ASP Mining Starter Pack - Data Dictionary Classes

Version	Author	Comments
2020.09.10	Andres Ulloa	Document Creation

ASPMining.CanonicalModel.Equipment.Equipment

Parameters

- parameter **DSTIME** = "AUTO";

Properties

- property **Capacity** as %Float;

Capacity of Vehicle ASSIGNED BY USERS

- property **EquipmentCategory** as EquipmentCategory;

Category of Equipment, eg: Truck, Excav.

- property **EquipmentModel** as EquipmentModel;

Model of Equipment: e.g: CAT 770

- property **Id** as %String;

Mine Unique Identifier of Vehicle

- property **LastProductionUpdate** as ASPMining.CanonicalModel.Production.ProductionEventSliced;

Used to keep track of travel/unknown times in the production cycle, for production slices only

- property **LastUnifiedEvent** as ASPMining.CanonicalModel.Operation.UnifiedEvents;

Used to check Last production DumpEvent

- property **Name** as %String;

Readable Display Name of Vehicle

Indices

- index **(EquipmentIndex** on **Id)** [Unique];

- index **(NameIndex** on **Name)** [Type = bitmap];

ASPMining.CanonicalModel.Equipment.EquipmentCategory

Properties

- property **Description** as %String;

Description of the Category, e.g: Newest fleet of excavators.

- property **Name** as %String;

Display name of the Category e.g: Truck, Excav, etc.

Indices

•index **(EquipmentCategoryIndex** on **Name)** [Unique];

ASPMining.CanonicalModel.Equipment.EquipmentModel

Properties

- property **Brand** as %String;

Manufacturer's brand name

- property **Description** as %String;

Model Description

- property **Model** as %String;

Model of the equipment provided by manufacturer

- property **NominalCapacityTons** as %Float;

Manufacturer's specified Tonnage of equipment (if applies)

Indices

•index **(EquipmentModelIndex** on **Model)** [Unique];

ASPMining.CanonicalModel.HWMonitoring.OEMEvent

Parameters

- parameter **DSTIME** = "AUTO";

Properties

- property **Equipment** as ASPMining.CanonicalModel.Equipment.Equipment;

GPS Latitude if available GPS Longitude if available Local copy of EquipmentId for this event, not reference to an Equipment Object, in case it changes to a different equipment

- property **Id** as %Integer;

Unique Id of event provided by Mining software

- property **OEMEventType** as OEMEventType;

Type of the event, e.g: Overspeed event.

- property **ReadTime** as %DateTime;

HW Interface that generator this event Shift when this event happened Mine Site location where the event happened Reading time of the event

- property **Value** as %Float;

Value of the event (if it has one)

Indices

- index **(OEMEventIndex** on **Id)** [Unique];

- index **(ReadTimeIndex** on **ReadTime)**;

ASPMining.CanonicalModel.HWMonitoring.OEMEventType

Properties

- property **Description** as %String;

Assigned description of the event: e.g: "generated when driver is over accelerating"

- property **Other** as %String;

Other explainatory content: e.g: "for calibration only purposes"

ASPMining.CanonicalModel.HWMonitoring.OEMInterface

Properties

- property **Brand** as %String;

Brand of the Interface (optional)

- property **Name** as %String;

Display Name of the OEM Interface

Indices

- index **(BrandIndex** on **Brand)** [Type = bitmap];

- index **(NameIndex** on **Name)** [Type = bitmap];

ASPMining.CanonicalModel.HWMonitoring.OEMInterfaceMap

Properties

- property **Equipment** as ASPMining.CanonicalModel.Equipment.Equipment;

Equipment Reference where this Interface is connected to

- property **OEMInterface** as OEMInterface;

OEMInterface Reference

Indices

- index **(EquipmentIndex** on **OEMInterface)** [Type = bitmap];

- index **(OEMInterfaceIndex** on **OEMInterface)** [Unique];

ASPMining.CanonicalModel.Mine.Location

Properties

- property **Grade** as %String;

Grade of the material in this location

- property **Name** as %String;

Site to which this Location belongs

- property **Site** as Site;

Site to which this Location belongs

Indices

- index **(GradeIndex** on **Grade)** [Type = bitslice];

- index **(NameIndex** on **Name)** [Unique];

ASPMining.CanonicalModel.Mine.Site

Properties

- property **Name** as %String;

Display Name of the Site

Indices

- index **(NameIndex** on **Name)** [Type = bitmap];

ASPMining.CanonicalModel.Mine.Trip

Properties

- property **AvgTravelTime** as %Double;

Average Registered Time

- property **DestinationLocation** as Location;

Site to which this Location belongs

- property **Id** as %String(MAXLEN=70);

Unique Id of this Trip

- property **MaxTravelTime** as %Double;

Maximum Registered Time

- property **MinTravelTime** as %Double;

Minimum Registered Time

- property **Name** as %String(MAXLEN=70);

Site to which this Location belongs

- property **OriginLocation** as Location;

Site to which this Location belongs

- property **ReferenceTravelTime** as %Double;

Reference Time Used to Compare data automatically calculated.

- property **TotalTravelTime** as %BigInt;

Sum of all times of this trip, useful for calculations

- property **TravelCount** as %Double;

Amounts of Travels Performed

- property **UserReferenceTravelTime** as %Double;

Reference Time Used to Compare data specified by th euser

Indices

- index **(DestinationLocationIndex** on **DestinationLocation)** [Type = bitmap];
- index **(IdIndex** on **Id)** [Unique];
- index **(NameIndex** on **Name)**;
- index **(OriginLocationIndex** on **OriginLocation)** [Type = bitmap];

ASPMining.CanonicalModel.Operation.StatusEvent

Parameters

- parameter **DSTIME** = "AUTO";

Properties

- property **Duration** as %Double;

Calculated duration of the event

- property **EndDate** as %DateTime;

End date of the event

- property **Equipment** as ASPMining.CanonicalModel.Equipment.Equipment;

Equipment that generated this event

- property **Id** as %BigInt;

Unique Id

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Shift where this event happened

- property **StartDate** as %DateTime;

Start date of the event

- property **StatusReason** as StatusReason;

Status Reason Link

- property **StatusType** as StatusType;

Status Type Link

Methods

- classMétodo **HourToInt(datetime As %DateTime)** as %Integer

Indices

- index **(DurationIndex** on **Duration)** [Type = bitslice];

- index **(EndTimeIndex** on **EndDate)**;

- index **(EquipmentIndex** on **Equipment)** [Type = bitmap];

- index **(IdIndex** on **Id)** [Unique];

- index **(StartTimeIndex** on **StartDate)**;

- index **(StatusReasonIndex** on **StatusReason)** [Type = bitmap];

- index **(StatusTypeIndex** on **StatusType)** [Type = bitmap];

ASPMining.CanonicalModel.Operation.StatusEventSliced

Parameters

- parameter **DSTIME** = "AUTO";

Properties

- property **Equipment** as ASPMining.CanonicalModel.Equipment.Equipment;

Equipment that generated this event

- property **PartialDuration** as %Float;

Calculated duration of the event

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Shift where this event happened

- property **SliceStartDate** as %DateTime;

Start date of the event

- property **StatusReason** as StatusReason;

Status Reason Link

- property **StatusType** as StatusType;

Status Type Link

Indices

- index **(PartialDurationIndex** on **PartialDuration)** [Type = bitslice];

- index **(SliceStartTimeIndex** on **SliceStartDate)** [Type = index];

- index **(ddbkeyIndex** on **ddbkey)** [Type = index];

ASPMining.CanonicalModel.Operation.StatusReason

Properties

- property **Description** as %String;

Description for the Reason, e.g: "operator went to restroom"

- property **Id** as %String;

Reason Unique Id = Concat StatusType+Reason

- property **Reason** as %String;

Reason code, e.g: "112"

- property **StatusType** as StatusType;

Link to Status Type

Indices

- index **(IdIndex** on **Id)** [Unique];
- index **(ReasonIndex** on **Reason)** [Type = bitmap];
- index **(StatusTypeIndex** on **StatusType)** [Type = bitmap];

ASPMining.CanonicalModel.Operation.StatusType

Properties

- property **Description** as %String;

Description of the status, e.g: Operative, Standby, Downtime, Delay. Very important for Operational KPIs

- property **Type** as %String;

Type of the status, e.g: 1,2,3,4

Indices

- index **(TypeIndex** on **Type)** [Unique];

ASPMining.CanonicalModel.Operation.UnifiedEvents

This unifies the production events and operation events such that each row represents only 1 change in status or production for each equipment. The duration field of each field is only calculated when a new event is received and then the difference is calculated. LoadEvents + DumpEvents + StatusEvents

Parameters

- parameter **DSTIME** = "AUTO";

Properties

- property **Duration** as %Double;

Duration in seconds of this state, is the time between this interval and the next

- property **Equipment** as ASPMining.CanonicalModel.Equipment.Equipment;

Related Equipment has to exist

- property **EventDateTime** as %DateTime;

Saves the datetime of the received event to simplify the Duration calculation of next event

- property **Id** as %BigInt;

Unique Id

- property **OperativeDuration** as %Double;

Only Operative Duration of this state, will basically be 0 for any statusevent with Non Operative StatusType, and Duration if statustype is operative. It could exists an equivalent way to do this in the Cube.

- property **ProductionEvent** as ASPMining.CanonicalModel.Production.ProductionEvent;

Production Event, NEED TO ADD THIS UNIFIED TABLE

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Related Shift has to exist

- property **StatusEvent** as ASPMining.CanonicalModel.Operation.StatusEvent;

Status Event

- property **Trip** as ASPMining.CanonicalModel.Mine.Trip;

Trip Id

- property **TripTime** as %Double;

Trip Time of the most Recent Trip Made by this Truck

Indices

- index **(EquipmentIndex** on **Equipment)** [Type = bitmap];

- index **(EventDateTimeIndex** on **EventDateTime)**;

- index **(IdIndex** on **Id)** [Unique];

- index **(ProductionEventIndex** on **ProductionEvent)**;

- index **(ShiftIndex** on **Shift)** [Type = bitmap];

- index **(StatusEventIndex** on **StatusEvent)**;

ASPMining.CanonicalModel.Personnel.Crew

Properties

- property **Name** as %String;

Display Name of the Crew, usually 1 letter, e.g: A,B,C,D,E

Indices

- index **(CrewIndex** on **Name)** [Unique];

ASPMining.CanonicalModel.Personnel.Operator

Properties

- property **Crew** as Crew;

Current Crew Link

- property **FirstName** as %String;

Operator's first name

- property **LastName** as %String;

Operator's last name

Indices

- index **(FirstNameIndex** on **FirstName)** [Type = bitmap];

- index **(LastNameIndex** on **LastName)** [Type = bitmap];

ASPMining.CanonicalModel.Planning.Shift

Properties

- property **Crew** as ASPMining.CanonicalModel.Personnel.Crew;

Crew Assigned to this shift

- property **DateTime** as %DateTime;

Start time of the Shift

- property **Id** as %Integer;

Uniquie Shift Id

- property **ShiftType** as ShiftType;

Shift's Shift Type

- property **StartDay** as %Integer;

Auxiliar variable, offset in days from start of IRIS internal time

- property **StartSeconds** as %Integer;

Auxiliar variable, offset in seconds from start of the day

Indices

- index **(CrewIndex** on **Crew)** [Type = bitmap];

- index **(ShiftIndex** on **Id)** [Unique];

ASPMining.CanonicalModel.Planning.ShiftType

Types of shifts have a basic schedule day/night, and can have some extra information like holyday information

▼ Properties

- property **Other** as %String;

Any special details about this shift, e.g: holyday

- property **Type** as %String;

Shift Type, commonly day and night, e.g: A,B

▼ Indices

•index **(ShiftTypeIndex** on **Type)** [Unique];

ASPMining.CanonicalModel.Production.DumpEvent

Properties

- property **DumpId** as %BigInt;

Unique DumpId

- property **EquipmentExcav** as ASPMining.CanonicalModel.Equipment.Equipment;

Excavator Link

- property **EquipmentTruck** as ASPMining.CanonicalModel.Equipment.Equipment;

Truck Link

- property **Latitude** as %Float;

GPS of event (if available)

- property **Location** as ASPMining.CanonicalModel.Mine.Location;

Location where the Dump happened

- property **Longitude** as %Float;

GPS of event (if available)

- property **MeasuredTons** as %Float;

Measured tons of the Dump (if available)

- property **OperatorExcav** as ASPMining.CanonicalModel.Personnel.Operator;

Excavator's Operator Link

- property **OperatorTruck** as ASPMining.CanonicalModel.Personnel.Operator;

Truck's Operator Link

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Shift when this Dump Happened

- property **TimeArrive** as %DateTime;

Time when the Truck Arrived

- property **TimeDump** as %DateTime;

Time when the Load Started

- property **TimeEmpty** as %DateTime;

Time when the Load Completed

- property **TravelTime** as %Integer;

Calculated Travel Time of the vehicle to arrive at the dump point

Indices

- index **(DumpIndex** on **DumpId)** [Unique];

ASPMining.CanonicalModel.Production.LoadEvent

Properties

- property **EquipmentExcav** as ASPMining.CanonicalModel.Equipment.Equipment;

Excavator Link

- property **EquipmentTruck** as ASPMining.CanonicalModel.Equipment.Equipment;

Truck Link

- property **Latitude** as %Float;

GPS of event (if available)

- property **LoadId** as %BigInt;

Unique LoadId

- property **Location** as ASPMining.CanonicalModel.Mine.Location;

Location where the Dump happened

- property **Longitude** as %Float;

GPS of event (if available)

- property **MeasuredTons** as %Float;

Measured tons of the Dump (if available)

- property **OperatorExcav** as ASPMining.CanonicalModel.Personnel.Operator;

Excavator's Operator Link

- property **OperatorTruck** as ASPMining.CanonicalModel.Personnel.Operator;

Truck's Operator Link

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Shift when this Load Happened

- property **TimeArrive** as %DateTime;

Time when the Truck Arrived

- property **TimeFull** as %DateTime;

Time when the Load Completed

- property **TimeLoad** as %DateTime;

Time when the Load Started

- property **TravelTime** as %Integer;

Calculated Travel Time of the vehicle to arrive at the load point

▼ Indices

- index **(LoadIndex** on **LoadId)** [Unique];

ASPMining.CanonicalModel.Production.ProcessTimes

▼ Properties

- property **ExpectedTime** as %Integer;

ExpectedTime for this process at this site in seconds

- property **Location** as ASPMining.CanonicalModel.Mine.Location;

Location for this site

- property **ProductionStatusType** as ProductionStatusType;

Production Type of the process, this table should not have "traveling" types

▼ Indices

- index **(LocationIndex** on **Location)**;
- index **(ProductionStatusTypeIndex** on **ProductionStatusType)** [Type = bitmap];

ASPMining.CanonicalModel.Production.ProductionEvent

This table holds all the ProductionEvents of the production cycle in one table: Transit to Load, Waiting to Load, Loading, Transit to Dump, Waiting for Dump and Dumping

▼ Properties

- property **Duration** as %Double;

Duration of the Event in Seconds

- property **Equipment** as ASPMining.CanonicalModel.Equipment.Equipment;

Equipment asociated, must exist

- property **EquipmentExcav** as ASPMining.CanonicalModel.Equipment.Equipment;

Equipment Excav, must exist

- property **EquipmentTruck** as ASPMining.CanonicalModel.Equipment.Equipment;

Equipment Truck, must exist

- property **Id** as %BigInt;

Unique Id

- property **Location** as ASPMining.CanonicalModel.Mine.Location;

Location where event happens, should exist

- property **MeasuredTons** as %Double;

Measured Tons related to this event

- property **OperatorExcav** as ASPMining.CanonicalModel.Personnel.Operator;

Equipment Excav, must exist

- property **OperatorTruck** as ASPMining.CanonicalModel.Personnel.Operator;

Equipment Truck, must exist

- property **PreviousLocation** as ASPMining.CanonicalModel.Mine.Location;

Target Location where the equipment came from, only exists when this event is a moving ProductionEventType e.g: transitToDump, transitToLoad

- property **ProductionStatusType** as ProductionStatusType;

Production Status Type: loading, dump, waiting, etc.

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Shift where that event belongs to

- property **StartTime** as %DateTime;

Registration Time of the Event

▼ Indices

•index **(EquipmentIndex** on **Equipment)** [Type = bitmap];

•index **(IdIndex** on **Id)** [Unique];

•index **(LocationIndex** on **Location)** [Type = bitmap];

•index **(PreviousLocationIndex** on **PreviousLocation)** [Type = bitmap];

•index **(ShiftIndex** on **Shift)** [Type = bitmap];

•index **(StartTimeIndex** on **StartTime)**;

ASPMining.CanonicalModel.Production.ProductionEventSliced

▼ Parameters

- parameter **DSTIME** = "AUTO";

▼ Properties

- property **DumpLocation** as ASPMining.CanonicalModel.Mine.Location;

Location of origin of the cycle

- property **EquipmentExcav** as ASPMining.CanonicalModel.Equipment.Equipment;

Excavator Link

- property **EquipmentTruck** as ASPMining.CanonicalModel.Equipment.Equipment;

Truck Link

- property **EventEndDate** as %DateTime;

Real End date of the event

- property **EventStartDate** as %DateTime;

Event Start Date INSIDE this time slot

- property **LoadLocation** as ASPMining.CanonicalModel.Mine.Location;

Location of origin of the cycle

- property **MeasuredTons** as %Double;

Measured tons of the Dump (if available)

- property **PartialDuration** as %Double;

Calculated duration of the event

- property **ProductionStatusType** as ProductionStatusType;

Status Type Link

- property **Shift** as ASPMining.CanonicalModel.Planning.Shift;

Shift where this event happened

- property **SliceStartDate** as %DateTime;

Start date of the event

▼ Indices

- index **(PartialDurationIndex** on **PartialDuration)** [Type = bitslice];

- index **(SliceStartTimeIndex** on **SliceStartDate)** [Type = index];

- index **(ddbkeyIndex** on **ddbkey)** [Type = index];

ASPMining.CanonicalModel.Production.ProductionStatusType

Standarized Production Status. These status are 100% related to the production cycle and are independant of the Operation Status.

▼ Properties

- property **Description** as %String;

Description of the status, e.g: TransitToDumpSite, TransitToLoadSite, Loading, Dumping, WaitingForLoad, WaitingForDump. Very important for Production KPIs

- property **Type** as %Integer;

Type of the status, e.g: 1,2,3,4,5,6

▼ Methods

- classMétodo **CreateTypes()**

▼ Indices

- index **(TypeIndex** on **Type)** [Unique];