of 5

  If you want to know the total number of machine cycles, use the Counter type instead.

- Cumulative – A cumulative maintenance item contains the actual total of all values received for the data item. The value rises as new positive values are received and falls as new negative values are received. For example, if the first value received is 1, the second 2, the third 3, and the fourth -2, the cumulative maintenance item is 4.

- Counter – A counter contains a cumulative value, without subtracting all values when the counter is reset to 0. For example, a counter might represent the number of machine cycles and, when the machine is powered off, is reset to zero. The Counter returns the values as follows:

  1, 2, 3, 4

  Off (reset to 0)

  1, 2, 3

  Off (0)

  1, 2, 3, 4, 5

  The value of this maintenance item is calculated as follows:

  4 + 0 + 3 + 0 + 5 = 12

The next several terms are used in the online help that discusses Maintenance Items and other features of the Axeda Preventive Maintenance application.

Maintenance interval

The amount of time or the number of counts between maintenance calls.

Maintenance interval count

The number of machine cycles between maintenance events.

Maintenance interval time

The amount of time (in days) between maintenance events.

Maintenance target

The expected calendar date or maintenance counter value for which the next service call is required.

Maintenance target count

The value of the counter (configured for the maintenance item) that will necessitate the next service call.

Maintenance target date

The date that the next service is required.

Maintenance item counter

A data item that is tracking the accumulated machine cycles of an asset. The counter serves a similar purpose to an asset as an odometer in an automobile.

Uninitialized Assets

Assets that have maintenance items and do not have target count/time initialized.

Message Authentication

Message Authentication is the process of verifying message integrity through the use of the Message Digest. The sender generates and sends the Message Digest along with a message. The Message receiver recalculates the Message Digest and compares the results with the received digest.

Message Broker Service

The Axeda Platform includes a message broker service and supporting message publishing features, which enable integration with enterprise applications as well as near-real-time data streams for selected assets. Message queues are ideal for reliably sending messages from a message producer to a message consumer. This communication is asynchronous because the message broker queues messages from the producer and sends them to the consumer on request.

The Axeda platform provides two different message broker services:

Integration Queue -- sets up a permanent queue between a messaging client and the Axeda message broker.

Event Subscription Service (ESS) -- allows a client to dynamically create a queue and be notified of specific events.

Message Digest

The Message Digest is a fixed-size result, obtained by applying a mathematical function (the hashing algorithm) to an arbitrary amount of data.

Model

A model is a set of assets that have the same properties and are considered to be of the same type. For example, ModelXYZ may contain 50 identical connected, intelligent assets, all installed at different locations but performing the same tasks. Numerous assets of different model types can be connected to a single Gateway asset.

All assets of a Model share the same set of data items. You are limited to viewing only models for which you have privileges.

Model Profile

A Model Profile is a specification for wireless assets that defines the communications command set or protocol that the assets use, including the message format and delivery method for sending and receiving messages. Model profiles are stored on the Axeda Platform, which uses them to determine how to encode messages to send to assets and to decode messages received from assets of the model.

In addition, a model profile identifies the threshold rules supported for execution at the asset. When it contacts the Platform, an asset of that model will receive only the threshold rules it can run, based on the definitions in the profile.

Monitor (asset states)

A monitor consists of two or more asset state values plus the Unknown state. These state values can be converted into virtual (calculated) states using asset state groups. An asset state group is a set of two or more asset states that are members of the same monitor.

Monitors and asset states are created in projects using Axeda® Builder. Asset state groups are created using the Axeda Configuration application so that the information can be used by other applications. For example, the information can be used to calculate Key Performance Indicators (KPIs).

For complete information, refer to the topic, “About Asset State Monitoring,” in the online help for the Axeda applications.

N

Named User (Axeda Report)

A Named User for the Axeda Report application is an Axeda Applications user with an account defined in the LDAP directory service for the Platform. This user can create and edit reports or Ad Hoc query reports, or perform Axeda Report Web administration (including Content Administration and updating the Axeda Report pack). Only a limited number of users can be defined as Named Users for Report Studio, Query Studio, or Report Web Administration. The number of named users is specified by your license. There are separate named users for Report Studio (reports), for Query Studio (Ad Hoc reports), and for Report Web Administration (Report Web Admin). You use the Axeda Administration application to set up named users.

Note: *You must have the appropriate license for Axeda Service Intelligence for this feature to be available.*

Networked Assets (Axeda Builder)

The Networked Assets component of Axeda Builder lists the assets to which Builder can download and upload data and files. For Connector projects, the Networked Assets lists includes all assets running Axeda Connector to which files will be downloaded, and it includes all Axeda Platforms to which data and displays will be uploaded. One of the configured assets is flagged as the project's *active asset*, indicating that any Active Asset commands apply to that asset. For Gateway projects, the Networked Assets list includes the project's Development Target asset (the *active asset* for a Gateway project), and it includes all Axeda Platforms to which data and displays will be uploaded. The Remote Server utility must be running on the assets for you to be able to download files to those assets.

Notification (Axeda Platform)

A notification contains information about an asset and is sent to specified recipients when a triggering condition occurs. Recipients can include individual users, groups of users, or other valid e-mail addresses. You can configure notification actions that are triggered by a rule evaluating to true or that users can run manually. Users can run actions from the Actions module on the asset dashboard of the Axeda Service application; they can also configure them to run as part of a package.

When sending notifications, the Axeda Platform can use various methods, such as e-mail, pager, or Web alerts. By default, the Platform contains support for e-mail notifications only. To create custom notification styles for your Platform, go to *http://help.axeda.com* for more information.

A notification definition includes a name for the notification in the Platform, the name of sender, and one or more entries. An entry is a combination of locale, title, and body for the message. For example, you could create different entries for different locales or different messages. You can include template placeholders and timestamp placeholders in the body of e-mail notifications.

Notification (Policy Server)

For every asset group in Policy Server, whether automatically or manually created, you can configure an e-mail message to be sent when Policy Server receives a request for an action from the Axeda Agent running on an asset. You can specify the e-mail address of a Policy Server user or a different user. You can also select a Policy Server role as the recipient. The notification settings also include a Sender address, the Subject of the message, and a message body. *See also Asset Group (Axeda Policy Server)*.

O

OPC Data Source

The OLE for Process Control (OPC) is an industry software standard, designed to provide business applications with easy access to industrial plant floor data. Only the Axeda Connector Agent supports OPC data sources.

Organization

A company or other entity considered to be an “owner” for the particular asset. Organizations have associated *contacts* (not to be confused with *Partner contacts*) and *Location (stationary assets)*. You create organizations in the Platform and define locations for those organizations, and then assign the organizations to assets and assign one or more contacts to those organizations. One organization can have multiple associated locations.

P

Package (Software Management Application)

In the Axeda Software Management application, you can create sets of instructions to send to the Axeda Agents for execution. These sets of instructions are called *packages*. Packages provide the following instructions for agents to execute: Upload, Download, Execute, Set Data Item, Custom (your own SOAP call for a custom component), and Restart. A Wait instruction allows you to set a time period during which certain conditions must be met before the instructions are executed. These packages provide a means for service technicians to upload log files, for project developers to download revised project files or upgrades to the Axeda Agents, and for administrators to run scripts to perform maintenance on assets.

Parent Asset Group

Every *Asset Group (Axeda Configuration)* in the Platform is an immediate child group of one, and only one, Parent Asset Group. When creating asset groups manually, you are prompted to select a Parent Asset Group for the new group. The Platform creates a name for the asset group that is a combination of:

Root Asset Group + <name of Parent Asset Group> + <name of new Asset Group>.

It is possible for additional asset groups to exist in the hierarchy between the Root Asset Group and the selected Parent Asset Group. For example, suppose you create an asset group called SP1 and select the Root Asset Group as its Parent. The fully qualified path name of SP1 in the Platform would be `/Root Asset Group/SP1.`

Next, suppose you create an asset group called Laptops and select SP1 as its Parent Asset Group. The fully qualified path name of this new asset group would be `/Root Asset Group/SP1/Laptops.`

When the Platform creates asset groups dynamically, any asset group created is created with only one parent asset group. The asset group name also becomes a fully qualified path. For example, suppose you created a hierarchy under SP1 using the values of the properties, Product and Operating System. If an asset registered with the property values "Desktops" for Product and "Windows" for Operating System, the fully qualified path name of the asset group would be `/Root Asset Group/SP1/Desktops/Windows`.

> **Note:** *The default model asset groups that are created when an asset registers with the Platform for the first time are created in a special asset group that is an immediate child of the Root Asset Group. This group is called Default Model Group.*

Partner Login (formerly “Partner Access”)

In the Axeda Platform environment, “Partners” are typically organizations, groups, and customers with an interest in one or more deployed assets. For example, a partner may be the support organization for your asset. Occasionally, this partner would need to be able to log in to the Axeda Applications to manage, troubleshoot, or repair your assets remotely.

Axeda Applications users are typically defined in user groups that have been created in the LDAP directory service of the Platform; they have their own login accounts and privileges granted through user group settings. However, partners are defined in a *secondary* directory service and granted limited access to Axeda Applications. Each partner has an associated user group, which is automatically created by the Platform in the directory service defined for partner login accounts.

For example, through Axeda Case Management you can create a case and assign it to a partner. The Platform sends the case information to the specified partner. Part of the information sent includes a login code that provides the partner’s contacts with privileges and access to the Axeda Applications. Partner contacts cannot log in to the Applications until they receive the session code and Axeda Applications URL for the asset to be managed. At login, the partner contact provides the login code created for the login session. Partner contacts have a limited amount of time to use the Axeda Applications and are restricted in the actual tools they can use.

During case assignment, only partners associated with user groups defined in the partner directory service are available for assignment. A contact of that partner can log in to the Applications to use the tools and information.

Partner contacts

Partner contacts are the people associated with the Partner defined in the secondary directory service. Once they have received a session code and Axeda Applications URL for an asset, partner contacts can log in to the Axeda Applications and access information about the asset and the case for which they received a notification.

Partner session

A partner session refers to the time that a partner contact spends logged in to the Axeda Applications through a provided one-time login. During a session, a partner contact may view logs of uploaded files and audit records, run actions on the asset such as a script or software download, and create a remote access session to the asset.

The Axeda Platform creates a session code when you select to create a new partner session. The session code specifies the application displayed when the partner contact logs in, the information for the asset in question, and the partner information for purposes of tracking and auditing partner activity in the Axeda Applications.

Permissions (Policy Server)

In the Policy Server environment, the policy for each asset group contains a set of permissions defined for the supported actions. Each permission is a combination of at least a name (identifier) and an access right. Optionally, permissions can also have one or more assigned filters. The permission defines how the Axeda Agent should handle a request for an action. For example, one of the supported actions is Upload File. Its default permission is for all files (*.*) and the access right is Ask for Approval. With this default permission, the Agent will send a message to Policy Server to request approval and will wait for that approval every time that a user at the Platform requests to upload a file from that asset. Depending on how often upload requests are made, this setting could degrade performance. If you have certain files (for example, log files) that you want the Agent to upload whenever a user at the Platform requests the upload, you can create a permission that specifies these files (xGate*.log, for example) and that has the access right, Always Allow. Refer to the online help for Policy Server for more information. *See also File Upload Manager, Asset Group (Axeda Policy Server)*, *Policy (Policy Server)*.

Ping rate (“heartbeat”)

In the Axeda Platform, a ping rate or “heartbeat” is the interval (in seconds) that an Axeda Gateway or Axeda Connector Agent waits before checking its connection to the Axeda Platform to ensure that it is still active. This interval is the number of seconds that the Axeda Agent waits *between* connection checks. The default ping rate for Axeda Gateway and Axeda Connector Agents is 180 seconds (3 minutes). The rate selected depends on the particular site and can be adjusted from the Platform after it has been running.

Some sites use 30 seconds while others prefer 5 minutes. The ping rate affects how long users must wait before they can connect to an asset through a Remote Session as well as how long they must wait for results after requesting an action or script be run on an asset. If connections from an asset to the Platform will use dial-up, then a 5-minute ping rate is preferable to a 30-second ping rate.

The documentation for the Axeda IDM Agents refer to the ping rate as the “poll rate.” The meaning is exactly the same.

Policy (Policy Server)

A policy is a set of actions, each with one or more permissions and their associated filters. Each asset group in Policy Server has one - and only one - policy. The policy defines how the Axeda Gateway or Axeda Connector Agent handles requests for the actions from the Platform. When an Agent first connects, Policy Server creates at least two asset groups, one for the model and one for the serial number of the asset where the Agent is running. For Axeda Gateway, Policy Server also creates Model and Asset groups for the assets that Gateway is managing. These automatically created asset groups inherit the Global policy, and it is that policy that is first sent to the Agents on first contact. If you move the asset groups or edit their policies, Policy Server sends the revisions to the Agent on the next contact. *See also File Upload Manager*, *Asset Group (Axeda Policy Server)*, *Permissions (Policy Server)*.

Polling (Axeda IDM Agents)

After registering with the Platform, Axeda IDM Agents can contact the Platform periodically through polling messages. Polling messages are similar to the ping messages sent to the Platform by the Axeda Gateway and Connector Agents. Once an agent contacts the Platform, the Platform checks for any messages for the asset where the agent is running. If any messages are waiting to be sent to the asset, the Platform sends them to the agent.

Preview tool

This Axeda Builder tool simulates runtime without security while still allowing configuration changes. Using this tool for an Axeda Connector project, you can view and interact with the project just as a project operator will at runtime. To start and stop Preview, use the commands on the **Active Asset** menu, or right-click the name of the project in the Project Window and select the commands on the context menu. You cannot use this tool for Gateway projects.

Privileges (Policy Server)

In the context of Axeda® Policy Server, a privilege defines activity that a user is allowed to perform in a given component of the application. For example, the Add/Edit privilege for the Pending Requests component enables a user to approve or deny pending requests. The Add/Edit privilege for the Configuration component allows a user to add, modify, and delete asset groups to change the settings for the Audit Log, and to delete missing assets. Privileges exist in the Platform and cannot be created or destroyed. You select them when defining profiles. Once you have created profiles, you can combine them into roles and assign roles to each user and users to each role. Users derive their privileges to the components of the Policy Server application from the roles assigned to them.

> **Note:** *Although the terms "permission" and "privilege" are often used interchangeably, that is not true for Policy Server, either in the user interface or the documentation. As defined in this topic, "permission" always describes how an asset action is managed within a policy. Think of permissions as asset related. On the other hand, "privilege" always refers to actions that users logging in to the Policy Server application are allowed to perform. Think of privileges as user related.*

Privileges (Axeda Applications)

In the Axeda Platform environment, privileges define the operations that users can perform in the pages of the AxedaApplications. In addition, privileges define the operations that can be performed through the Axeda Web Services. Privileges control viewing, modifying, adding, and deleting items stored in the Platform; viewing and using the tools in the Axeda Applications' pages and viewing specific data in those pages; and viewing entire applications. For complete information about all the user group privileges and their functionality within Axeda Applications, see the topic, "User Group Privileges in Axeda® Connected Product Management Applications," in the online help.

Privileges are assigned to user groups, in the Add and remove application privileges for user group page. The set of privileges is defined in the Axeda Platform and cannot be modified from within Axeda Applications.

Process

A process is the series of steps that a company takes to generate a product. For example, a manufacturing process for semiconductor fabrication equipment.

Process Value

In Axeda Builder, a Process Value refers to the real-time value of a data item, counter, or expression.

Profiles (Policy Server)

In the Axeda Policy Server application, a profile is a set of privileges to one or more of the application's main components.

Except for Audit Log and Remote (Sessions), each application component has two privileges - View and Add/Edit. If you have the Add/Edit privilege to one of the Policy Server components, you also have the privilege to Remove. For example, if you have the Add/Edit privilege for the Users component of the application, then you can add, edit, and delete profiles, roles, and users. For Audit Log, the only activity is viewing, so the only privilege is View. For Remote (Sessions), you can view and end remote sessions, so the privileges are View and End; no adding or editing is involved.

You may want to create several profiles that provide different sets of privileges. You can then create roles, which are sets of profiles, and assign varying combinations of the profiles to the roles. When creating a role, you can also assign existing users to the role. Finally, when you create user accounts, you can assign roles to the users. *See also Role (Policy Server)*.

Project (Axeda Builder)

In Axeda Builder, the project is an interface that consists of multiple configuration files for the Axeda Gateway or Axeda Connector Agent. A project may include one or more graphical displays, in which data from a process is reported. How that data is reported and what happens when value changes indicate problems is up to the project developer to configure as part of the project or to handle through the Axeda Applications or the Axeda Platform Web services.

The result of saving a project is a set of XML configuration files that are downloaded to the Agents that are running on the assets that the project monitors. Depending on the needs of the organization, the project can gather data, upload it to the Axeda Platform, display the data to operators either at the device or across the Internet, generate alarms and other events, and send values for data items to the assets.

Project Log (Axeda Builder)

Visible in Axeda Builder for Connector projects only, the Project Log is a log file for a project. This log file maintains a list of all design-time events for the project. The log records the type of event and a description, the source of the event, and the date and time of the event.

Project Log (Agents)

Configurable through Axeda Builder for Axeda Gateway and Axeda Connector Agents, a project log is one or more files that maintain a record of all messages generated by the Agent at runtime, including error messages. Using Axeda Builder, you can enable or disable the log and set a size limit for the files in which the information is stored. You can also have the messages sent to the console of the asset at runtime.

Property (Axeda Platform)

For models in the Axeda Applications, a property is a piece of user-specific or use-specific information for all assets of a model. When configuring models, you can configure any information that you want to maintain for all assets of a model.

For the Axeda Platform, a property is a configurable parameter that defines an aspect of system operation. For example, the name of the LDAP server that provides authentication for the system. The properties of the Platform and their current settings are visible through the Axeda Administration application (System Configuration page); you cannot change the values in the application. Contact your Platform administrator if any changes are needed.

Property Reading (Axeda IDM Agents)

In the context of the Axeda IDM Agents, a property reading is a data item. When contacting the Platform, an Axeda IDM Agent can send property readings and alarms (faults), which are displayed in the Asset dashboard for the asset where the IDM Agent is running. This functionality is the same as for the Axeda Gateway and Connector Agents. From the Asset dashboard, you can view current and historical data item values. You can also view current alarms and acknowledge them. Once alarms are acknowledged, they are moved to the historical alarms page.

Provisioning (Axeda Gateway)

In the context of the Axeda Platform, *provisioning* refers to supplying network connectivity information to gateway assets so that they can connect to assets that you want them to monitor. Using the Axeda Configuration application, you can add a gateway asset and the models of

assets that it will monitor to the Platform. You can also construct a provisioning instruction for the gateway asset. This instruction contains the network connectivity information for the assets that the gateway asset will monitor.

Proxy Server

A proxy server is an application server that acts as both a server and a client for the purpose of making requests on behalf of other clients. The proxy server may service requests internally or, after performing some processing on the requests, pass them on to other servers.

If your network is set up such that the Axeda Gateway and Axeda Connector Agents must go through a proxy server to communicate with the Axeda Platform, you can either configure the proxy server using the Axeda Deployment Utility or allow the Agent to detect the appropriate HTTP connection method and flags to use, based on the current network topology. When you choose automatic detection, the Agent saves the connection method and flags in an XML-based configuration file for use on subsequent restarts. If your network topology changes such that the Agent should perform the automatic detection again, you can either delete the configuration file (*AutoProxy.xml*) or use the Axeda Deployment Utility.

Publish (Scripts)

Publishing a script from the Axeda Applications makes that script available for registration and use on other assets of the same model, or for configuration as Run Script asset actions. Note that you cannot un-publish a script except to delete it from the assets. In addition, you can delete only scripts downloaded from the server, not scripts installed directly on the asset.

Publish (Reports)

Publishing a report or report view makes that report or report view that was created in a Private folder available in the Public folder, where users with the appropriate privileges to Axeda Report can access it. What the users can do with a published report depends on their privileges to the Report application and its various tools.

Q

Query Studio

Query Studio is the tool you use to create Ad Hoc query reports. This tool is available only if your organization has purchased the appropriate license for Axeda Service Intelligence and if you are set up as a Named User for Axeda Report and the Query Studio tool. *See also Ad Hoc Report (Axeda Report)*, *Axeda® Report and Axeda® Dashboard (Axeda® Service Intelligence)*, or *Report Studio*.

Queue

The Axeda Gateway and Axeda Connector Agents collect XML messages in a queue for transmission to the Axeda Platform. These messages can contain data updates, e-mail messages, and alarms. When the queue reaches the Flush Size or the Flush Timeout period expires, the messages are forwarded to the Axeda Platform. The maximum Queue Size comes into play when a connection to the Axeda Platform is not available.

Queue Size

The Queue Size is the amount (in KB) of data that can accumulate in the queue while the Axeda Platform is not connected before the Axeda Gateway or Axeda Connector Agent starts discarding the oldest data.

R

Recipe

In the context of Axeda Builder and Axeda Connector, a recipe is a list of data items and values that can be sent to assets in response to a trigger. The RunRecipe action sends a recipe to an asset when an associated trigger occurs. You might use a recipe to reset a temperature on an asset, for example, when the reported temperature exceeds a certain value. Recipes are configured in Axeda Builder.

Recipes Component

In Axeda Builder, you can configure recipes using the Recipe component in the Project Window. You can set up lists of data items and values in one or more recipes. You may, for example, want to use one recipe for a particular group of data items, and another recipe for a different group of data items.

Reevaluate Action (Expression Rules)

The Reevaluate() action evaluates the current message for the expression rule again after a specified amount of time (for example, after two days), or on a specific date and time. When the Reevaluate action runs, all the data in Alarm-related messages, Data messages, and Location messages is refreshed to get the latest values from the Axeda Platform. For an Alarm message, this action updates the Alarm state, severity, and extended data. For a Data message, the most recent data item value is retrieved, and for a Location message, the most recent location is retrieved. For any other type of Rule, the message data is *not* refreshed.

This action takes one parameter, `timeSpan` OR `timeInSeconds`. The `timeInSeconds` is returned from the `Date()` function. The `timeSpan` is in intervals of seconds, minutes, hours, days, weeks, months, or years.

Important! *For the Reevaluate() action to run, you must enable the Consecutive flag for the expression rule. (Refer to Consecutive / Non-Consecutive Execution (Actions).)*

If the trigger is InactiveAlarm, the Reevaluate() action re-runs the expression rule associated with the Inactivetrigger message if the alarm was first active and subsequently went inactive. If it started as inactive, then Reevaluate does not run the expression rule. The Reevaluate() action will refresh alarm data if the alarm was initially active and then went inactive.

For details about this action, refer to the online help for the Axeda Applications.

Region

In the context of Axeda Platform, a region is a collection of locations defined and maintained in the Platform. Depending upon the needs and structure of your company, you can create regions that correspond to actual regional organizations within your organization (for example, Northeast, Southeast, Northwest, Southwest, and so forth) and then group the applicable locations within the related regions. Alternatively, you can create regions that are used simply as a way to group locations.

One region can contain multiple levels of nested regions. For example, the North America region could contain the USA region, which contains the Northeast region, which contains the Massachusetts region. In this case, the Axeda Service application (navigation path) would show the resulting region list as North America: USA: Northeast: Massachusetts.

If you associate a location with a region, all assets associated with that location are now associated with that region. This automatic association enables you to search by region to find associated assets. In the preceding example, all locations associated with the Massachusetts region are also associated with the Northeast, USA, and North America regions.

You can create and configure regions using the Axeda Configuration application. Also, you can use the Axeda Deployment Utility to provide location information for assets to the Axeda Platform.

Registration (Axeda Agents, Scripts)

The first message sent from an Axeda Agent to the Axeda Platform. The registration message from an Axeda Gateway or Axeda Connector Agent contains information about the asset (model number, serial number) and the scripts that are available on the asset. This information can be stored on the Platform. For scripts, registering is the process of downloading all files for a script (sequence and asset scripts) from the Axeda Platform to the asset. An Axeda Gateway or Axeda Connector Agent cannot run a script until it is installed directly on the asset or is "registered" with the asset.

Like the Axeda Gateway and Connector Agents, the Axeda IDM Agents can register with the Axeda Platform, and the Platform can automatically create the models and assets in the Platform for the assets where the IDM Agents are running.

Remote Session

A remote session is a connection between a remote, deployed asset and a user of the Axeda Applications. Remote Sessions are established through the Axeda Platform. The remote assets must be configured to support Remote Sessions. In contrast with other desktop-sharing connections, remote access sessions can support machines that do not have monitors. Axeda Connector and Axeda Gateway Agents and the Axeda Platform support the following types of remote sessions: Remote Terminal, Remote Application, Remote Browser, and Remote Desktop. For details, refer to the online help for the Axeda Applications, the online help for the Remote Sessions tool in Axeda Builder, and the *Axeda Builder User's Guide*. For information about remote sessions with assets running Axeda IDM Agents, refer to *TotalAccess Server*.

Report

A report is an object created using Report Studio. Report Studio provides a robust development environment for creating custom reports, including a wide variety of formatting and calculation options. You must be a Named User to be able to create new and edit existing Reports. All of the standard reports provided with the base system have been created using Report Studio. For more information refer to the online help available within the Report Studio application.

Note: *You must have the appropriate license for Reports and Report Studio to be able to create reports through Axeda Report.*

Report Studio

Report Studio is the tool you use to create Reports. This tool is available only if your organization has purchased the appropriate license for Axeda Service Intelligence.

Report View (Axeda Report)

A report view is a copy of a report that is linked to the original report, such that any edits made to the original report become available in this copy of the report. Report views enable you to customize an existing report from the Public folder and store it in your Private folder. You can then set up the report view so that it always uses the same values for the prompts (for example). Your changes do not change the original report.

If the original report is moved, the Platform maintains the link to it from the report view. However, if the original report is deleted, the *link* to the original report is removed. Your report view is *not* removed.

REST

REST stands for “Representational State Transfer”. In general, REST defines access to a service using HTTP.

In the context of *ActiveMQ*, REST means sending and receiving messages using HTTP POST and GET.

Role (Axeda Applications)

In the context of the Axeda Administration application, a role is a list of users defined in the directory service that authenticates users for the Axeda Applications. You can use roles as a way to group users for notifications. Roles act as distribution lists for activities of the Axeda Platform, such as e-mail notifications. Role assignments are maintained in the Platform and have no application outside of the Platform. Refer to the online help for the Axeda Administration application for more information about roles.

Role (Policy Server)

For the Axeda Policy Server application, a role is a set of profiles, which combine to provide users with privileges to each main component of the application. You can assign one or more profiles to a role. You can then assign one or more users to a role, and conversely, one or more roles to a user. The activities a user can perform after logging in are limited by the privileges defined in the profiles (which have been assigned to the roles) for the user. *See also Profiles (Policy Server)*.

Root Asset Group

The Root Asset Group is the global set that contains all other asset groups created in the Platform. Platform administrators can change the name of this group by selecting the name of the group in the View and manage asset groups page. The View or edit properties for Root Asset Group page appears, where you can change the name of the Root Asset Group and add a description. However, the Root Asset Group cannot be deleted.

Rule Timers

A Rule Timer defines a periodic timer and provides a way to apply the timer to *Expression Rules (Axeda Applications)*, thereby enabling you to configure Expression Rules that trigger based on periodic timers. When the timer period is reached, the Axeda Platform retrieves the Expression Rules defined for the timer event and runs them, evaluating their conditions (“If” expressions).

Each Rule Timer includes a schedule and a set of expression rules or state machines. The set of rules can include any number of expression rules or state machines. For example, RuleTimerXYZ can be scheduled to run two Expression Rules and one State Machine.

When you define a rule for a Rule Timer, you can specify whether you want that expression rule to run for all assets or not.

S

Script file

A script file is created outside the Platform (for example, in a text editor or other language editor). You add script files to the Axeda Platform to deploy to an asset or you can include them when installing the Axeda Gateway or Axeda Connector Agent on the asset. These scripts are called and run by commands defined in the *sequence script file*. These scripts can be written in a language supported by the assets on which they will run. *See also Sequence Script File* and the *Axeda® Platform Script Developer's Reference Guide*.

Scripto

When building a connected application, the Axeda Platform allows developers to extend the functionality of their M2M applications by using a Groovy Custom Object, and Scripto allows you to use this extra functionality and include the output where needed.

Scripto is a programming language that provides adaptable Domain Specific Services for use with building connected applications. In the context of a Web Service, the Domain Specific Language or Domain Specific Services act as an intermediary between the application and the web service. With the Axeda Platform, an application can be created as the entry point for the data, and a Custom Object can be created as the end point for the data. This gives developers complete control over how the data is processed from start to finish. Scripto, as the broker, allows the data to pass between the two points via a simple POST request, and Scripto does not require special attention as the intermediary (whereas a typical Web Service fails unless the data conforms to the schema of its domain specific language).

Security Component

In Axeda Builder, a list of users and groups configured for the runtime operation of the project. You can add groups, users to groups, and assign the Operate privilege to groups for each of the displays in a project. This operator-level security is supported by Axeda Connector only.

Sequence Script File

A script file that contains commands or programs to run in sequential order (similar to UNIX shell scripts or DOS batch files). A sequence script is a text-based file in which each line is a separate command to the Axeda Gateway or Axeda Connector Agent. You can create a sequence script that runs external programs on the asset as well as internal Agent actions. The content and format of the sequence file control the interaction between the external programs and Agent actions. For example, the sequence script can extract the model and serial numbers of the asset and then run a script that extracts specific asset registry settings. The Agent can then

be made to save this information to a snapshot XML file and parse that file before uploading it to the Platform for access by Axeda Applications users. To make a sequence file available for deployment to other assets or for configuration in a Run Script action, you need to publish the script. *See also Script file*, *Publish (Scripts)*.

Settings Tools (Axeda Builder)

This set of tools enables you to configure the runtime operation of the project. Among several other runtime settings, these tools enable you to configure the communication between an Axeda Gateway or Axeda Connector Agent and the Axeda Platform and from Axeda Connector to remote operators (using the Web visualization applets). Refer to the online help for Axeda Builder for details on using these tools.

Shoulder Tap Messaging

Used by wireless assets, a Shoulder Tap message is basically a message sent from the Platform to an asset to let it know it has pending messages.

When an outgoing message has been queued for the asset, the message is delivered to the asset in the poll response. Since the percentage of time that an outgoing message is likely to be queued when it polls is quite small, the overhead of polling is very high.

The alternative to polling is a feature called Shoulder Tap whereby the Platform notifies the asset that a message has been queued for the asset. The asset can then intelligently poll only when necessary.

For more information on Shoulder Tap messaging, refer to the Axeda Applications Online Help, or the Axeda Codec Server Developer's Guide.

Snapshot

A snapshot is an XML document sent by an Axeda Gateway or Axeda Connector Agent to the Axeda Platform as a result of a script. Users of the Axeda Applications can view snapshots in the Axeda Snapshot Viewer, which is a browser-based explorer tool for searching, filtering, and navigating the contents of a snapshot. The Snapshot Viewer can display XML files as long as they are created correctly, that is in accordance with the *snapshot.dtd* (the XML Document Type Definition file for snapshots*)*. For more information and an example of a snapshot.xml file, refer to the online help for the Axeda Applications.

SOAP (Simple Object Access Protocol)

In the context of Axeda, SOAP is the protocol used to enable the Enterprise Server and other applications to communicate with Axeda Agents (Axeda Gateway, Axeda Connector, or Axeda IDM Agents). For more information about using SOAP with Axeda Agents, refer to the Axeda Gateway User's Guide, Axeda Connector User's Guide, and the Axeda IDM Developer's Reference.

State Machines

State machines identify the ability to track and evaluate a "state", and determine when an object transitions from one state to another. For example, you can define a state location as "At the port" or "At the warehouse". A transition may be when the related object moves from the "At the port" state to the "At the warehouse" location.

String Data Item

A data item whose data type is String.

Symbols (in expression rules)

Symbols are used when creating expression rules Symbols are attributes of a related namespace and are used when evaluating the If statement or running the Then action for that rule.

An example for the Registration namespace is the `first` symbol. For example, an expression rule defined to run for a Registration message can be configured to determine if the registration message contains a first time registration for the related device:

```
If Registration.first
```

Synchronization

See *Data Synchronization*.

System (Axeda Configuration)

In the context of the Axeda Configuration application, a system is a collection of heterogeneous assets maintained in the Axeda Platform. You can organize assets into systems to aid in navigating to assets within the Axeda Applications or to help you track and view assets.

You can create and configure systems using the Axeda Configuration application. Also, you can use the Axeda Deployment Utility to send system association information for assets to the Platform.

As opposed to an asset Group, the assets associated with a system are not bound by model. If you want to configure rules for groups of assets, configure those assets within an asset group and set the rule on that group.

T

Threshold Rules

A Threshold Rule provides a way for your assets or the Platform to evaluate conditions and run actions for conditions that evaluate to true. As opposed to Expression rules, Threshold rules support a discrete set of conditions and actions. The resulting threshold rules are of a fixed format, enabling manufacturers to configure their assets to support running this type of rule. The threshold rules are saved to the Platform and can be sent to assets for running, if supported by their defined asset profiles.

Each threshold rule defines a pairing of a threshold condition and a threshold action. For example, for a threshold rule that creates an alarm, AlarmX with severity 250, when the value of a data item called “temp” is above 100, the threshold action is “createAlarm” and the threshold condition is “AboveLevel”.

If assets do not support running and evaluating threshold rules, rules for those assets can be run at the Platform, as with *Expression Rules (Axeda Applications)* and *State Machines*.

> **Note:** *Currently, threshold rules can be created, edited, deleted, enabled, and disabled only through Axeda Platform Web services. Refer to the Axeda® Platform Web Services Developer’s Reference Guide for detailed information.*

Timer

In the context of Axeda Platform, a timer is a Platform event that can cause an operation to run. Using the Rule Timers pages in Axeda Configuration, you can configure timers that will trigger expression rules and associate the timers with assets or models. *See also Rule Timers*.

Using the Timer Definition dialog box in Axeda Builder, you can specify whether the timer executes based on a defined time and date (Absolute), such as a yearly timer that executes on January 1 at 01:00:00 of each year, or in relation to a defined time and date (Relative), such as 30 seconds after the project starts in runtime (Start Point - Immediately). Timers can occur on a regular basis (Periodic) or just once (One Shot) and then have to be reset. You can also use logic schemas to start, stop, and reset timers. *See Also Dynamic Timers.*

Tokens (Custom applications)

Custom applications can pass SessionID, CSRF, and identifying tokens into the Axeda Platform, allowing you to navigate between custom applications in extended tabs and the standard Axeda Applications with security in place. See the Axeda Applications *Using CSRF with Custom Applications* online help topic for more details.

TotalAccess Server

TotalAccess™ Server is an application server that processes remote sessions for Axeda IDM Agents. It is tightly integrated with the IDM Agents and the Axeda® Platform. To provide horizontal scalability and failover, the TotalAccess Server (TAS) supports multiple concurrent servers on one or more computers. The current cluster solution requires a commercial load balancer. All asset-side connections to the TAS cluster are made through a single URL managed by the load balancer that distributes the load across the configured TotalAccess servers. The TotalAccess servers may reside on the same physical/virtual host or on different hosts, as long as there are no conflicts with addresses and port numbers.

The TotalAccess application records the connections for each TotalAccess server in the cluster and directs the TotalAccess HCC (“client” side ActiveX control) to the one TotalAccess server that is managing the connection(s) for the desired application. This is known as the “client URL”. For more information about the TotalAccess Server, refer to the installation guide for the Total Access Server.

Tracking Asset

A tracking asset is a mobile (typically) tracking device, equipped with radios or modems for various wireless communication technologies, such as GSM (Global System for Mobile communications, CDMA (Code Division Multiple Access), and satellites. Tracking assets often contain GPS receivers and either built-in or wire-connected sensors. These assets provide the communication link between a mobile asset and the Axeda Platform. They transmit data to

the Platform over GPRS. A wide variety of tracking assets are available, some that can be programmed in Java and others that provide much more limited capability (can send only a few data items, for example).

A tracking asset is configured in the Axeda Configuration application like other assets. It has a model and an individual asset configuration. The model has adjunct metadata linked to it; this metadata is referred to as a profile. The profile represents the type of tracking asset. It also describes, in detail, the characteristics and capabilities of the asset. Details include the manufacturer, model, transport capabilities, sensory data the asset can report, and messaging capabilities.

To communicate with tracking assets, the Platform uses pluggable transport adapters that are installed in the Platform and configured in the profile of the asset. A transport adapter may be specific to wireless channels that the asset supports, such as GSM. An example of a transport on a tracking asset is a wireless radio that communicates over GSM using GPRS.

A tracking asset can have one or more communications components, depending on the complexity of the asset. For each communication component, there is a particular way to identify the asset, such as a SIM ID. When the Platform needs to communicate directly with a tracking asset, the appropriate identification information must be passed to the carrier network in the message. This information is stored in the Platform with the properties of the tracking asset. The transport adapter uses the appropriate property to initiate communications with the tracking asset.

Trigger

A trigger is a condition that causes an action (operation) to run when the condition evaluates to true. Components and conditions that generate triggers include alarms, changes in data item values or in extended data items, changes in alarm severity or state, file uploads, changes in the mobile location of an asset, timers, registration, change in the state of a state machine, and user login or logout events.

You can configure a change in the value of a data item as a trigger. Use the Logic Schema dialog box in Axeda Builder to associate triggers and actions.

Triggers (for expression rules)

In the context of expression rules, triggers are internal or external messages or stimuli that can be evaluated to determine if the Platform executes the associated rules. Rules are configured to run based on associated triggers. When a trigger message occurs, the Platform runs any associated rules. The `If` expression (the condition) evaluates only when its specified trigger message occurs.

The trigger you select for a rule determines which functions or actions you should configure in the rule. Not all functions and actions can be used for each trigger type.

Trigger messages can be created when assets report data or information, or when a rule runs an action that creates a trigger message, or by the Axeda Platform SDK

Triggers fall into two categories, asset-related and system-related. In general, use only asset-related namespaces with the asset-related triggers. You cannot use the system-related User namespace with asset-related triggers.

The asset-related triggers currently supported for expression rules are:

- **Alarm** — Alarm messages trigger whenever an alarm is created in the Platform. Alarms that have been disabled or suppressed do NOT trigger an Alarm message. The Alarm may have been reported by an asset or generated from a rule. When an Alarm message occurs, the rules engine evaluates all rules assigned to the Alarm trigger. You can use Alarm namespace symbols to evaluate Alarm expression rules.
- **AlarmExtendedDataChange** — AlarmExtendedDataChange messages trigger whenever the extended data associated with an alarm is changed. Extended data is additional, custom information that may be associated with alarms. Several Alarm namespace symbols are provided for use when evaluating AlarmExtendedDataChange expression rules.
- **AlarmSeverityChange** —- AlarmSeverityChange messages trigger whenever the severity of an alarm is changed. This trigger can be used to track alarm severity levels and react (via an Expression rule action) to a change in severity. Several Alarm namespace symbols are provided for use when evaluating AlarmSeverityChange expression rules.

- **AlarmStateChange** — AlarmStateChange messages trigger whenever the state of an alarm is changed by a SetAlarmState action Several Alarm namespace symbols are provided for use when evaluating AlarmStateChange expression rules.
- **AssetTimer** — AssetTimer messages trigger whenever the schedule for a timer is reached (for example, the timer period defined by the related timer schedule is reached). When an Asset Timer message occurs, the Platform creates separate messages for each device associated with the rules assigned to the AssetTimer trigger. This ensures the Platform applies the message for each device associated with the rules defined for that AssetTimer.
- **Data** — Data messages trigger whenever data items change, such as when an asset reports new readings, or a rule modifies a data item. Some functions can only be used with Data triggers; see the list of functions. When a Data message occurs, the rules engine evaluates all rules assigned to the Data trigger.
- **Event** - Event messages generate when an agent sends an event message to the Platform, such as when an asset reports new readings. When an Event message occurs, the Platform evaluates all rules assigned to the Event trigger.
- **File** — File upload messages are generated when a file upload completes. An expression rule can evaluate various attributes for the File namespace to then take appropriate action.
- **MobileLocation** — MobileLocation messages generate whenever a new location (set of latitude/longitude coordinates) is received for an asset. For example, when a tracking asset reports a new location (latitude, longitude coordinates). When a MobileLocation message occurs, the rules engine evaluates all rules assigned to the MobileLocation trigger.

  Note: *When specifying the Location variable for a rule assigned to a MobileLocation trigger, you can specify an unqualified "location". For all other triggers, you need to qualify this variable as "Location.location.*

- **Registration** — Registration messages generate when an asset contacts the Platform, whether for the first time or after having been offline for a period of time. Use the Registration namespace and its attributes in an expression rule if you want to take action based on this message.
- **StateChange** — StateChange messages generate whenever the state of a named state machine changes, for example, the state machine moves from one state to another. Using the *isentry* symbol, you can configure rules that evaluate when a state machine is entering or leaving a state (`If StateChange.isentry`). It is expected that expression rules for this trigger type are used within state machine states. (Note that state machines can be created and managed using Axeda Platform Web Services. Refer to the *Axeda® Platform Web Services Developer's Reference Guide* for complete information about state machine support.) Several State namespace symbols are provided for use when evaluating StateChange expression rules.

The system-related triggers currently supported for expression rules are:

- **SystemTimer** — SystemTimer messages generate whenever the schedule for a timer is reached. For example, a SystemTimer message may be generated at the beginning of the next hour (for a rule timer schedule of “hourly”) or on the first day of each month (for a rule timer schedule of “1st day of each month”. Note that this is not the actual syntax for defining timer schedules.) When a Timer message occurs, the rules engine evaluates all rules assigned to the SystemTimer trigger.

  Note: *Expression rules associated with System Timer, UserLogin, and UserLogout messages (triggers) cannot contain actions that require information regarding an asset. Actions supported with these system triggers are PublishObject, EnableRule, and DisableRule.*

- **UserLogin** and **UserLogout** — These messages generate whenever a user logs in to or logs out of the Axeda Applications. The attributes of the User namespace allow you to take action based on their values.

U

Unattended operation (Axeda Access or Axeda Desktop)

The mode of operation set for Remote Sessions on the Axeda Desktop server or Access Remote computer that allows users to connect to the computer without the intervention of an operator at the Axeda Desktop server or Access Remote computer. The operator at the Desktop Viewer or Access Viewer computer must connect to the session within a specified timeout period.

When you want to create a session with an Axeda Desktop server or Access Remote computer configured for unattended operation, select **Manage Remote Hosts** from the Home page of the Axeda Access application. The Manage Remote Hosts page shows all computers configured to allow unattended operation, and lets you create a new session for a specific Axeda Desktop server or Access Remote computer.

The session is shared once the Desktop Server or Access Remote computer contacts the Platform (based on a user-defined poll rate) and connects to the waiting session, and once the Access Desktop or Access Viewer operator types the session ID and password within the defined timeout period.

See also: *Axeda Administration*

Unit (for data items)

The unit of measure for the value that is represented by the data item. For instance, a data item representing an internal temperature could be measured in degrees Fahrenheit, Celsius, or Kelvin. You can specify a unit of measure for that data item to remove ambiguity and enable more accurate evaluation of the values of the data item.

In the Axeda Applications, the unit of measure assigned to a data item is shown anytime data values are displayed, for example from the Data and Chart pages of the Axeda Service application.

An asset can send its own units of measure to the Axeda Platform along with its data. In addition, you can create units of measure from the Axeda Configuration application. The unit attribute for data items is optional.

Usage Item (Axeda Usage)

A measurement in discrete units of the uses of an asset. Examples of usage items include number of black and white copies, number of blood tests, among others. A usage item is not tied to a particular model. Rather, a usage item is the result of a calculation performed on one of a set of data items. That is to say, different data items in different models may be transformed into the same usage item. For example, two copier models have different data items that both mean “number of black and white copies:”

1. One usage item calculates the running total for the asset.
2. Another usage item is a delta from the last reading.

For example, if a copier has made 10,000 copies as of Monday and 11,000 copies as of Tuesday, the Tuesday delta value is stored as “1000.” Usage items are defined as arriving from the asset as either a delta or a running total. This definition enables the application to determine which of the two values needs to be calculated. If your environment requires, you can write custom components that perform additional calculations on usage items before they are stored. The data from assets are transformed into usage data on a pre-configured, but system-configurable, schedule (using the XML configuration file of the Scheduler component). By default, the calculation thread runs twice a day (every 12 hours).

> **Note:** *When a usage item is configured, an optional configuration parameter indicates that the usage calculation should be performed in real time (when the asset data is processed) rather than on a schedule.*

Users (Axeda Applications)

In the Axeda Applications environment, users are the individuals with login access to the Axeda Applications. These individuals must have user accounts and associated user groups within the's directory service (LDAP). Authentication for the applications distinguishes among the following types of users:

- Non-administrative with no LDAP responsibilities — Typically, this type of user is assigned to one or more user groups. Through the privileges assigned to those user groups, this user has access to multiple assets and application tools.
- Non-administrative with LDAP responsibilities — This type of user needs to be able to modify LDAP information for users and/or user groups. These users can log in to the LDAP directory service through the Axeda Administration application; they must be members of the ServiceLinkUsers group and have access to an account defined in the ServiceLinkLdapAdmins group in LDAP. These users cannot edit the information of Administrative users nor can they change non-administrative users into Administrative users.
- Administrative —This type of user has access to all user groups, privileges, and tools available in the Axeda Applications. By default, only Administrative users can access the Axeda Administration application. Non-administrative users who need to access the Administration application must be members of user groups to which the privileges for the Administration application have been assigned, including the "Administration - View" privilege.

User Groups (Axeda Applications)

A collection of user accounts to which privileges to access assets and use Axeda Applications are assigned. You can create user groups in the LDAP directory service and in the Axeda Administration application. When you create user groups in the directory service, they become available for Axeda Applications. When you delete a user group from the Administration application, the user group is no longer available in the Axeda Applications and is also removed from the directory service.

> **Notes:** *By default, two groups must be created in the directory service for Axeda Applications, ServiceLinkAdmins and ServiceLinkUsers. In addition, for Sun ONE directory service implementations, the ServiceLinkLdapAdmins group must be created for non-admin users who require LDAP access. Refer to the installation guide for the Axeda Applications Platform for more information.*
>
> *All individuals who require access to the Axeda Applications must be defined in the ServiceLinkUsers group. In addition, any users who require administrative access to Axeda Applications must be defined in the ServiceLinkAdmins group (as well as the ServiceLinkUsers group). Any non-administrative users who require LDAP access must have access to an account defined in the ServiceLinkLdapAdmins group.*

User groups are assigned to asset groups such that only those user groups can view and manage the selected asset groups, In addition, user groups are assigned to rules and actions so that only users belonging to those user groups can change properties for a rule or action. *See also Asset Group (Axeda Configuration)*.

V

Validate (Data Items)

A Recipe in Axeda Builder enables you to test that the values you enter in the Recipe for a given data item are valid. Right-click on the Recipe row, and select **Validate data item** on the context menu. Axeda Builder returns a message with the results of the validation.

Validate (Expression Rules)

The pages of the Axeda Configuration application allow you to test an expression rule to be sure that it will run properly. The Platform does not allow you to save an expression rule that cannot be validated.

Value Alarms (Axeda Builder)

In the context of Axeda Builder and the Axeda Gateway and Axeda Connector Agents, value alarms identify data values that should trigger an alarm state. When configuring value alarms, you can choose among the following levels:

- LoLo and Low alarms occur when the data item value reaches or falls below the defined alarm value.
- High and HiHi alarms occur when the data item value reaches or rises above the defined alarm value.

If the data item value is within the defined limits, the alarm is inactive. If you set a deadband (percentage) for the alarm value, the alarm event becomes inactive when the data item value drops back within the defined alarm value range, beyond the value of the deadband. The deadband is a percentage of the full data item value range.

For example, you can define a HiHi value alarm to occur for an analog data item when the data item's value is 80 or higher (based on user-specified engineering units). If the data item's full value range is 0 to 100 and you set a deadband of 10, the data item is removed from HiHi alarm condition when the data item value falls below 70, which is 80 - [10% x (100 - 0)].

Variables (Expression Rules)

When creating an expression rule, you can use Variables in the expressions to represent Axeda Platform objects such as alarms, data items, locations, assets, locations, files, registrations, triggers, or users, and a specific attribute of the related object. A Variable consists of a domain object, or *namespace*, a period, and an symbol. For example, in the Variable, Alarm.severity, "Alarm" is the namespace, the period is the delimiter, separating the namespace from its attribute, and "severity" is the symbol.

When you select a trigger, that trigger provides a context for the Variables used in the expression rule. For example, if you select a DataItem trigger, then you can specify the name of a particular data item and a symbol for that data item without explicitly using the namespace. An Axeda “best practice” is to always use the namespace when entering the Variables in an expression.

W

Warm Restart

The ability of an Axeda Gateway or Axeda Connector Agent to recover data after an unexpected or regular shutdown. If this feature is enabled, the Agent starts a special thread at runtime that periodically instructs various components to save their data. The default period is 15 seconds. For performance reasons, it is recommended that you do not use a shorter period than the default. The data from the components is saved to disk, in special warm restart files. These files are saved and used to restore the components that were active at the time of shutdown, the values and states of data items, alarms that had not been sent, expressions in any state, the current values of counters, pending e-mail messages, pending messages in the queue for the Axeda Platform, and more. On restart, there may be some duplication of e-mail or messages sent to the Axeda Platform.

Web Alert

A notification generated by the Axeda Platform based on a predefined configuration. Users of Axeda Applications can determine when new alerts are generated for them, and they can view and manage all of their current, unacknowledged alerts. Any condition that generates a notification is shown to the user as a Web alert.

Web Visualization Applet

The browser-based engine that accompanies Axeda Connector Agents and Axeda Builder. Also known as Web@aGlance (WAAG), this engine enables remote operators to view Axeda Connector project displays from web browsers.

Wireless

In the context of the Axeda Platform Axeda Applications, the communication medium that transfers data over long or short distances, *without* electrical conductors (“wires”). Wireless devices communicating over supported wireless carrier networks can communicate with the Axeda Platform. Examples of wireless devices include equipment with embedded sensors and bar scanners. You need to configure the wireless device and Axeda Platform to communicate with each other.

The following figure shows how wireless devices communicate with Axeda Platform: