

# Vanguard Nexus

Nexus – Technical Specification and Database Schema  
Overview

May 2021

0	Internal Review	4/02/19	BS	GB
1	First Release	5/02/19	BS	GB
2	Made some small changes	5/12/20	BS	GB
3	Added Network, DB load section. Minor other changes	11/2/21	BS	
4	Updated section 3.1	15/05/21		

*Copyright of this document or any part of this document remains with Vanguard Pty Ltd and cannot be used, transferred or reproduced in any manner or form without prior written consent from Vanguard Pty Ltd.*

# Table of Contents

<b>1</b>	<b>Introduction .....</b>	<b>3</b>
<b>2</b>	<b>System Requirements .....</b>	<b>4</b>
	2.1 Service and User Interface .....	8
	2.2 Database Server .....	8
<b>3</b>	<b>Configuration Files .....</b>	<b>10</b>
<b>4</b>	<b>Reporting .....</b>	<b>14</b>
<b>5</b>	<b>Database Schema .....</b>	<b>15</b>
	5.1 Tables .....	15
	5.1.1 Blasting.....	15
	5.1.2 Calculation .....	17
	5.1.3 Logging .....	19
	5.1.4 Scenarios .....	20
	5.1.5 Scheduling.....	22
	5.1.6 Settings .....	28
	5.1.7 Simulation .....	29





## **1** Introduction

Nexus is a system that allows rapid evaluation of environmental noise impacts of mobile and fixed plant equipment in a mining environment. It uses weather data, equipment location information, topography, and information about nearby sensitive receivers to predict whether noise impacts are likely to exceed regulatory limits, and provides the user with information to quickly identify which activities or equipment are likely to cause problems, and correct any issues.

The user interacts with Nexus through a user interface (UI). Detailed information about how to use the UI is included in the User Manual. This document focuses on architecture and configuration of the Nexus system.

Nexus consists of two base components – Nexus Scheduler and Nexus Execute. Nexus Scheduler is used for planning future operations based on mine plans and forecast weather. Nexus Execute is used for the management of noise impacts during operations.

### **1.1** Architecture

Nexus allows multiple users or automated data sources to simultaneously update its database. To achieve this, it relies on a central service that is responsible for receiving new data and informing connected clients of any changes that have been made.

A high level overview of the system architecture is shown in Illustration 1.

Nexus consists of three main components:

- The service, responsible for automated data input, keeping all clients up-to-date, and performing noise calculations.
- The client, or user interface, responsible for providing a user interface that allows the user to view and modify the contents of the database.
- The database, responsible for storing all persistent information relevant to the application.

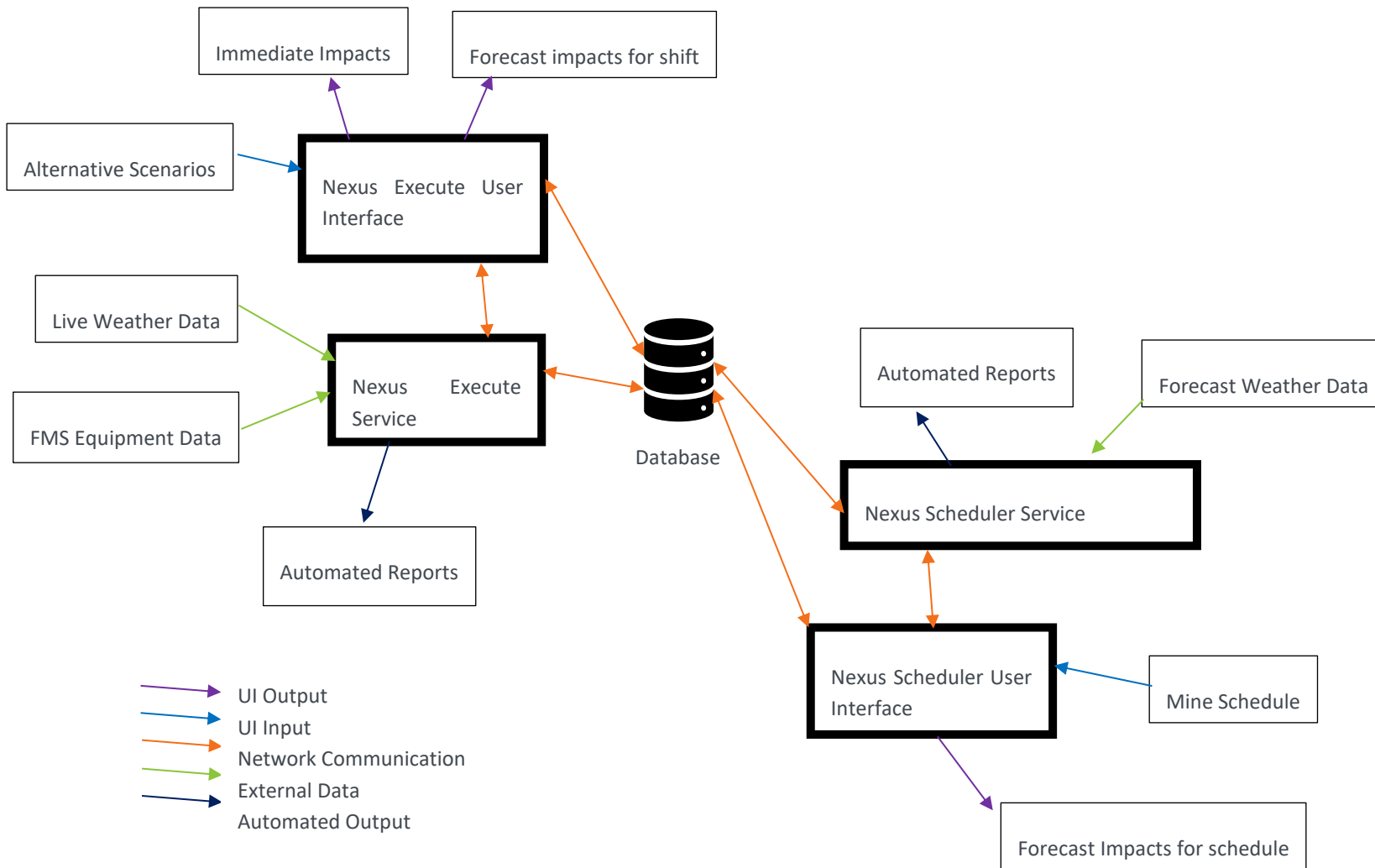


Figure 1 Nexus Architecture Overview

## 2 Network, Firewall, and Virtual Machine configuration

Each Nexus Service listens for TCP connections on a single, configurable port. The default ports are 4533 for Nexus Scheduler and 4534 for Nexus Service. See chapter 4 for configuration details.

When the Nexus Client is started, it opens a single long-lived connection to the Nexus Service. Various packets are exchanged on this connection using a proprietary binary protocol. The connection is not encrypted.

All Nexus Services and Clients must also be able to connect to the SQL Server instance that hosts the Nexus Database. Database connections are authenticated using either Windows Authentication or SQL Server username and password. Database connections may be short or long lived.

The majority of network traffic is between the clients and the SQL Server. Communications between clients and the Nexus service are mostly small packets used to notify that the database has changed.

For simplicity, it is not recommended to run the Nexus Services on the same virtual machine as other business software unless there is a compelling reason to do so.



## 2.1 Example Configurations

### 2.1.1 Single (Virtual) Machine for all services and SQL Server, Clients on separate machines

In this configuration, the Nexus Execute and Scheduler services run on a single machine or virtual machine (VM), and the SQL Server database runs on the same machine.

Each user interface client application will run on its own machine or VM and connect to the database and Nexus service.

This configuration has the advantage of requiring only one virtual machine for the long running back-end portion of the architecture. It is only appropriate when the SQL Server instance will only host Nexus data; this may necessitate the purchase of SQL Server licenses solely for Nexus.

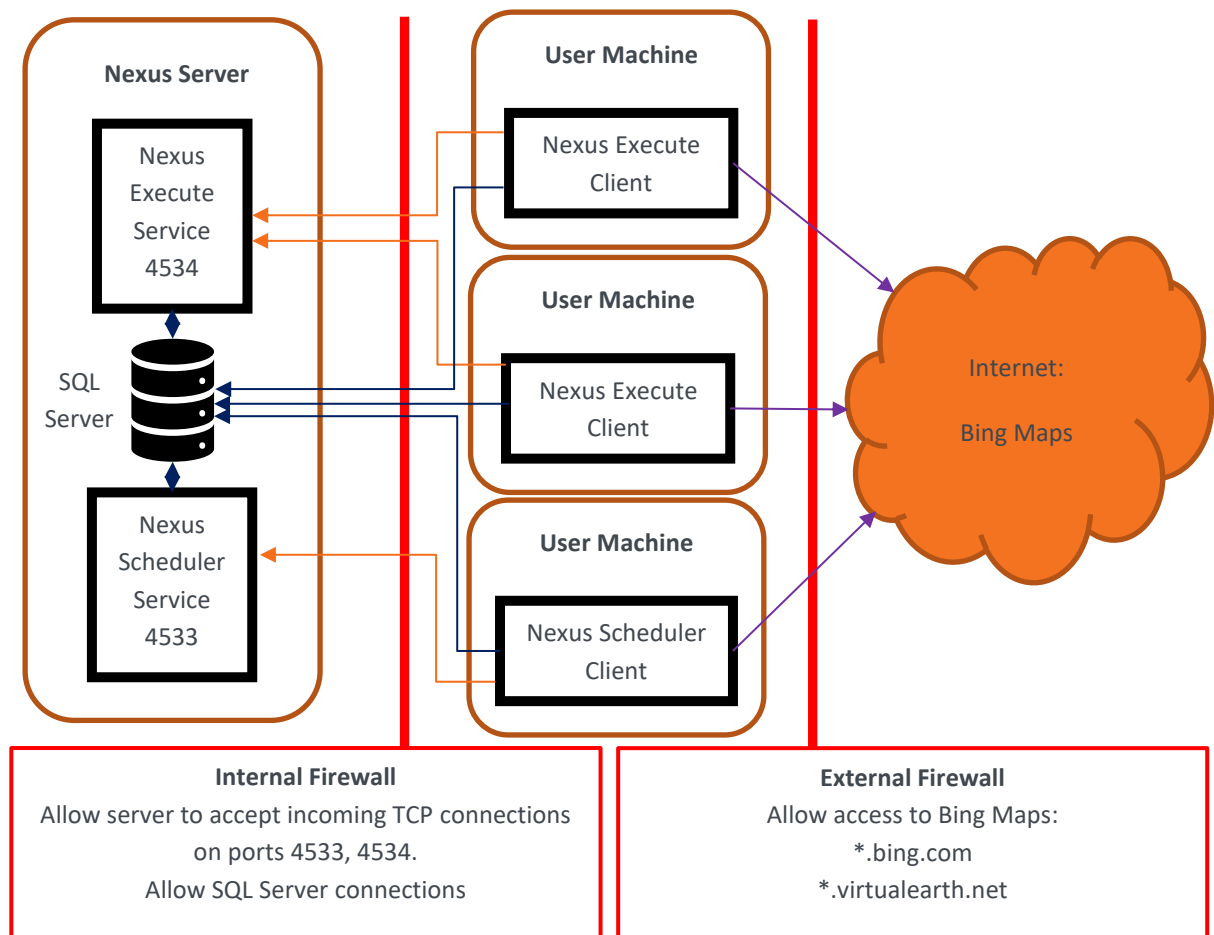


Figure 2: Network connections with single Nexus Server

### 2.1.2 Nexus Services on one machine, SQL Server on different instance. Clients on separate machines

In this configuration, the Nexus Execute and Scheduler services run on a single machine or VM. The SQL Server instance runs on a separate machine.

As before, each user interface client application runs on its own machine and connects to both the database and the Nexus service.

This configuration is appropriate when the SQL Server instance will also host other databases unrelated to Nexus, or when business rules require database servers to be located on specific hardware.

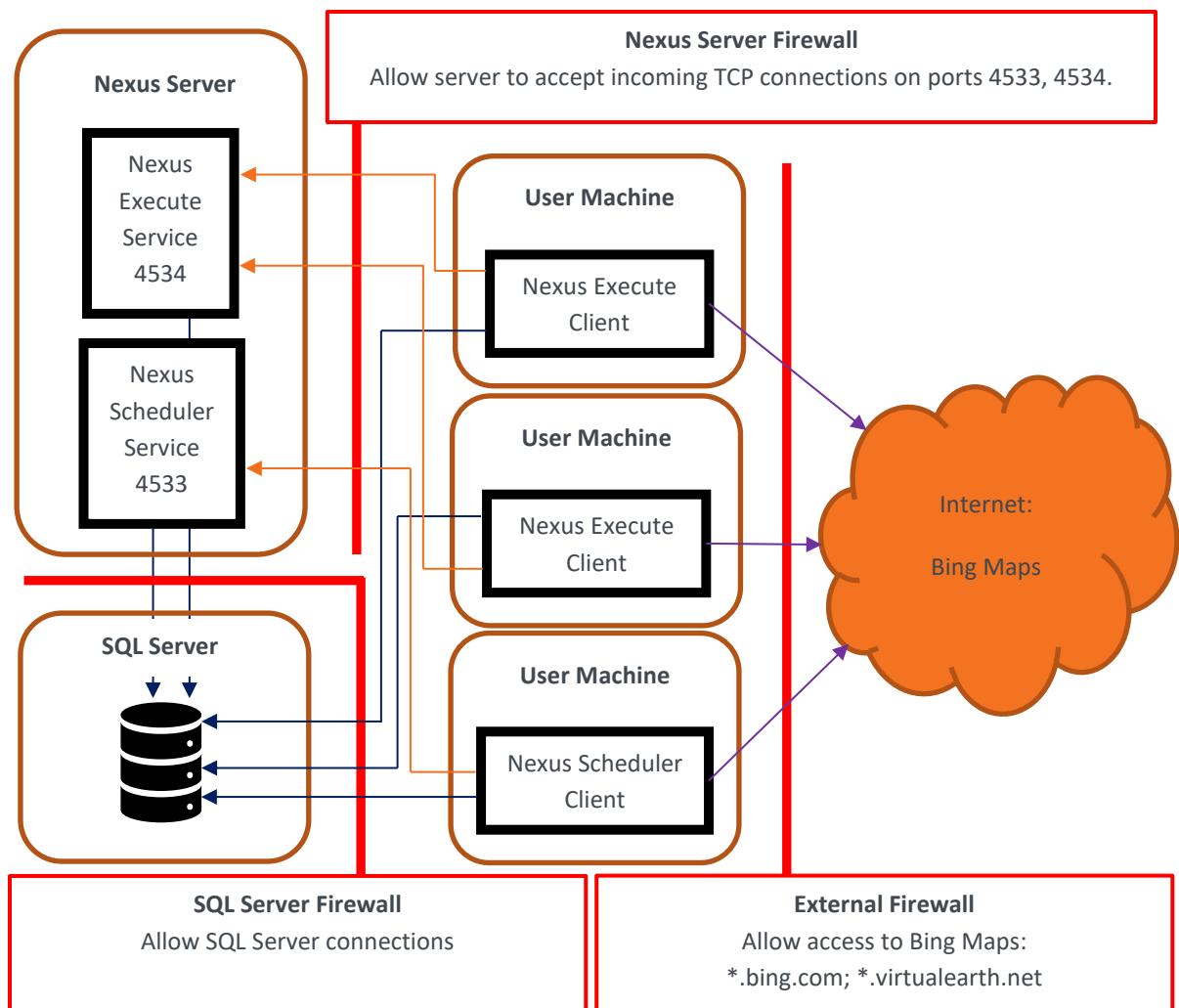


Figure 3 Network connections with Nexus Services and SQL Server running on different machines

### 2.1.3 Other Options

There is no need to run both Nexus Scheduler and Execute Services on the same machine. Given that the different user groups for each service may reside on different networks, it may be advantageous to run the services on separate machines.

## 3 System Requirements

### 3.1 Service and User Interface

Nexus Service and User Interface have mostly the same system requirements, with the exception that the service does not require display capabilities. Nexus Execute and Scheduler have the same system requirements.

#### 3.1.1 Nexus User Interface

- CPU Passmark: 10000 (i.e. Intel i5 or i7)
- RAM: 16 GB (preferably 32 GB) or more
- Network: 100 Mbps
- HDD: 100 MB
- Display: 1920 x 1080
- Operating System: 64bit Windows 8, 8.1, or Windows 10.
- dotNET Framework 4.6.1
- Internet connection required for maps

#### 3.1.2 Nexus Service

- CPU Passmark: 10000 (i.e. Intel i5 or i7)
- RAM: 16 GB or more
- Network: 100 Mbps
- HDD: 100 MB
- Operating System: 64bit Windows 8, 8.1, or Windows 10.
- dotNET Framework 4.6.1

### 3.2 Database Server

- Microsoft SQL Server Standard<sup>1</sup> 2016 or newer.
- Network latency less than 10 ms<sup>2</sup>
- Space requirements are dependent on model complexity. Typical requirement is on the order of 100 GB.
- Memory: 16 GB minimum, 32 GB or more recommended
- Solid state storage is required.

#### 3.2.1 Database Load and Shared Servers

Users may want to locate the Nexus database on an SQL Server instance that also hosts databases for other business needs. The following information is provided to assist in the decision whether the server will be able to handle the load from Nexus while providing acceptable performance serving other databases.

Typical Nexus database load is read-heavy, write-light. Furthermore, Nexus implements a client-side cache so that data that rarely changes is stored in memory. As a result, typical operation is that Nexus may

---

<sup>1</sup> Microsoft SQL Server Express may be used for databases smaller than 10GB.

<sup>2</sup> With higher latency, the system will continue to function but with degraded user experience.

occasionally burst read a large amount of modelling data – on the order of tens of millions of records, or tens of gigabytes – then place much lighter loads on the database, usually writing and reading some tens of records as the equipment schedule changes. The large burst reads are performed on client start-up, and when a new schedule is loaded – expect once or twice per day for long running client applications.

As mining progresses, new modelling data may need to be loaded into the Nexus database, either because of changed topography or moving into new areas. This involves a large write operation on the order of the size of the database, which can often take several hours on a dedicated server. If using a shared database server, plans should be in place to ensure that the server will continue to adequately serve other applications, for example through the Resource Governor.

## 4 Configuration Files

Nexus stores configuration information in ConfigDb files. These can be edited using the ConfigDb Editor, supplied with the application

### 4.1 Service Configuration

Service configuration files are stored in the same folder as the executable. All settings are stored in the 'LocalMachine' section of the configDb.

#### 4.1.1 appSettings

The appSettings key contains general settings. These are:

Setting	Description	Default Value
CachedResultsPurgeDays	Cached results older than this many days are periodically purged from the database	700
MessagingPort	The network port on which the service listens for incoming connections	Scheduler: 4533 Execute: 4534
ReportsFolder	The folder in which to output the received levels report	C:\Vanguard Nexus\Reports

#### 4.1.2 connectionStrings

The connectionStrings key contains connection strings that the service uses to access the database.

We recommend a connection string format similar to the following:

```
server=<SERVER NAME>; database=<DATABASE NAME>; Connect Timeout=10;
MultipleActiveResultSets=True; App=NexusSvc;
```

Where <SERVER NAME> physically identifies the database server, and <DATABASE NAME> identifies the Nexus database provided by Talis.

If using active directory authentication, add the following: `Integrated Security=true;`

If using SQL Server authentication, add the following: `User Id=<USERNAME>; Password=<PASSWORD>;`

##### 4.1.2.1 NexusScheduler

This specifies the main database that the service will connect to.

#### 4.1.3 NoiseLoggerMonitors

Each subkey within the NoiseLoggerMonitors key specifies a particular directory to watch for new noise logger data. The entries within each subkey are as follows:

Setting	Description	Default Value
DirectoryToScan	Specifies which directory to scan for new logger data	User entered

Setting	Description	Default Value
Filter	A file filter to specify that only particular file extensions will be watched	*.csv
HasColumnHeaders	Specifies whether the imported files have column headers	True
DateColumnId	Specifies the date column, if date and time are in different columns	<i>Not Present</i>
TimeColumnId	Specifies the time column	Start Time
LevelColumnId	Specifies the level column	LAeq
ColumnDelimiter	Specifies the column delimiter	,
HasSpectrum	Specifies whether the data includes spectral information	false
DateCulture	Specifies the date culture to use when parsing dates. If not specified, Nexus will use the system date culture.	<i>Not Present</i>
SpectrumWeighting	If spectral data is present, specifies the weighting that has been applied to the spectrum	Z
SpectrumIdentifiers	Up to individual entries, with a format of ###_Hz_id or ###_kHz_id, each of which specifies a column that contains a particular spectral band.	<i>Note Present</i>

Additionally, each monitor must also specify a *Source* subkey, which must contain the following entries:

Setting	Description	Default Value
Name	A name to identify the data source	<i>User entered</i>
Description	A description of the data source	<i>User entered</i>
ReceiverName	Identifies which receiver is associated with the noise monitor	<i>User entered</i>
MeasurementType	Specifies the type of the measurement. Possible values are: Calculated, PermanentLiveMonitor, TemporaryLiveMonitor, OfflineMonitor , AttendedMeasurement	<i>User entered</i>
RecordDuration	The duration of each individual record of the noise logger source, or the logging interval of the source. Specified in HH:MM:SS	<i>User entered</i>

#### 4.1.4 ReceiverReportSettings

This key contains configuration for the automated receiver report.

Setting	Description	Default Value
ReportFrequency	The interval between consecutive records in the report, in minutes unless changed by <i>ReportFrequencyIsMinutes</i>	15
ReportFrequencyIsMinutes	If <i>true</i> (default), then the report frequency is in minutes. If <i>false</i> , then the report frequency is in seconds	true

#### 4.1.5 WeatherMonitors

Each subkey within this key specifies a particular directory to watch for new weather data. The settings possible in subkeys are:

Setting	Description	Default Value
DirectoryToScan	The directory to scan for new weather data	C:\Vanguard Nexus\Weather Data
Filter	A file filter to specify that only particular file extensions will be watched	*.csv
HasColumnHeaders	Specifies whether the imported files have column headers	True
DateColumn	Specifies the date column, if date and time are in different columns	<i>Not Present</i>
TimeColumn	Specifies the time column	Start Time
DateCulture	Specifies the date culture to use when parsing dates. If not specified, Nexus will use the system date culture.	<i>Not Present</i>
InversionRule	<p>Specifies how inversions are treated in the input data. Possible values are:</p> <p><b>AlwaysTrue:</b> It will be assumed that there is always an inversion</p> <p><b>AlwaysFalse:</b> It will be assumed that there is never an inversion</p> <p><b>MatchValue:</b> An inversion is present when the value in <i>InversionColumn</i> matches the value specified by <i>InversionValue</i></p> <p><b>UseExisting:</b> If there is existing met data in the database, its inversion predictions will be copied to the new data.</p> <p><b>NumericValue or LessThan:</b> An inversion is present if the value in <i>InversionColumn</i> is less than the value specified by <i>InversionValue</i></p> <p><b>GreaterThan:</b> An inversion is present if the value in <i>InversionColumn</i> is greater than the value specified by <i>InversionValue</i></p> <p><b>LessThanOrEqual, GreaterThanOrEqual:</b> See above, but add ‘...or equal to...’</p>	AlwaysTrue
InversionColumn	Specifies which column contains inversion data. Not required if <i>InversionRule</i> is set to <b>AlwaysTrue</b> , <b>AlwaysFalse</b> , or <b>UseExisting</b> .	<i>User entered</i>
InversionValue	Specifies the comparison value to use to determine if an inversion is present. Not required if <i>InversionRule</i> is set to <b>AlwaysTrue</b> , <b>AlwaysFalse</b> , or <b>UseExisting</b> .	<i>User entered</i>
RainColumn	Specifies the column that contains rain information	<i>User entered</i>

Setting	Description	Default Value
TemperatureColumn	Specifies the column that contains temperature information	<i>User entered</i>
WindDirectionColumn	Specifies the column that contains the wind direction	<i>User entered</i>
WindSpeedColumn	Specifies the column that contains the wind speed	<i>User entered</i>
WindSpeedCorrectionFactor	<p>A number that is used to convert the value in <i>WindSpeedColumn</i> to wind speed at 10m height, in m/s. The following formula is used:</p> $WS_{10m} = WS * WSCF$ <p>Where:</p> <p><b>WS<sub>10m</sub></b> is the windspeed at 10m, in m/s.</p> <p><b>WS</b> is the value in the <i>WindSpeedColumn</i></p> <p><b>WSCF</b> is the value of <i>WindSpeedCorrectionFactor</i></p>	.277777778 (The conversion factor for km/h -> m/s)

Additionally, each subkey must contain a source subkey, with the following settings:

Setting	Description	Default Value
Interpolating	Specifies whether interpolation should be used between consecutive data points. If data is low resolution (e.g. 3 hours), it is suggested that this is set to <b>True</b> , otherwise set to <b>False</b> .	True
MaxValidity	The maximum valid duration of each record within the data source, in HH:MM:SS	1:00:00
Name	The name of the data source. E.g. Forecast, or Live	<i>User entered</i>
Priority	A priority to determine whether data from this source should overwrite other data sources. A larger number indicates higher priority. Multiple sources with the same priority are able to overwrite each other	1 for forecast, 2 for live



## 5 Reporting

Nexus provides automated reporting capabilities of received levels for compliance assessment. These reports are generated by the Nexus Service based on parameters defined in the configuration file.

Additionally, the user can manually generate one off reports through the user interface, as described in the user manual.

## 6 Database Schema

The following section is provided for information only. It is to be noted that the database schema does change from time to time as necessitated by functional changes made to Nexus; as such any automated procedures based on direct database access need to be made in consultation with the Nexus team. It is also to be noted that modifications made to the database using tools other than the Nexus User Interface may cause unexpected or inconsistent behaviour within Nexus, or may prevent the program from working and providing correct results.

### 6.1 Tables

#### 6.1.1 Blasting

The Blasting schema contains tables that relate to the Blasting extension.

##### 6.1.1.1 Blasting.BlastActivities

This table contains one entry for each activity that uses the Blasting extension

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
ActivityID	INT	Scheduling.Activities (ID)	The Activity that each entry corresponds to
EmulsionCategoryID	INT	Blasting.EmulsionCategories	The emulsion type that will be used in each blasting activity
StemmingCategoryID	INT	Blasting.StemmingCategories	The stemming type that will be used in each blasting activity
EmulsionWeightPerHole	FLOAT		The mass of emulsion that will be used in each hole, in kilograms
EmulsionTotalWeight	FLOAT		The total mass of emulsion that will be used, in kilograms
StemmingWeightPerHole	FLOAT		The mass of stemming material that will be used in each hole, in kilograms
StemmingTotalWeight	FLOAT		The total mass of stemming material that will be used, in kilograms

Column Name	Datatype	References	Description
DelayPerBlast	FLOAT		The delay between consecutive blasts, in milliseconds

#### 6.1.1.2 Blasting.BlastActivityDefaults

This table contains information about default values that will be used when an Activity is added that uses the blasting extension

See Blasting.BlastActivities for a description of each column. The only difference is that this table contains ActivityCategoryID that references Scheduling.ActivityCategories instead of ActivityID.

#### 6.1.1.3 Basting.BlasticConfig

This table contains a single record that stores some blast configuration settings

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
BaseVOD	INT		The base detonation speed used in blasting calculations, in m/s
EquipmentSWLID	INT	Simulation.Equipment(ID)	The ID of the entry in the EquipmentSWL table that contains the spectrum to be used for blast calculations

#### 6.1.1.4 Blasting.EmulsionCategories

This table contains one record for each emulsion type that can be used for blast calculations

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	VARCHAR(200)		OEM name of the emulsion
VOD	FLOAT		The detonation speed of the emulsion

#### 6.1.1.5 Basting.StemmingCategories

This table contains one record for each stemming material that can be used for blast calculations

Column Name	Datatype	References	Description
ID	INT		The primary key of the table

Column Name	Datatype	References	Description
Name	VARCHAR(200)		The friendly name of the material
Impedance	FLOAT		The impedance of the material
AttenuationFactor	FLOAT		The attenuation factor of the material

### 6.1.2 Calculation

This schema contains tables that store calculation results and information about which calculations need to occur.

#### 6.1.2.1 Calc.CalcCategories

This table contains two entries, used to differentiate between Assessed and Received level calculation records:

ID: 1, Name: Logged Level

ID: 2, Name: Assessed Level

#### 6.1.2.2 Calc.CalcResultsCache

This table records the calculated impacts of each equipment roster. This table contains one record for each change in the assessed or received level, due to each equipment roster in the database.

Column Name	Datatype	References	Description
CalcCategoryID	SMALLINT	Calc.CalcCategories(ID)	The calculation category of each record
StartTime	DATETIME2(2)		The time from which each record is valid
EquipmentRosterID	INT	Scheduling.EquipmentRoster(ID)	The equipment roster that the record corresponds to
ReceiverSet	SMALLINT	Simulation.ReceiverCategories(ID)	Indicates which receivers this record contains data for.
PredictedLevels	VARBINARY(2000)		Binary data that contains the received or assessed level at each receiver.

### 6.1.2.3 Calc.ExpiredEquipmentSchedules

This table contains information about which equipment rosters require recalculation

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
EquipmentRosterID	INT	Scheduling.EquipmentRoster(ID)	The equipment roster that requires recalculation

### 6.1.2.4 Calc.ExpiredMetPeriods

This table contains information about which time periods require recalculation

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
StartTime	DATETIME		The start of the period requiring recalculation
EndTime	DATETIME		The end of the period requiring recalculation

### 6.1.2.5 Calc.MetConditionsCache

This table contains information instructing the calculation engine which met condition to use at any point in time

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
StartTime	DATETIME		The time from which this met condition should be used
MetID	SMALLINT	Simulation.MetConditions(ID)	The met condition to be used

### 6.1.2.6 Calc.ReceiverThresholdsCache

This table contains information that describes how prescribed receiver limits vary with time.

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
StartTime	DATETIME		The time from which this receiver limit should be used

Column Name	Datatype	References	Description
ReceiverID	INT	Simulation.Receivers(ID)	The receiver to which this limit applies
NoiseCategoryID	SMALLINT	Simulation.NoiseCategories(ID)	The noise category for which this limit is relevant
Limit	FLOAT		The limit in dimensionless linear level. Equal to $10^{(\text{Limit\_db}/10)}$

### 6.1.2.7 Calc.ScenarioModifiedResults

This table records information that overrides information in Calc.CalcResultsCache for specific scenarios. This table contains results for equipment rosters and activities that have entries in the Scenarios.ActivityChanges or Scenarios.RosterCanges tables.

This table has the same schema as Calc.CalcResultsCache, with the addition of ScenarioID that references Scenarios.Scenarios, and specifies which scenario the modified results are for.

## 6.1.3 Logging

This schema contains information related to physical noise monitors and measurements

### 6.1.3.1 Logging.Categories

This table contains an enumeration of available logging categories. This table is populated at database creation and is never modified.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	VARCHAR(200)		The friendly name of the logging category

### 6.1.3.2 Logging.LoggingIdentifiers

This table contains one entry for time series measurement.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	VARCHAR(100)		The friendly name of the measurement
Description	VARCHAR(3000)		A description of the measurement
ReceiverID	INT	Simulation.Receivers	The receiver for which this measurement is relevant

Column Name	Datatype	References	Description
LoggingCategoryID	INT	Logging.Categories	The logging category of this measurement
RecordDuration	INT		The maximum valid duration, in seconds, of each individual measurement within the time series

### 6.1.3.3 Logging.Values

This table contains one record for each individual measurement at any point in time

Column Name	Datatype	References	Description
LoggingIdentificationID	INT	Logging.LoggingIdentifiers	The identifier to which this measurement belongs
Time	DATETIME		The time at which this measurement occurred
Level	FLOAT		The overall level of this measurement, in decibels
Spectrum	BINARY(66)		This spectrum of the measurement, if available

### 6.1.4 Scenarios

This schema contains information relevant to scenarios, that can be created to provide quick alternative plans.

#### 6.1.4.1 Scenarios.ActivityChangesEnum

This table contains a list of possible activity changes. It is populated at database creation and does not change.

Column Name	Datatype	References	Description
ID	TINYINT		The primary key of the table
Name	VARCHAR(255)		The name of the change

#### 6.1.4.2 Scenarios.ActivityChanges

This table records changes to Activities that occur in a particular scenario.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
ScenarioID	SMALLINT	Scenarios.Scenarios	The scenario for which this change is relevant
ActivityID	INT	Scheduling.Activities	The activity to which this change applies
ChangeType	TINYINT	Scenarios.ActivityChangesEnum	The type of change this record contains
NewDateTime	DATETIME2(2)		The new Date/Time value that should be used for the particular property in the scenario
NewBoolean	BIT		The new Boolean value that should be used for the particular property in the scenario

#### 6.1.4.3 Scenarios.RosterChangesEnum

This table contains a list of possible equipment roster changes. It is populated at database creation and does not change.

Column Name	Datatype	References	Description
ID	TINYINT		The primary key of the table
Name	VARCHAR(255)		The name of the change

#### 6.1.4.4 Scenario.RosterChanges

This table records the changes that apply to EquipmentRosters in a particular scenario.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
ScenarioID	SMALLINT	Scenarios.Scenarios	The scenario for which this change is relevant
EquipmentRosterID	INT	Scheduling.EquipmentRoster	The equipment roster to which this change applies



Column Name	Datatype	References	Description
ChangeType	TINYINT	Scenarios.RosterChangesEnum	The type of change this record contains
NewValue	INT		The new value that should be recorded against the particular property in the scenario

#### 6.1.4.5 Scenarios.Scenarios

This table contains a list of scenarios that exist in the database.

Column Name	Datatype	References	Description
ID	SMALLINT		The primary key of the table
Name	VARCHAR(255)		The name of the scenario

#### 6.1.5 Scheduling

This schema contains information about the mining schedule.

##### 6.1.5.1 Scheduling.Activities

This table contains one record for each activity in the database.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
ActivityCategoryID	BIGINT	Scheduling.ActivityCategories	The category of the activity
PathID	INT	Simulation.Paths	The op area on which the activity occurs
StartTime	DATETIME2(2)		The start time of the activity
EndTime	DATETIME2(2)		The end time of the activity
Muted	BIT		Indicates whether this activity is muted
ScenarioID	SMALLINT	Scenarios.Scenarios	Indicates if this activity only exists in a particular scenario

##### 6.1.5.2 Scheduling.ActivityCategories

This table contains a list of pre-defined activity categories that can be placed in the schedule.

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
Name	NVARCHAR(200)		The name of the activity category
OpGroupID	BIGINT	Scheduling.OpGroups	The op group to which this activity belongs
PathID	INT	Simulation.Paths	For fixed activities only: Specifies on which op area this activity category can occur
NoiseCategoryID	SMALLINT	Simulation.NoiseCategories	Specifies which type of noise this activity category generates
CalculationMethod	NVARCHAR(200)		Specifies if this activity uses an extension to perform its noise calculations.
UnID	UNIQUEIDENTIFIER		A GUID that identifies this activity category, possibly across different databases

### 6.1.5.3 Scheduling.ActivityTemplates

This table contains information about the default equipment that is placed when an activity category is instantiated into a new activity.

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
ActivityCategoryID	BIGINT	Scheduling.ActivityCategories	The activity category this template applies to
EquipmentSWLID	INT	Simulation.EquipmentSWLs	The equipment to be created
EquipmentCount	INT		The number of pieces of equipment to be placed
SourceLocID	INT	Simulation.SourceLocs	For fixed activities only: specifies

Column Name	Datatype	References	Description
			where to place the equipment
LocationCagoryOverrideID	SMALLINT	Simulation.LocationCategories	Specifies an optional override that affects where the equipment is placed
UnID	UNIQUEIDENTIFIER		A GUID that identifies this activity template, possible across different databases.

#### 6.1.5.4 Scheduling.AuditLog

This table contains a record of user interactions

#### 6.1.5.5 Scheduling.EquipmentRoster

This table contains information about individual pieces of equipment within each activity

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
SourceLocID	INT	Simulation.SourceLocs	The location of the equipment roster
EquipmentSWLID	INT	Simulation.EquipmentSWL	The type of equipment of this roster
EquipmentCount	INT		The number of pieces of equipment operating
ActivityID	INT	Scheduling.Activities	The activity to which this roster belong
IsAnimated	BIT		Indicates whether this equipment roster contains moving equipment
ScenarioID	SMALLINT	Scenarios.Scenarios	Indicates if this equipment roster exists only in a particular scenario

### 6.1.5.6 Scheduling.IndividualEquipmentRoster

This table contains information about each piece of equipment within an equipment roster, if the roster is animated.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
EquipmentRosterID	INT	Scheduling.EquipmentRoster	The equipment roster to which this piece of equipment belongs

### 6.1.5.7 Scheduling.IndividualEquipmentLocations

This table contains information specifying how each individual piece of equipment moves in an animated equipment roster.

Column Name	Datatype	References	Description
IndividualEquipmentRosterID	INT	Scheduling.IndividualEquipmentRoster	The individual equipment roster for which this record specifies a location
StartTime	DATETIME2(2)		The time from which this record is valid
SourceLocID	INT	Simulation.SourceLocs	The location of the equipment

### 6.1.5.8 Scheduling.LocationDefaultEquipment

This table contains information that overrides the information in Scheduling.ActivityTemplates for particular locations in particular op areas.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
PathSourceLocationID	INT	Simulation.PathSourceLocations	The path - source location combo for which this record applies.

Column Name	Datatype	References	Description
EquipmentSWLID	INT	Simulation.EquipmentSWLs	The equipment that should be placed
EquipmentCount	SMALLINT		The number of pieces of equipment that should be placed
CycleTime_Seconds	FLOAT		Not Used
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

#### 6.1.5.9 Scheduling.MetCache

This table contains the weather data that Nexus has read.

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
Time	DATETIME		The time of each record
InUse	BIT		Indicates whether this record is to be used in calculations
Windspeed	FLOAT		The windspeed, in m/s at 10m AGL
WindDireciton	FLOAT		The wind direction, in degrees
Temperature	FLOAT		The temperature, in degrees C
MetSourceID	BIGINT	Scheduling.MetSources	The met source from which this record was imported
Rain	NVARCHAR(200)		The recorded rain
Inversion	BIT		Whether an inversion is present

#### 6.1.5.10 Scheduling.MetSources

This table contains a list of data sources from which weather data has been imported

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
Name	NVARCHAR(200)		The name of this data source
MaxValidity	INT		The maximum validity of each record, in seconds
Interpolating	BIT		Indicates whether interpolation should be used on data from this source
LastImport	DATETIME		The last time data was imported from this source

#### 6.1.5.11 Scheduling.MetThresholdAdjustments

This table contains information about custom receiver limit adjustments that can be made based on weather conditions.

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
Col	NVARCHAR(200)		The column to test for this adjustment
Value	NVARCHAR(200)		The value to test against for this adjustment
NoiseCategoryID	SMALLINT	Simulation.NoiseCategories	The type of noise to which this adjustment applies
Variation	FLOAT		The amount to adjust the receiver limit, in decibels
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

#### 6.1.5.12 Scheduling.OpGroups

This table contains a list of operations groups into which activities are divided

Column Name	Datatype	References	Description
ID	BIGINT		The primary key of the table
Name	NVARCHAR(200)		The name of the op group
ReadOnly	BIT		Not used
DisplayOrder	INT		The display order of the op group
Deleted	BIT		Indicates whether this op group has been effectively deleted from the database, in which case this record only exists to provide historical referential integrity
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

### 6.1.6 Settings

This schema contains database specific settings

#### 6.1.6.1 Settings.Config

This table contains single value settings

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	NVARCHAR(100)		The name of the setting
Value	NVARCHAR(MAX)		The value of the setting
Description	NVCARCHAR(500)		A user friendly description of the setting

#### 6.1.6.2 Settings.ConfigUpdates

This table contains a record of major modifications made to the database

Column Name	Datatype	References	Description
ID	INT		The primary key of the table

Column Name	Datatype	References	Description
Version	NVARCHAR(200)		The Nexus Version of the tool that was used to apply the update
Comments	NVARCHAR(MAX)		Comments describing the update
DateCreated	DATETIME		The date that the update package was created
DateApplied	DATETIME		The date that the update package was applied to the database

### 6.1.6.3 Settings.GraphConfig

Reserved for future expansion.

### 6.1.7 Simulation

This schema contains information necessary to perform noise calculations.

#### 6.1.7.1 Simulation.Contour

This table contains a mesh that is used to calculate noise contours. Each record describes one triangle of the mesh.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Rec1ID	INT	Simulation.Receiver	The receiver on the first corner
Rec2ID	INT	Simulation.Receiver	The receiver on the second corner
Rec3ID	INT	Simulation.Receiver	The receiver on the third corner
Neighbour12ID	INT	Simulation.Contour	The first neighbouring
Neighbour23ID	INT	Simulation.Contour	The second neighbouring
Neighbour31ID	INT	Simulation.Contour	The third neighbouring

#### 6.1.7.2 Simulation.EquipmentLocationCategories

This table allows a many-to-many relationship between Simulation.EquipmentSWL and Simulation.LocationCategories



### 6.1.7.3 Simulation.EquipmentSWLs

This table contains a list of equipment that is available.

### 6.1.7.4 Simulation.LocationCategories

This table contains a list of location categories, that are used to determine where equipment is placed when an activity category is instantiated into an activity.

Column Name	Datatype	References	Description
ID	SMALLINT		The primary key of the table
Name	NVARCHAR(200)		The name of the location category
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

### 6.1.7.5 Simulation.MetConditions

This table contains a list of met conditions for which Nexus can perform noise calculations

Column Name	Datatype	References	Description
ID	SMALLINT		The primary key of the table
Windspeed	FLOAT		The windspeed, in m/s
WindDirection	FLOAT		The wind direction, in degrees
Inversion	BIT		Whether or not an inversion is present
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

### 6.1.7.6 Simulation.NoiseCategories

This table contains the various noise categories present in the database (i.e. LAEq and LLPeak)

Column Name	Datatype	References	Description
ID	SMALLINT		The primary key of the table
Name	NVARCHAR(200)		The name of the noise category
CumulativeOperator	NVARCHAR(200)		The operator that is used to combine multiple impacts

Column Name	Datatype	References	Description
Annotation	NVARCHAR(20)		The units that are associated with this noise category
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

#### 6.1.7.7 Simulation.PathGroups

This table provides an alternative grouping of op areas.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	NVARCHAR(200)		The name of the path group
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

#### 6.1.7.8 Simulation.Paths

This table contains a list of operations areas present in the database

Column Name	Datatype	References	Description
ID	SMALLINT		The primary key of the table
Name	NVARCHAR(200)		The name of the op area
PathGroupID	INT	Simulation.PathGroups	The group of the op area
OpGroup	NVARCHAR(200)	Scheduling.OpGroups (Name)	The op group of the op area
Deleted	BIT		A value indicating whether this record is effectively deleted from the database, and only maintained for historical referential integrity
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

#### 6.1.7.9 Simulation.PathSourceLocations

This table provides a multi-way relationship between Simulation.Paths and Simulation.SourceLocations, with Simulation.LocationCategory as a discriminator

#### 6.1.7.10 Simulation.PublicHolidays

This table contains a list of public holidays

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	NVARCHAR(200)		The name of the holiday
Holiday	DATETIME		The date of the holiday

#### 6.1.7.11 Simulation.ReceiverCategories

This table contains the possible receiver categories in Nexus: Contour, Logged, and Assessed. It is used to determine what type of results need to be calculated for each receiver.

#### 6.1.7.12 Simulation.Receivers

This table contains a list of receiver locations for which noise impacts can be calculated.

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	NVARCHAR(200)		The name of the receiver
X	FLOAT		The x-coordinate of the receiver
Y	FLOAT		The y-coordinate of the receiver
ReceiverCategoryID	SMALLINT	Simulation.ReceiverCategories	The category of the receiver
UnID	UNIQUEIDENTIFIER		A GUID that identifies this record, possibly across multiple databases.

#### 6.1.7.13 Simulation.ReceiverThresholds

This table contains instructions on how to calculate receiver thresholds

#### 6.1.7.14 Simulation.Shifts

This table contains information about the shifts used at site

Column Name	Datatype	References	Description
ID	INT		The primary key of the table
Name	NVARCHAR(200)		The name of the shift
StartHour	FLOAT		The start hour of the shift
EndHour	FLOAT		The end hour of the shift