Database Gate User's Guide
javAPRSSrvr 4.3.3

Table of Contents

Section 1 - Introduction

DBGate extends net.ae5pl.aprssrvr.InternalClientRcv. Because of this, there can be multiple DBGates in one javAPRSSrvr instance, each making independent use of server-side filter, etc. DBGate takes packets from javAPRSSrvr and populates multiple SQL tables: APRSPosits, APRSTrack, APRSWx, APRSPackets, APRSObjects, APRSItems, APRSTelemetryDesc, and APRSTelemetryData.

Section 2 - Program Requirements and Description

DBGate is designed to run on any OS and Java VM supported by javAPRSSrvr.

DBGate is comprised of a number of classes which Java looks at as objects. The main class is net.ae5pl.dbgate.DBGate. This class is called at startup, sets parameters, and begins execution. Database updates are done in their own thread to ensure that javAPRSSrvr will not be delayed due to database constraints.

DBGate utilizes the generic JDBC capabilities of recent JVMs to facilitate populating SQL tables in a single database. While a basic definition of these tables is included here, most of the tables can be modified in any way desired if stored procedures are used to update them.

Section 3 - Configuration Properties

The configuration properties reside in properties files for each client adjunct, server adjunct, and port. The main properties file is called javaprssrvr.properties by default. You can use any text file for the main properties file if you pass the name into javAPRSSrvr as a command line parameter.

The property names are not case sensitive but the values can be. Defaults are shown below.

NOTE: UNLESS YOU REQUIRE A SETTING OTHER THAN THE DEFAULT, DO NOT INCLUDE ANY PARAMETERS WITH DEFAULT SETTINGS.

List parameters (L) may be defined on the property line or may be defined in a text file with the suffix .lst. If defined on the line, each entry is separated by a semicolon. If defined in a file, each entry is put on a separate line in the .lst file and the file name is the property value. Do not put blank lines in the file. For instance, this could be a definition for ListProperty (example only):

ListProperty=first.aprs.net:1313;second.aprs.net:1313

Or you could have the following 2 lines in a file named hubs.lst:

first.aprs.net:1313
second.aprs.net:1313

with ListProperty=hubs.lst

Properties preceded by a (M) are unchangeable and should not be included in your properties files. They are included in the descriptions below to indicate what common properties are available vs. those that have been forcibly overridden.

javAPRSSrvr Properties

Clients=

This must include the DBGate properties file.

DBGate General Properties

ClassPath=
(Deprecated)(L) Must be set to DBGate.jar.

Packages=
(Deprecated)(L) Must be set to the JDBC driver jar name (do not include .jar or any version number).

Class=
(Deprecated) Must be set to net.ae5pl.dbgate.DBGate.

ClientType=
`Set to db or SQL or database`

(M)StationCall=DATABASE
This is the callsign-SSID for DBGate. This is set to DATABASE and cannot be changed.

(M)Upstream=false

(M)History=false

(M)ReadOnly=true

(M)LocalOnly=false

ServerCommand=
This sets server adjunct command(s) for the client.

(M)FixedCommand=false
(true if ServerCommand set)

FullFeed=false
(defaults to true if ServerCommand not set)Indicates all packets except those received from this port should be sent to the client.

(M)LoginCommands=

(M)MessageHoldTime=-1

(M)LastHeardTime=-1

JDBC Driver Properties

DatabasePath=

This is the "path" to the database.
This will vary depending on where you put the tables and the driver that you use (e.g. jdbc:odbc:APRSDB). The format is defined by your JDBC vendor.

ConnProps=

This is a list parameter which defines the Properties for DriverManager.getConnection(String url, Properties info).
This is where you can define Properties for opening the connection to the database. Most often this will include user and password to give the driver the database login information. The list is set in pairs. For instance,
ConnProps=user;testing;password;1234
Valid connection properties are defined by your JDBC vendor.
Note that this can also be a .properties file name that is loaded directly.

PasswordPropName=password

This is the property name to be set in the connection Properties using the decrypted DBEncPassword value.

DBEncPassword=

This is the encrypted/encoded database password to be decrypted and set in the connection Properties property defined by PasswordPropName. Use encpwd application to generate the string for this property.

LogAll=true

If set to true, all packets will be logged to the APRSPackets table.

MultipleConnections=false

If set to true, a connection for each database statement will be made to the database server. Otherwise, only one connection will be used.
This is for driver and database optimization.

ConvertCallsign=false

If set to true, callsigns and/or icons containing apostrophe(s) will be modified for JDBC drivers that do not check for apostrophes in the string.
This is a workaround for non-compliant JDBC drivers. Any apostrophe found in the callsign or icon will be converted to a 2 apostrophes so the SQL statements will succeed.

ConvertPacket=false

If set to true, packets containing apostrophe(s) will be modified for JDBC drivers which convert setBytes() to a string and then do not check for apostrophes in the string.
This is a workaround for non-compliant JDBC drivers. This should never be set to true, but is here for diagnostic purposes.

MaxPacketSize=512

Maximum size of Packet column in APRSPosits table.

SeparateObjectsItems=true

Store Objects in APRSObjects and Items in APRSItems.
APRSObjects and APRSItems are the same schema as APRSPosits allowing for simple UNIONs and JOINs to be used for display and information delivery.
If UsePositStoredProc is also set to true, the PacketType argument will be set to 20 for Items to simplify stored procedure functionality. It will still be stored as 2 in APRSPackets.

DeleteKilled=false

Delete “killed” object or item from the APRSObjects/APRSItems tables.

This setting, if set to true, will delete an object/item row from the table if the object/item is killed (underscore following object/item name). If UsePositStoredProc=true, PacketType will be 10 greater than for live Objects/Items (12 for Objects, 30 for Items).

LogTelemetry=false

Populate the APRSTelemetryDesc and APRSTelemetryData tables.

Decimal=DECIMAL

This is the type definition for decimal columns in APRSWx. Valid values are DECIMAL, NUMERIC, and DOUBLE (must be all upper case). If UseWxStoredProc=true, this property is always used.

If UseWxStoredProc=false and this property does not exist, the database metadata will be accessed to determine the type for HourRain. If found, that type will be used, otherwise this property will be used. A log entry will be made indicating what the value is set to.

This allows for optimizations of the weather INSERT statement where decimal columns may be NULL. This parameter explicitly tells the JDBC driver to use a specific format when setting the column to NULL and whether to use setBigDecimal (DECIMAL and NUMERIC) or setDouble (DOUBLE) when setting a value. Note that while many JDBCs support DECIMAL, they may consider a null DECIMAL to be a different format causing multiple prepared statements to be generated for most weather INSERTs. For instance, the JTDS JDBC driver for MS SQL performs better with Decimal=DOUBLE. Most modern JDBC drivers properly support decimal data types.

TinyInt=TINYINT

This defines the column type for Humidity in APRSWx and UpdateWx, and PacketType in APRSPackets. It may be TINYINT or SMALLINT (must be all upper case). If the database supports tinyint, use it for those columns and parameter. If not (Postgres is an example that does not support tinyint), use smallint and set this property to SMALLINT.
If UseWxStoredProc=true, this property will always be used.

If UseWxStoredProc=false and this property does not exist, the database metadata will be accessed to determine the type for Humidity. If found, that type will be used, otherwise this property will be used. A log entry will be made indicating what the value is set to.

Update/Insert/Stored Procedure Parameters

MinTrackDist=0.02

Minimum distance in degrees (1 degree = 60 nm) to move before a track point is made.
This can be set to 0.0.

MinTrackTime=30

Minimum time in seconds to elapse before a track will be recorded.
This can be set to 0. This also affects how fast APRSPosits is updated.

MinWxTime=270

Minimum time in seconds to elapse before a weather report will be recorded.
This can be set to 0.

UsePositStoredProc=false

Use a stored procedure to update the posit and track tables.
MinTrackDist and MinTrackTime are ignored if this is true and are expected to be handled within the stored procedure.
See the chapter 7 for the stored procedure format. See SeparateObjectsItems for more information if set to true.

UseWxStoredProc=false

Use a stored procedure to update the weather table.
MinWxTime is ignored if this is true and is expected to be handled within the stored procedure. See the chapter 7 for the stored procedure format.

UsePcktStoredProc=false

Use a stored procedure to update the packet tables.
See the chapter 7 for the stored procedure format.

Section 4 - Recommended Configurations

Make sure your JDBCDriver name and DatabasePath are set correctly for your configuration. Set ConnProps if required by your JDBC driver. The JDBC driver jar file(s) must be included in ClassPath property with DBGate.jar if started with the -jar switch.

Section 5 - Installation Instructions

Run the scripts to populate your database with the proper tables. They may require modification to run properly on your database. Most often, datetime will need to be changed to timestamp and varbinary to another database specific data type for non-Microsoft SQL databases.

DBGate is in a separate jar file (DBGate.jar) from javAPRSSrvr.

Section 6 – XML Status Page

General XML


```
<clientrcv>
<time>
<connect utc="1341166062722"/>
<lastlinein utc="1341166062722"/>
</time>
<upstream>
false
</upstream>
<readonly>
true
</readonly>
<login>
<callssid verified="false">
jFindu-DB
</callssid>
<software version="4.0.0">
DBGate
</software>
</login>
<rcvdfrom bytes="0" lines="0" packets="0">
<udp bytes="0" lines="0" packets="0"/>
</rcvdfrom>
<clientxmt>
<upstream>
false
</upstream>
<sentto bytes="18119526" lines="179775" packets="179775">
<lastlinems>
1341242778113
</lastlinems>
<udp bytes="0" lines="0" packets="0"/>
</sentto>
<xmtqueue depth="0" depthms="0"/>
<adjuncts>
<filteraprsclient>
<filter>
<![CDATA[filter d/DSTAR s//&a/D]]>
</filter>
</filteraprsclient>
</adjuncts>
</clientxmt>
</clientrcv>
```

Detail XML


```
<clientxmt>
<class name="DBGateXmt">
<package name="net.ae5pl.dbgate" revision="b01" title="APRS JDBC Client" version="4.0.0"/>
</class>
<upstream>
false
</upstream>
<sentto bytes="18121110" lines="179791" packets="179791">
<lastlinems>
1341242785807
</lastlinems>
<udp bytes="0" lines="0" packets="0"/>
</sentto>
<xmtqueue depth="0" depthms="0"/>
<adjuncts>
<filteraprsclient>
<filter>
<![CDATA[filter d/DSTAR s//&a/D]]>
</filter>
<class name="FilterClient">
<package name="net.ae5pl.aprsfilter" revision="b01" title="APRS Server Filter Adjunct"
version="4.0.0"/>
</class>
<passfilters>
<filterset>
<digi>
DSTAR
</digi>
<symbols>
<secondary>
<![CDATA[&a]]>
</secondary>
<overlays>
D
</overlays>
</symbols>
</filterset>
</passfilters>
<rejfilters>
<filterset/>
</rejfilters>
</filteraprsclient>
</adjuncts>
</clientxmt>
<jdbcdriver compliant="false" version="1.2">
net.sourceforge.jtds.jdbc.Driver
</jdbcdriver>
<rows inserted="179791" updated="179788"/>
<nonpositwxpackets>
```

0
</nonpositwxpackets>
</clientrcv>

Section 7 – Table and Stored Procedure Definitions

These definitions are derived from the Microsoft SQL Server 2000 setup. You may need to alter these, depending on your database. The tables must consist of at least these columns. Order of the columns is not important. Names and case of the names of the columns are important. Other columns may be added as you desire.

The CallsignSSID column in all tables must be collated, at a minimum, case-sensitive. Binary collation is preferred.

APRSPosits

Table

```
CREATE TABLE [dbo].[APRSPosits] (
        [CallsignSSID] [varchar] (9) NOT NULL , (Trimmed Callsign-SSID or object/item name)
        [ReportTime] [datetime] NOT NULL , (Updated using the java.sql.ResultSet.updateTimestamp())
        [Latitude] [float] NOT NULL , (Decimal degrees, south is negative)
        [Longitude] [float] NOT NULL , (Decimal degrees, west is negative)
        [Course] [smallint] NULL , (0-359, NULL indicates speed is invalid too)
        [Speed] [int] NULL , (Miles per hour)
        [Altitude] [int] NULL , (Feet)
        [Packet] [varbinary] (512) NULL , (May contain any byte, 0-255)
        [Icon] [char] (2) NULL (Symbol table, symbol code)
) ON [PRIMARY]
```

Indexes

```
ALTER TABLE [dbo].[APRSPosits] ADD
        CONSTRAINT [PK_APRSPosits] PRIMARY KEY  NONCLUSTERED
        (
                [CallsignSSID]
        ) ON [PRIMARY]
```
(For lookups by callsign)

```
CREATE  INDEX [IX_APRSPosits] ON [dbo].[APRSPosits]([ReportTime]) ON [PRIMARY]
```
(For deleting aged reports)

APRSTrack

This table does not include the latest position report which is found in APRSPosits.

Table

```
CREATE TABLE [dbo].[APRSTrack] (
        [CallsignSSID] [varchar] (9) NOT NULL , (Trimmed Callsign-SSID)
        [ReportTime] [datetime] NOT NULL , (Set using the java.sql.PreparedStatement.setTimestamp())
        [Latitude] [float] NOT NULL , (Decimal degrees, south is negative)
        [Longitude] [float] NOT NULL , (Decimal degrees, west is negative)
        [Icon] [char] (2) NULL , (Symbol table, symbol code)
        [Course] [smallint] NULL , (0-359, NULL indicates speed is invalid too)
        [Speed] [int] NULL , (Miles per hour)
        [Altitude] [int] NULL (Feet)
) ON [PRIMARY]
```

Indexes

```
CREATE  INDEX [IX_APRSTrack_RT] ON [dbo].[APRSTrack]([ReportTime]) ON [PRIMARY]
(For deleting aged reports)

CREATE  INDEX [IX_APRSTrack] ON [dbo].[APRSTrack]([CallsignSSID], [ReportTime]) ON [PRIMARY]
(For easier lookups by callsign from other applications)
```

APRSWx

Table

```
CREATE TABLE [dbo].[APRSWx] (
        [CallsignSSID] [varchar] (9) NOT NULL , (Trimmed Callsign-SSID)
        [ReportTime] [datetime] NOT NULL , (Set using the java.sql.PreparedStatement.setTimestamp())
        [WindDir] [smallint] NULL , (0-359)
        [WindSpeed] [smallint] NULL , (Miles per hour)
        [GustSpeed] [smallint] NULL , (Miles per hour)
        [Temperature] [smallint] NULL , (Fahrenheit)
        [HourRain] [decimal](4, 2) NULL , (Inches)
        [DayRain] [decimal](6, 2) NULL , (Inches)
        [MidnightRain] [decimal](6, 2) NULL , (Inches)
        [Humidity] [tinyint] NULL , (Percent, setByte() for dbTinyInt=true, setShort() otherwise)
        [BarPressure] [decimal](5, 1) NULL (Millibars)
) ON [PRIMARY]
```

Indexes

```
CREATE  INDEX [IX_APRSWx_RT] ON [dbo].[APRSWx]([ReportTime]) ON [PRIMARY]
```
(For deleting aged reports)

```
CREATE  INDEX [IX_APRSWx] ON [dbo].[APRSWx]([CallsignSSID], [ReportTime]) ON [PRIMARY]
```
(For easier lookups by callsign from other applications)

APRSPackets

Table

```
CREATE TABLE [dbo].[APRSPackets] (
        [CallsignSSID] [varchar] (9) NOT NULL , (FromCall ToCall Length CRC32)
        [ReportTime] [datetime] NOT NULL , (Set using the java.sql.PreparedStatement.setTimestamp())
        [PacketType] [tinyint] NOT NULL , (0-3 (see below); setByte() dbTinyInt=true, setShort() otherwise)
        [IsWx] [bit] NOT NULL , (Weather packet, setBoolean() used)
        [Packet] [varbinary] (512) NOT NULL  (May contain any byte, 0-255)
) ON [PRIMARY]
```

PacketType Enumeration:

- 0 = Not identified
- 1 = Message/Bulletin
- 2 = Object/Item
- 3 = Position

Indexes

```
CREATE  INDEX [IX_APRSPackets_RT] ON [dbo].[APRSPackets]([ReportTime]) ON [PRIMARY]
```

(For deleting aged reports)

```
CREATE  INDEX [IX_APRSPackets] ON [dbo].[APRSPackets]([CallsignSSID], [ReportTime]) ON [PRIMARY]
```

(For easier lookups by callsign from other applications)

APRSObjects

Table

```
CREATE TABLE [dbo].[APRSObjects] (
        [CallsignSSID] [varchar] (9) NOT NULL , (Trimmed Callsign-SSID or object/item name)
        [ReportTime] [datetime] NOT NULL , (Updated using the java.sql.ResultSet.updateTimestamp())
        [Latitude] [float] NOT NULL , (Decimal degrees, south is negative)
        [Longitude] [float] NOT NULL , (Decimal degrees, west is negative)
        [Course] [smallint] NULL , (0-359, NULL indicates speed is invalid too)
        [Speed] [int] NULL , (Miles per hour)
        [Altitude] [int] NULL , (Feet)
        [Packet] [varbinary] (512) NULL , (May contain any byte, 0-255)
        [Icon] [char] (2) NULL (Symbol table, symbol code)
) ON [PRIMARY]
```

Indexes

```
ALTER TABLE [dbo].[APRSObjects] ADD
        CONSTRAINT [PK_APRSObjects] PRIMARY KEY  NONCLUSTERED
        (
                [CallsignSSID]
        ) ON [PRIMARY]
```

(For lookups by callsign)

```
CREATE  INDEX [IX_APRSObjects] ON [dbo].[APRSObjects]([ReportTime]) ON [PRIMARY]
```

(For deleting aged reports)

APRSItems

Table

```
CREATE TABLE [dbo].[APRSItems] (
        [CallsignSSID] [varchar] (9) NOT NULL , (Trimmed Callsign-SSID or object/item name)
        [ReportTime] [datetime] NOT NULL , (Updated using the java.sql.ResultSet.updateTimestamp())
        [Latitude] [float] NOT NULL , (Decimal degrees, south is negative)
        [Longitude] [float] NOT NULL , (Decimal degrees, west is negative)
        [Course] [smallint] NULL , (0-359, NULL indicates speed is invalid too)
        [Speed] [int] NULL , (Miles per hour)
        [Altitude] [int] NULL , (Feet)
        [Packet] [varbinary] (512) NULL , (May contain any byte, 0-255)
        [Icon] [char] (2) NULL (Symbol table, symbol code)
) ON [PRIMARY]
```

Indexes

```
ALTER TABLE [dbo].[APRSItems] ADD
        CONSTRAINT [PK_APRSItems] PRIMARY KEY  NONCLUSTERED
        (
                [CallsignSSID]
        ) ON [PRIMARY]
```

(For lookups by callsign)

```
CREATE  INDEX [IX_APRSItems] ON [dbo].[APRSItems]([ReportTime]) ON [PRIMARY]
```

(For deleting aged reports)

APRSTelemetryDesc

Table

```
CREATE TABLE [dbo].[APRSTelemetryDesc] (
        [CallsignSSID] [varchar] (9) NOT NULL , (Trimmed Callsign-SSID)
        [UpdateTime] [datetime] NOT NULL , (Updated using the java.sql.ResultSet.updateTimestamp())
        [PARMA1] [varchar](7) NULL,
        [PARMA2] [varchar](6) NULL,
        [PARMA3] [varchar](5) NULL,
        [PARMA4] [varchar](5) NULL,
        [PARMA5] [varchar](4) NULL,
        [PARMB1] [varchar](5) NULL,
        [PARMB2] [varchar](4) NULL,
        [PARMB3] [varchar](3) NULL,
        [PARMB4] [varchar](3) NULL,
        [PARMB5] [varchar](3) NULL,
        [PARMB6] [varchar](2) NULL,
        [PARMB7] [varchar](2) NULL,
        [PARMB8] [varchar](2) NULL,
        [UNITA1] [varchar](7) NULL,
        [UNITA2] [varchar](6) NULL,
        [UNITA3] [varchar](5) NULL,
        [UNITA4] [varchar](5) NULL,
        [UNITA5] [varchar](4) NULL,
        [UNITB1] [varchar](5) NULL,
        [UNITB2] [varchar](4) NULL,
        [UNITB3] [varchar](3) NULL,
        [UNITB4] [varchar](3) NULL,
        [UNITB5] [varchar](3) NULL,
        [UNITB6] [varchar](2) NULL,
        [UNITB7] [varchar](2) NULL,
        [UNITB8] [varchar](2) NULL,
        [EQNA1a] [float] NULL,
        [EQNA1b] [float] NULL,
        [EQNA1c] [float] NULL,
        [EQNA2a] [float] NULL,
        [EQNA2b] [float] NULL,
        [EQNA2c] [float] NULL,
        [EQNA3a] [float] NULL,
        [EQNA3b] [float] NULL,
        [EQNA3c] [float] NULL,
        [EQNA4a] [float] NULL,
        [EQNA4b] [float] NULL,
        [EQNA4c] [float] NULL,
        [EQNA5a] [float] NULL,
        [EQNA5b] [float] NULL,
        [EQNA5c] [float] NULL,
        [B1] [bit] NOT NULL,
        [B2] [bit] NOT NULL,
        [B3] [bit] NOT NULL,
        [B4] [bit] NOT NULL,
        [B5] [bit] NOT NULL,
        [B6] [bit] NOT NULL,
        [B7] [bit] NOT NULL,
        [B8] [bit] NOT NULL,
        [Project] [varchar](23) NULL,
) ON [PRIMARY]
```

Indexes

```
ALTER TABLE [dbo].[APRSTelemetryDesc] ADD
	CONSTRAINT [PK_APRSTelemetryDesc] PRIMARY KEY  NONCLUSTERED
	(
		[CallsignSSID]
	) ON [PRIMARY]
```

(For lookups by callsign)

```
CREATE  INDEX [IX_APRSTelemetryDesc] ON [dbo].[APRSTelemetryDesc]([UpdateTime]) ON [PRIMARY]
```

(For deleting aged reports)

APRSTelemetryData

Table

```
CREATE TABLE [dbo].[APRSTelemetryData] (
        [CallsignSSID] [varchar] (9) NOT NULL , (FromCall ToCall Length CRC32)
        [ReportTime] [datetime] NOT NULL , (Set using the java.sql.PreparedStatement.setTimestamp())
        [A1] [smallint] NULL,
        [A2] [smallint] NULL,
        [A3] [smallint] NULL,
        [A4] [smallint] NULL,
        [A5] [smallint] NULL,
        [B1] [bit] NOT NULL,
        [B2] [bit] NOT NULL,
        [B3] [bit] NOT NULL,
        [B4] [bit] NOT NULL,
        [B5] [bit] NOT NULL,
        [B6] [bit] NOT NULL,
        [B7] [bit] NOT NULL,
        [B8] [bit] NOT NULL,
        [Comment] [varchar](128) NULL
) ON [PRIMARY]
```

Indexes

```
CREATE  INDEX [IX_APRSTelemetryData_RT] ON [dbo].[APRSTelemetryData]([ReportTime]) ON [PRIMARY]
```
(For deleting aged reports)

```
CREATE  INDEX [IX_APRSTelemetryData] ON [dbo].[APRSTelemetryData]([CallsignSSID], [ReportTime]) ON
[PRIMARY]
```
(For easier lookups by callsign from other applications)

UpdatePosit

Stored Procedure

(Parameter order is important)

```
CREATE PROCEDURE dbo.UpdatePosit
        @CallsignSSID varchar(9), (Trimmed Callsign-SSID or object/item name)
        @ReportTime datetime, (Set using the java.sql.PreparedStatement.setTimestamp())
        @Latitude float, (Decimal degrees, south is negative)
        @Longitude float, (Decimal degrees, west is negative)
        @Course smallint, (0-359, NULL indicates speed is invalid too)
        @Speed int, (Miles per hour, may be NULL)
        @Altitude int, (Feet, may be NULL)
        @Packet varbinary(512), (May contain any byte, 0-255)
        @Icon char(2), (Symbol table, symbol code)
        @PacketType int (Same enumeration as APRSPackets, recommend only track PacketType=3)
AS
BEGIN
---Put your code here
END;
```

UpdateWX

Stored Procedure

(Parameter order is important)
(All parameters except @CallsignSSID and @ReportTime may be NULL)

```
CREATE PROCEDURE dbo.UpdateWX
        @CallsignSSID varchar(9), (Trimmed Callsign-SSID)
        @ReportTime datetime, (Set using the java.sql.PreparedStatement.setTimestamp())
        @WindDir smallint, (0-359)
        @WindSpeed smallint, (Miles per hour)
        @GustSpeed smallint, (Miles per hour)
        @Temperature smallint, (Fahrenheit)
        @HourRain decimal(4, 2), (Inches)
        @DayRain decimal(6, 2), (Inches)
        @MidnightRain decimal(6, 2), (Inches)
        @Humidity tinyint, (Percent, setByte() for dbTinyInt=true, setShort() otherwise)
        @BarPressure decimal(5, 1) (Millibars)
AS
BEGIN
--- Put your code here
END;
```

UpdatePackets

Stored Procedure

(Parameter order is important)

```
CREATE PROCEDURE dbo.UpdatePackets
        @CallsignSSID varchar(9), (FromCall ToCall Length CRC32)
        @ReportTime datetime, (Set using the java.sql.PreparedStatement.setTimestamp())
        @PacketType tinyint, (0-3 (see below); setByte() dbTinyInt=true, setShort() otherwise)
        @IsWx bit, (Weather packet, setBoolean() used)
        @Packet varbinary(512) (May contain any byte, 0-255)
AS
BEGIN
---Put your code here
END;
```

PacketType Enumeration:

- 0 = Not identified
- 1 = Message/Bulletin
- 2 = Object/Item
- 3 = Position