Database Schema

Tables

Authorizations

This table logs all active User access tokens that are required for access to the API. Authorizations marked as IsInteractiveSession = true will remain active as long as the User keeps accessing the API. After 30 days of inactivity they will be automatically removed by the system. Authorizations marked as IsInteractiveSession = false will persist forever. The unhashed version of the token can only be accessed when the Authorization is first created. It is up the client application or User to cache the token.

AuthorizationsActive

This table logs all currently active User sessions.

AuthorizationsActiveLog

This table logs all of the previously active User sessions.

BeaconEvent

This table contains the top level definitions of Beacon Events defined in the system. The details of the Sites used in the Beacon Events are accessed via the BeaconEventMsgtable.

BeaconEventActivationLog

When a Beacon Event Activation is created, edited, deleted or it has expired, the changes will be logged in this table.

BeaconEventActivations

Beacon Events can be manually activated by a Beacon Event Activation and this table stores all of the active instances.

BeaconEventMsg

The information in this table is associated with the Beacon Events defined in the BeaconEvent table. This table contains the Beacon Message and direction code details of each Site in every Beacon Event.

BeaconEventMsgLog

This table logs the Broadcasting of Beacon Event messages while the Beacon Event is active. It contains the details of every Site in the Beacon Event and is updated every 15 seconds. The information in this table is associated with the Beacon Events defined in the BeaconEvent table.

BeaconEventStaticBeaconMsg

The information in this table is associated with the Beacon Events defined in the BeaconEvent table. This table contains the Beacon Message and direction code details of each Static Beacon in every Beacon Event.

BeaconIncidentChangeLog

This table logs any changes Users make through the life cycle of an Incident. This includes automated system changes that occur such as the initial creation and final closure of Incidents. These are logged against the UserId = 0. This table is associated with the Incidents defined in the BeaconIncidents table.

BeaconIncidentCounter

This table is used by the AiIncidentManager service on startup to know which Id to use for the next Incident that is created. The service will increment this value after each Incident is created.

BeaconIncidentDelayCodes

This table stores the Incident Delay Codes defined in your system.

BeaconIncidentLinks

The Broadcasting locations for an Incident are based on the Links defined in this table. The origin Site of each Link is used as a Broadcasting location. This table is associated with the Incidents defined in the BeaconIncidents table.

BeaconIncidentMsgLog

This table logs the broadcasting of Incident messages. It contains the details of every Site that was used by the Incident. It is updated every 30 seconds. This table is associated with the Incidents defined in the BeaconIncidents table.

BeaconIncidents

This table contains the top level definitions of Incidents created in the system. The details of the Links used in the Incident are accessed via the BeaconIncidentLinks table.

BeaconIncidentTypes

This table stores the Incident Types defined in your system.

BeaconMsg

This table stores the Beacon Messages defined in your system.

BeaconSchedule

This table stores the basic definition for every Schedule defined in your system. The full definition of a Schedule includes the information defined in the BeaconScheduleDates, BeaconScheduleSpecDayTypes and BeaconScheduleTimeOfDay tables.

BeaconScheduleDates

This table stores the manually-specified dates that can be used to activate and deactivate a Schedule. This table is associated with the Schedules defined in the BeaconSchedule table.

BeaconScheduleLog

This table log when a Schedule becomes active or inactive based on its time of day and date criteria. This table is associated with the Schedules defined in the BeaconScheduletable.

BeaconScheduleSpecDayTypes

This table stores the Special Day Types that can be used to activate and deactivate a Schedule. This table is associated with the Schedules defined in the BeaconScheduletable.

BeaconScheduleTimeOfDay

This table stores the times of day that can be used to activate and deactivate a Schedule. This table is associated with the Schedules defined in the BeaconScheduletable.

BeaconTTDirectionPriority

This table stores the basic definition for every Broadcast Travel Time Direction defined in your system. The full definition of a Broadcast Travel Time Direction includes the information defined in the BeaconScheduleDates, BeaconScheduleSpecDayTypes and BeaconScheduleTimeOfDay tables.

BeaconTTRouteMsgLog

This table logs the Broadcasting of Broadcast Travel Time Direction messages. It contains the Travel Time details of every Route defined in the Broadcast Travel Time Direction. It is updated every 30 seconds. This table is associated with the Broadcast Travel Time Directions defined in the BeaconTTDirectionPriority table.

BeaconTTRoutes

The information in this table is associated with the Broadcast Travel Time Directionsdefined in the BeaconTTDirectionPriority table. This table contains the Routes assigned to the Broadcast Travel Time Directions and the order that they are spoken in.

BTBestLinkClusters

Addinsight determines the best Clusters to use for each Link based on the most recent set of Clusters generated that are accessible via the BTLinkClusters table. This table contains the top-scoring Clusters based on comparison dates two days into the past and 7 days into the future for each Cluster Period. The full list of Scores for each Clustercan be accessed via the BTLinkClusterScores table.

BTClusterGroups

This table stores the basic definition for every Cluster Group defined in your system. The full definition of a Cluster Group includes the information defined in the BTClusterPeriods table.

BTClusteringDateScores

To determine the best Clusters in the BestLinkClusters table, the system calculates the Date Score for each day from the past year based on comparison dates two days into the past and 7 days into the future. This table contains the Score summary information for the components that make up the Score.

BTClusteringSpecialDayTypesDateScores

To determine the best Clusters in the BestLinkClusters table, the system calculates the Date Score for each day from the past year based on comparison dates two days into the past and 7 days into the future. This table contains a breakdown of the Special Day Type Scores that contribute to the summary information in the BTClusteringDateScorestable.

BTClusterPeriods

The information in this table is associated with the Cluster Groups defined in the BTClusterGroups table. This table contains the Cluster Periods assigned to the Cluster Groups.

BTDeviceRawStats

This table logs the raw detection statistics for the Devices defined in the BTDevicestable. The statistics are generated every 15 seconds, but information is only logged if Probes are detected for a particular Source Id Type during that interval. For each interval there may be more than one entry per Devices if it capable of detecting more than one Source Id Type. This table is good for diagnosing issues with detections for a specific Source Id Type.

BTDevices

This table stores the Devices defined in your system. Devices are associated with Sitesthat are defined in the BTSites table.

BTDeviceStats

This table logs summary detection statistics for the Devices defined in the BTDevicestable. The statistics are generated every minute even if no Probes are detected during that interval. This table is good for diagnosing issues within PDUs and communication issues.

BTIeeeVendors

This table stores the names of Bluetooth hardware vendors assigned specific vendor codes (first 6 characters of a MAC Address). This is a lookup table that can be used in combination with the BTProbeVendorStats table.

BTLink5MinuteStats

This table logs summary detection and Travel Time statistics for the enabled Linksdefined in the BTLinks table. The statistics are generated every 5 minutes. This table is used by the Clustering process which runs in the AiClustering service. This is calculated by the AiLinkCalc service.

BTLinkClusterData

This table logs summary detection and Travel Time statistics for every Cluster generated by the Clustering process. The data is summarised for each 5 minute interval within a Cluster. This table is generated by the AiClustering service. The contents of this table are associated with the Clusters defined in the BTLinkClusters table.

BTLinkClusterDays

This table logs the dates grouped within every Cluster generated by the Clusteringprocess. This table is generated by the AiClustering service. The contents of this table are associated with the Clusters defined in the BTLinkClusters table.

BTLinkClusters

This table logs the basic summary information about the Clusters generated by the Clustering process. This table is generated by the AiClustering service. The full definitions of Clusters defined in this table need to reference the BTLinkClusterDays and BTLinkClusterData tables.

BTLinkClusterScores

Addinsight determines the best Clusters to use for each Link based on the most recent set of Clusters generated that are accessible via the BTLinkClusters table. This table contains all of the Scores for all Clusters based on comparison dates two days into the past and 7 days into the future for each Cluster Period. The short list of best-scoring Clusters for each Link can be accessed via the BTBestLinkClusters table.

BTLinkCongestionStats

This table logs summary performance statistics for the enabled Links defined in the BTLinks table. The statistics are generated every 30 seconds by the AiCongestionCalcservice.

BTLinkEmulatedTT

This table stores optional configuration criteria associated with setting up Links to fall back to emulated Travel Times. This table references the Links defined in the BTLinkstable.

BTLinkExcludedSourceIdTypes

This table stores optional configuration criteria associated with setting up Links to ignore Probe data from specific Source Id Types. This table references the Links defined in the BTLinks table.

BTLinkExpectedStats

This table logs the expected (predicted) performance statistics for the enabled Linksdefined in the BTLinks table. The statistics are logged every 5 minutes by theAiCongestionCalc service. This allows retrospective calculation of Excess Delay and Congestion Parameter values that are based on both the live and expected Travel Timevalues.

BTLinkFreeFlowTT

This table stores the free-flow Travel Times of every enabled Link. This is calculated daily by the AiLinkCalc service based on the Travel Times calculated across the previous day. This table references the Links defined in the BTLinks table.

BTLinkGeometry

This table stores the ordered sequence of point coordinates that make up the geometry of each Link. This table references the Links defined in the BTLinks table.

BTLinkIgnoreClose

This tables stores every request to set a Link as closed or ignored. This table references the Links defined in the BTLinks table and the Users defined in the Users table.

BTLinkParams

This table stores optional configuration criteria associated with the triggering of Incidents on a Link to override the system-wide thresholds. This table references the Links defined in the BTLinks table.

BTLinks

This table stores the Links defined in your system. Links are associated with Sites that are defined in the BTSites table.

BTLinkScores

This table stores the Link Scores defined in your system.

BTLinkStats

This table logs summary detection and Travel Time statistics for the enabled Linksdefined in the BTLinks table. The statistics are generated every 30 seconds. This is calculated by the AiLinkCalc service.

BTProbeBlacklist

This table stores the Blacklisted Probes defined in your system.

BTProbeCounter

This table is used by the AiMessageBus service on startup to know which Id to use for the next Probe that is created. The service will increment this value after a batch of Probes is created.

BTProbeODs

This table stores the origin and destination Site for every Probe. This occurs when the Probe expires. This does not necessarily contain the actual O-D for the Probe because it may make multiple trips before it expires. This table references the Site defined in the BTSites table.

BTProbes

This table stores the Probes currently being tracked in your system.

BTProbeTravelTimes

This table stores Probe Travel Times if the pass through an origin and destination Sitethat matches the definition of a Link. This table references the Links defined in the BTLinks table and also the Probes in the BTProbes table (only until they expire).

BTProbeVendorStats

Every day the system will provide a count of the number of Bluetooth Probes detected for each vendor code. These can be cross-referenced against vendors defined in the BTIeeeVendors table to get their name. This may double-count data if a Probe expires more than once throughout the day. Each existence of a Probe will increment the counters.

BTRecords

This table stores every Site that a Probe passes through. A Probe can be detected multiple times as it passes through a Site and this table stores the first and last time of detection.

BTRouteLinks

This table stores the ordered sequence of Links that make up a Route. This table references the Routes defined in the BTRoutes table.

BTRoutes

This table stores the Routes defined in your system.

BTRouteStats

This table logs Travel Time predictions and other statistics for the enabled Routesdefined in the BTRoutes table. The statistics are generated every 30 seconds. This is calculated by the AiRouteCalc service.

BTSiteActiveTimeInRangeStats

This table logs a count of unique Probes and their time in range statistics per 1-minute interval at every enabled Site defined in the BTSites table. This is calculated by the AiBTListener service.

BTSiteBeaconStatus

This table logs the Major and Minor values being Broadcast at every Site as part of an iBeacon Message. The statistics are generated every 15 seconds. This is calculated by the AiBeaconManager service.

BTSiteRawStats

This table logs a count of Probe records captured per 15-second interval at every enabled Site defined in the BTSites table. This is calculated by the AiBTListener service.

BTSites

This table stores the Sites defined in your system.

BTSiteStats

This table logs a count of Probe records, unique Probes and their average time in range per 5-minute interval at every enabled Site defined in the BTSites table. This is calculated by the AiBTListener service.

BTSpecialDays

This table stores the dates that have been defined in a Special Day Type. This table references the Special Day Type defined in the BTSpecialDayTypes table.

BTSpecialDayTypes

This table stores the Special Day Types defined in your system.

BTTrackedVehicles

This table stores the Tracked Vehicles defined in your system. This table references the Tracked Vehicle Types defined in the BTTrackedVehicleTypes table and also the probes defined in the BTProbes table.

BTTrackedVehicleTypes

This table stores the Tracked Vehicle Types defined in your system.

Cameras

This table stores the Cameras defined in your system.

ClientConfiguration

This table stores the Client Configurations of Users defined in your system. This table references the Users table.

ClientErrorLogs

This table stores the Client Error Logs. This table references the Users table.

CounterHistFilesConverted

This table stores a list of all of the SCATS History Files that have been converted (or attempted to be converted) by the AiSCATSImporter service.

Counters

This table stores the Counters defined in your system. A Counter must belong to a Counter Site, which are defined in the BTSites table.

CounterSitePhaseStats

This table stores the Counter Site Phases defined in your system.

CounterSites

This table stores the Counter Sites defined in your system. This table references the Counter Sources defined in the CounterSources table.

CounterSources

This table stores the Counter Sources defined in your system.

CounterStats

This table logs Counter count data. This is uploaded by the AiSCATSImporter service service as it reads in new VS Files. The data can also be added via the API. This table references the Counters defined in the Counters table.

CounterVSFilesConverted

This table stores a list of all of the SCATS VS Files that have been converted (or attempted to be converted) by the AiSCATSImporter service.

ForwardDataTargets

This table stores the Forward Data Targets defined in your system.

GroupExcludedObjects

This table stores the objects that are explicitly excluded from Groups defined in your system. It references the Groups defined in the Groups table.

GroupObjects

This table stores the objects that are included in every Group defined in your system and this is done automatically by the system via the AiGroup service. It references the Groups defined in the Groups table.

Groups

This table stores the Groups defined in your system. This table stores the basic group information such as its perimeter. If you have specified excluded objects for a Groupthey are defined in the GroupExcludedObjects table. The system automatically determines the included objects and these are stored in the GroupEObjects table.

HardwareStatusTypes

This table contains Hardware Status Types defined in your system. These are only used for fault reporting by some VMS. These are not to be edited.

IdentityProviders

This table contains Identity Providers defined in your system. These are used to connect Addinsight to a user directory via LDAP.

IncidentOverlayGeometry

This table contains the geographical objects associated with the Incident Overlaysdefined in the IncidentOverlays table.

IncidentOverlays

This table contains Incident Overlays defined in your system. This table stores the basic textural information and the geographical objects are defined in the IncidentOverlayGeometry table. This table also references the Incident Overlay Typesdefined in the IncidentOverlayTypes table.

IncidentOverlayTypes

This table contains Incident Overlay Types defined in the system. These are not to be edited.

Logs

This table stores the System Logs that are generated by the Addinsight services. This table is good for fault monitoring and investigation, but it is only useful if the services have been able to establish a database connection. If not, you will need to look at the Windows Event Viewer for more information.

NamedQueries

This table contains Named Queries defined in your system.

NetworkStats

This table stores all of the data for the defined Network Statistics in the system.

ProjectRouteProbeTTs

This table is used for a specific project in South Australia.

Replicas

This table contains Replicas defined in your system.

ReportQueryLog

This table logs every Report Query requested by Users on your system.

SchoolCrossings

This table contains School Crossings defined in your system. This table stores the basic information such as the IP Address and location and the remaining data such as the calendar and time of day schedules are stored in
the SchoolCrossingYearlySchedulesand SchoolCrossingWeeklySchedules tables respectively.

SchoolCrossingScheduleChangeLog

This table logs every attempt to alter the schedule of a School Crossing for fault monitoring purposes.

SchoolCrossingStats

This table logs the status of each School Crossing every 5 minutes for fault monitoring purposes.

SchoolCrossingWeeklySchedules

This table stores the times of day and the associated days of the week that are used to activate the lights at a School Crossing. This table references the School Crossingsdefined in the SchoolCrossings table. It is used in combination with the yearly schedules defined in the SchoolCrossingYearlySchedules table.

SchoolCrossingYearlySchedules

This table stores the dates that are used to activate the lights at a School Crossing. This table references the School Crossings defined in the SchoolCrossings table. It is used in combination with the weekly schedules defined in the SchoolCrossingWeeklySchedulestable.

ServiceOperationStats

This table periodically logs execution details about the various threads and methods running on the Addinsight services. It can be used to assist with the diagnosing of system performance and bottlenecks.

StaticBeacons

This table contains Static Beacons defined in your system.

SystemParameters

This table contains the System Parameters defined for your system. Only the values in the value column are to be edited or you should use the Settings page of the Real Time Client to make changes via the API

TableColumnMetadata

This table is used by Addinsight to know which columns can be ignored in the database triggers that are automatically created as part of the installation process. This table should not be edited.

TableMetadata

This table is used by Addinsight for many purposes and it should not be edited apart from the RetentionAge column where non null values already exist. Ideally it should not be edited at all. Addinsight uses this table to know which tables can be ignored in the database triggers that are automatically created as part of the installation process. It is used to store the Data Retention Policies that is used by the AiDataRetention service to remove old log data from specific tables. It is also used by the AiReplication service to know which tables are to be replicated for the Replicas defined in the system.

TilesetImages

This table stores the image tile for a specific grid location for a specific zoom level. This table is referenced by the TilesetMap table and also references the Tilesets defined in the Tilesets table.

TilesetMap

This table stores the identifier of the tile that belongs at a specific grid location for a specific zoom level. This table references the images defined in the TilesetImages table and also references the Tilesets defined in the Tilesets table.

Tilesets

This table contains Tilesets defined in your system. This table stores the basic information about the name and extent of the Tilesets and the remaining data such as the tile identifiers and tiles themselves are stored in the TilesetMap and TilesetImagestables respectively.

UserGroupMembers

This table stores the Users assigned to a User Group. This references the User Groupsdefined in the UserGroups table and the Users defined in the Users table.

UserGroupPermissions

This table stores the Permissions assigned to a User Group. This references the User Groups defined in the UserGroups table.

UserGroups

This table contains User Groups defined in your system. The Permissions for a User Group are defined in the UserGroupPermissions table. The Users assigned to a User Group are defined in the UserGroupMembers table.

Users

This table contains Users defined in your system. Permissions for a User are based on the User Groups that they are members of. This is defined in the UserGroupMemberstable

VariableMessageSignPanels

This table contains Variable Message Sign Panels defined in your system. This table references the Variable Message Signs defined in the VariableMessageSigns table.

VariableMessageSignPanelStatusLog

This table contains status information about Variable Message Sign Panels defined in your system.

VariableMessageSigns

This table contains Variable Message Signs defined in your system. The Variable Message Sign Panels that display the Travel Times are defined in the VariableMessageSignPanels table.

VariableMessageSignStatusLog

This table contains status information about Variable Message Signs defined in your system.

VectorMapSetLines

This table contains the polyline and polygon geometric data for the relevant Vector Map Sets defined in the VectorMapSets table.

VectorMapSetPoints

This table contains the point geometric data for the relevant Vector Map Sets defined in the VectorMapSets table.

VectorMapSets

This table contains Vector Map Sets defined in your system. This table only contains the basic definitions of the Vector Map Sets and the geometric data are stored in either the VectorMapSetPoints or VectorMapSetLines tables.