HEXAGON
SAFETY & INFRASTRUCTURE

INTERGRAPH INSIGHT® ADVANTAGE FOR I/CAD

ADMINISTRATOR'S GUIDE

2.0.1811
November 2018

Copyright

Restricted Rights Legend

Terms of Use

Warranties and Disclaimers

Trademarks

CONTENTS

INTRODUCTION: INTERGRAPH INSIGHT® ADVANTAGE FOR I/CAD

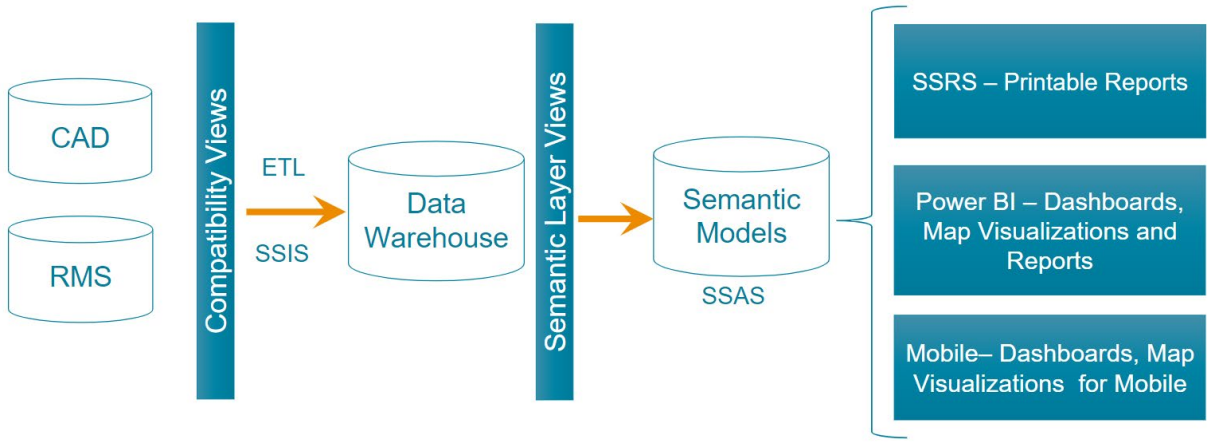
Intergraph InSight® Advantage for I/CAD from Hexagon Safety & Infrastructure® is a suite of products designed to transform the raw data captured by the Intergraph CAD application and turn it into meaningful and useful information that agencies can use to make informed data-driven decisions. InSight is a Business Intelligence (BI) application where BI refers to the management, transformation, storage, analysis, and presentation of this information. InSight leverages Microsoft's BI stack.

MICROSOFT® BI STACK

Microsoft's BI stack is a set of technology and platforms used to provide analytic and reporting capability to large data sets.

- **SQL Server Integration Services (SSIS)**: Provides data integration tools to extract, transform, and load data (ETL) into schemas optimized for reporting and analytics.
- **SQL Server Analysis Services (SSAS):** Provides tools to build multidimensional databases, develop data mining models, and query from cubes (semantic models).
- **SQL Server Reporting Services (SSRS)**: Provides visualization tools to build reports and a portal to deploy reports that are accessible to end users.
- **SQL Server Data Tools (SSDT)**: Provides a development environment of BI project authoring tools and project templates within Visual Studio for SQL BI.

INSIGHT TECHNICAL ARCHITECTURE



The architecture of the InSight solution is composed by six pieces: compatibility views, Extract Transform and Load (ETL), Data Warehouse, Semantic Layer Database, Semantic Models and Reporting Portal.

Compatibility views are a set of read-only views reading from the CAD or RMS database. It lets the ETL (SSIS code) support multiple versions of CAD and RMS and map custom columns with no custom code. Also, the compatibility views provide secure read-only access to the source database and filtering data before being read into the data warehouse (for example, remove sensitive or unwanted data).

ETL is the component responsible to extract the data from the source, transform and load into the data warehouse.

Data Warehouse is composed by databases that contains the data to answers the most relevant business questions. It transforms operational schemas into models that are understandable and usable by business users.

Semantic Layer Database models the data warehouse into subject areas.

Semantic Models is a layer in the solution where the models were built. The models are divided by subject areas and each of these subject areas with its own star schemas.

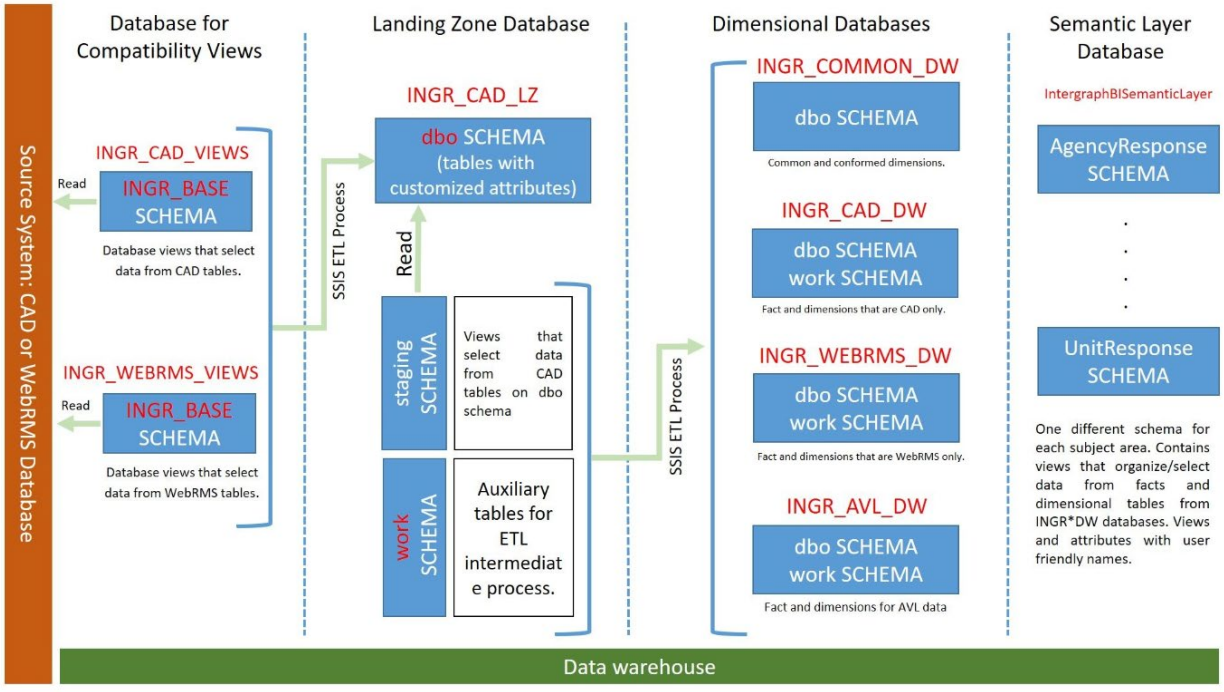
Power BI Report Server is the platform to store the reports. The reports connect to the semantic models to consume data, except for the operational reports that goes directly to the CAD database. There are three type of reports: Power BI Reports, Paginated Reports, and Mobile Reports.

DATA WAREHOUSE OVERVIEW

The Data Warehouse resides on a dedicated server running Microsoft SQL Server, a database management system (DBMS). SQL Server Integration Services (SSIS) is ETL software used to pull data from the source systems and load it into the data warehouse. The diagram below shows the data warehouse architecture built to support the InSight solution. It is composed by four categories of databases:

- Compatibility Views Database
- Landing Zone Database
- Dimensional Databases
- Semantic Layer Database

The ETL process has two distinct phases. One phase reads data from the source datasets and loads only the necessary range of data to the landing zone database. The landing zone database hosts a set of tables similar to the source. The second phase reads that data from the landing zone, transforms the data into fact and dimensions and loads the data into dimensional database..



DELIVERED BI DATABASES

Hexagon delivers seven databases as part of the BI deployment:

- **INGR_CAD_DW:** CAD dimensional database used for reporting.
- **INGR_CAD_LZ:** Stages the data for loading into the dimension databases.
- **INGR_CAD_VIEWS**: Stores the compatibility views and corresponding metadata.
- **INGR_CTL**: Controls/monitors the execution of ETL jobs.
- **INGR_COMMON**: Common dimension shares across applications: CAD and RMS used for reporting
- **INGR_AVL**: GPS/AVL dimensional database used for reporting.
- **IntergraphBISemanticLayer**: Semantic Layer database used to organize the dimensional databases into subject areas.

DELIVERED DATABASE LOGINS

There are three default database logins delivered with the product:

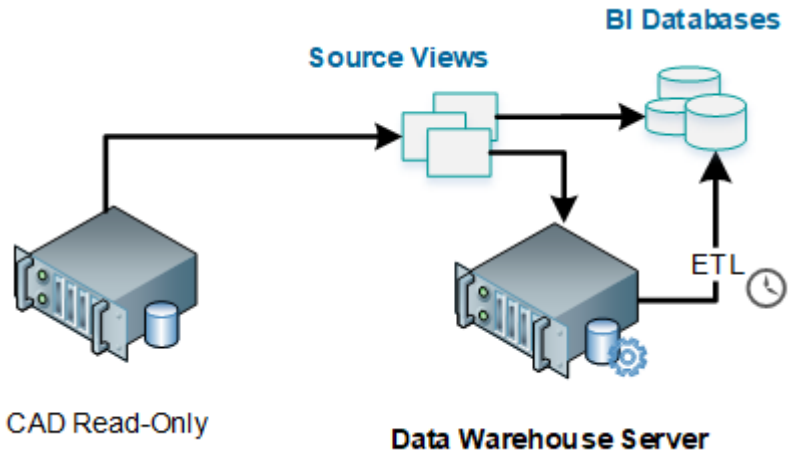
- **INGR_BASE**
  - Used to create and manage the Source Views, which directly link to the CAD database tables.
  - A user and a schema for INGR_BASE is created on INGR_CAD_VIEWS database and all views, tables, and functions are created under this schema by default.
  - Default owner of all objects in INGR_CAD_VIEWS database.
- **INGR_READER**
  - Provides read-only access on all delivered BI databases.
  - This user login is used by ALL delivered BI reports and dashboards by default.
- **INGR_ADMIN**
  - Default owner of all delivered BI Databases except for INGR_CAD_VIEWS database.

- Has read/write/delete permissions; used only by the ETL.

SOURCE VIEWS

For backward compatibility and isolation, several views are created against the source CAD database tables. These views are called Source Views**.**

- The ETL uses these Source Views to pull data from CAD that is loaded into the BI Data Warehouse through ETL.
- Source Views are created inside the INGR_CAD_VIEWS database under the INGR_BASE schema.



Source views have several advantages:

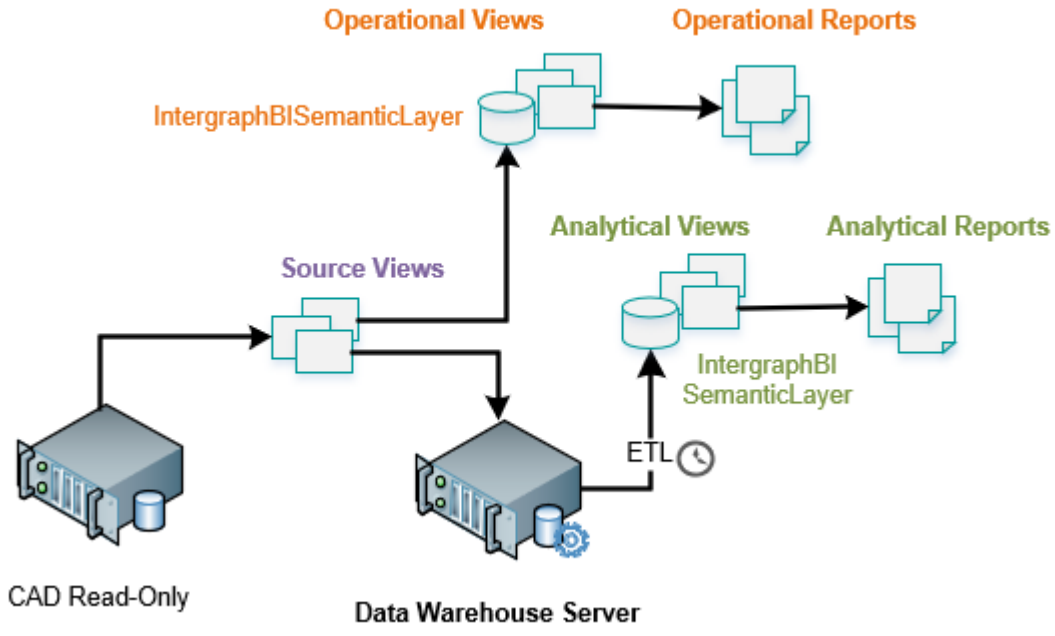
- They isolate CAD database tables from direct access.
- They protect ETL from CAD database changes and upgrades.
- They allow seamless migration from one CAD database to another.

SEMANTIC LAYER DATABASE

The Semantic Layer database simplifies the complexity of business data contained in the data warehouse. The Semantic Layer database includes the following:

- One schema per reporting subject area.
- A set of views that define the subject area as a dimensional model.
- Combines data from all BI databases into single consolidated views designed specifically for reporting.\
- Provides business users friendly and consistent names for the data objects

Views are organized by schemas into different subject areas within the semantic layer database.



OPERATIONAL VS. ANALYTICAL SEMANTIC LAYER VIEWS

OPERATIONAL VIEWS

The Operational Views are used for reporting on live data. They connect directly to a CAD database using the Source Views. The Source Views connect to a read-only replica of the live CAD database.

ANALYTICAL VIEWS

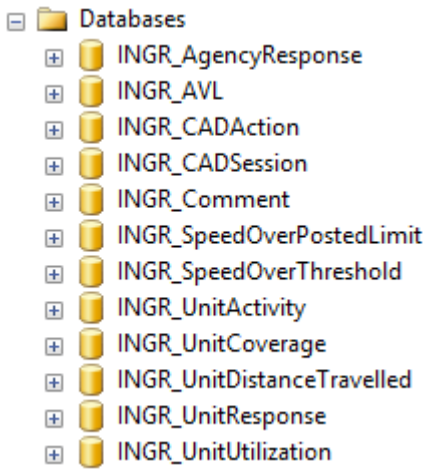
The Analytical Views are used to analyze the historical data, letting business users identify trends. These views connect to dimensional databases for historical data and are dependent on ETL for new data. The ETL must be run on regular intervals to keep data accessed by the analytical views up-to-date.

SEMANTIC MODELS (SSAS)

All reports use the semantic models as the source for their data. Semantic models are built using SQL Server Analysis Services (SSAS) models and these models connect directly to the Semantic Layer database.

There are two types of semantic layer models: analytical and operational.

ANALYTICAL

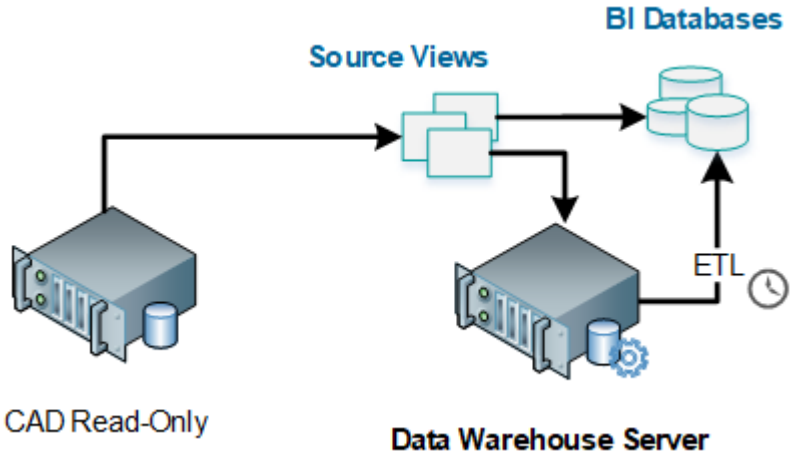


OPERATIONAL

Each type of model connects to the semantic layer database. Operational models are qualified with _Live to distinguish them from analytical models.

ETL

Extract, Transform, and Load (ETL) is the process of loading the CAD data into the data warehouse. The ETL process uses the *Source Views* (see " *Source Views*" on page 9) within the INGR_CAD_VIEWS database to read data. It does not connect to the CAD tables directly.



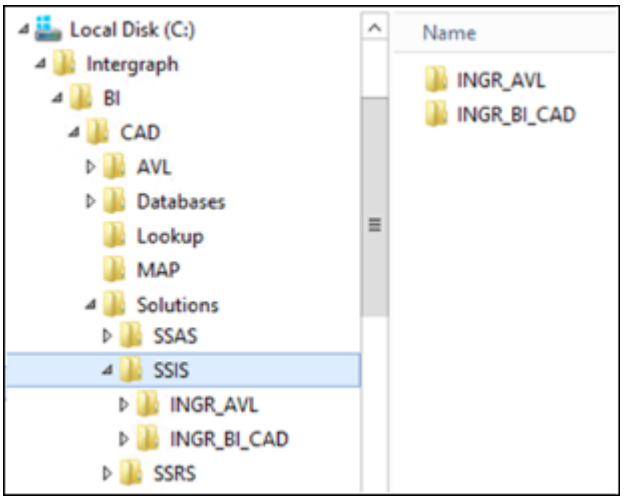
ETL jobs are split into two types of execution:

- The initial ETL job execution that reads an entire set of data and loads it into the data warehouse.
- Incremental ETL job executions that read only the incremental data that has been created or changed since the last ETL job was run. These ETL job executions are scheduled at regular intervals.

FORMAT AND LOCATION OF DELIVERED ETL

SQL Server Integration Services (SSIS) is a component of the Microsoft SQL Server database software that can be used to perform a broad range of data migration tasks. It features a data warehousing tool used for ETL processes, delivered in the form of Visual Studio 2015 solution files (.sln).

By default, the solution files are delivered at C:\Intergraph\BI\CAD\Solutions\SSIS.



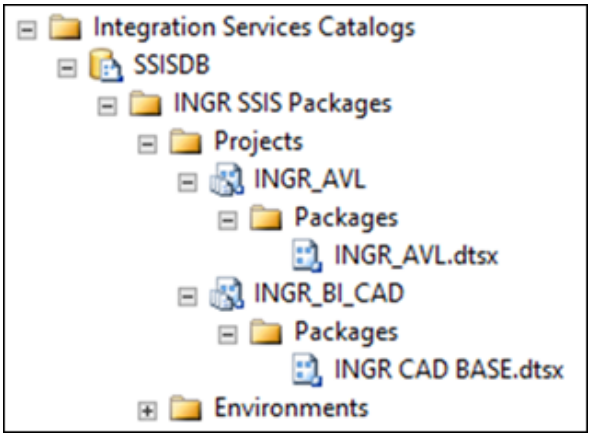
The two primary ETL solution files have different functions:

- INGR_BI_CAD performs most CAD ETL processes.
- INGR_AVL pulls automatic vehicle location (AVL) data only. INGR_AVL requires that INGR_BI_CAD is run first since the AVL data warehouse has dependencies on the CAD data warehouse

DEPLOYMENT OF ETL PACKAGES

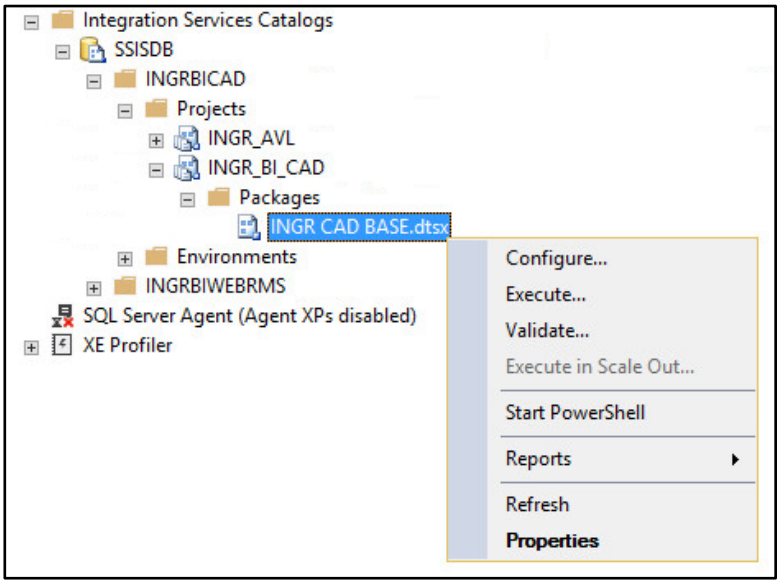
The ETL packages are deployed to the Integration Services Server automatically by the installer. The Integration Services Server is an instance of the SQL Server Database Engine that hosts the SSISDB database. By default, the Integration Services Server is the same as the Data Warehouse server.

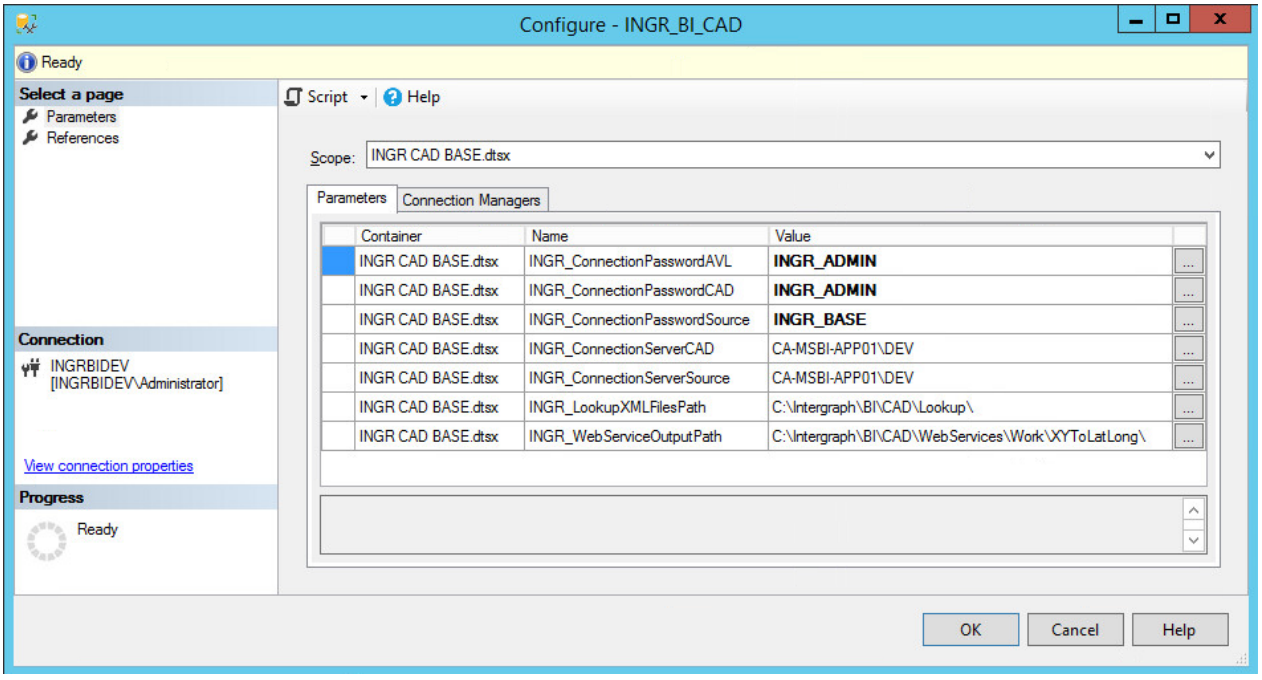
The \Integration Services Catalogs folder is the central point for working with Integration Services projects (such as ETL packages).



CONFIGURING ETL PACKAGES

All the parameters necessary for the ETL execution are defined during the installation process. If changes are needed after the installation, an administrator can re-define the parameter values accessing the configuration page (image below) of the package, using SQL Server Management Studio.





INGR_ConnectionPasswordAVL	Password of the INGR_ADMIN user; used to connect to INGR_AVL_DW database.
INGR_ConnectionPasswordCAD	Password of the INGR_ADMIN user; used to connect to INGR_CAD_DW database.
INGR_ConnectionPasswordSource	Password of the INGR_BASE user; used to connect to INGR_CAD_VIEWS database.
INGR_ConnectionServerCAD	Database instance where the data warehouse resides.
INGR_ConnectionServerSource	Database instance where the INGR_CAD_VIEWS reside. (the same instance as the data warehouse instance).
INGR_LookupXMLFilesPath	Path where the XML files are located.
INGR_WebServiceOutputPath	Path where the Web Service generate output files.

CONFIGURING A SQL JOB TO RUN THE ETL

The steps below detail the process to create a job inside SQL Server to run the ETL. ETL jobs are configured to run on a schedule. This is done by creating a job in SQL Server Agent.

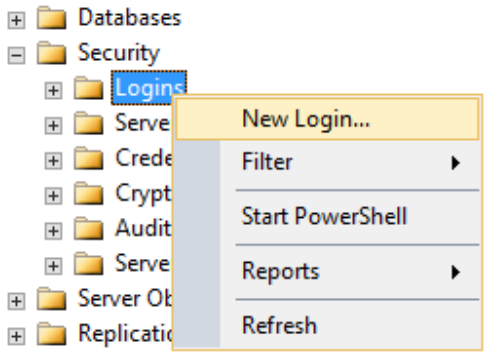
CREATE LOGIN, CREDENTIAL, AND PROXY

A SQL Server Agent proxy account defines a security context in which a job step can run. In this context, a job step can be defined by a SSIS package that needs to run in regular basis to load the data from the source to the data warehouse. A proxy provides SQL Server Agent with access to the security credentials for a Microsoft Windows user.

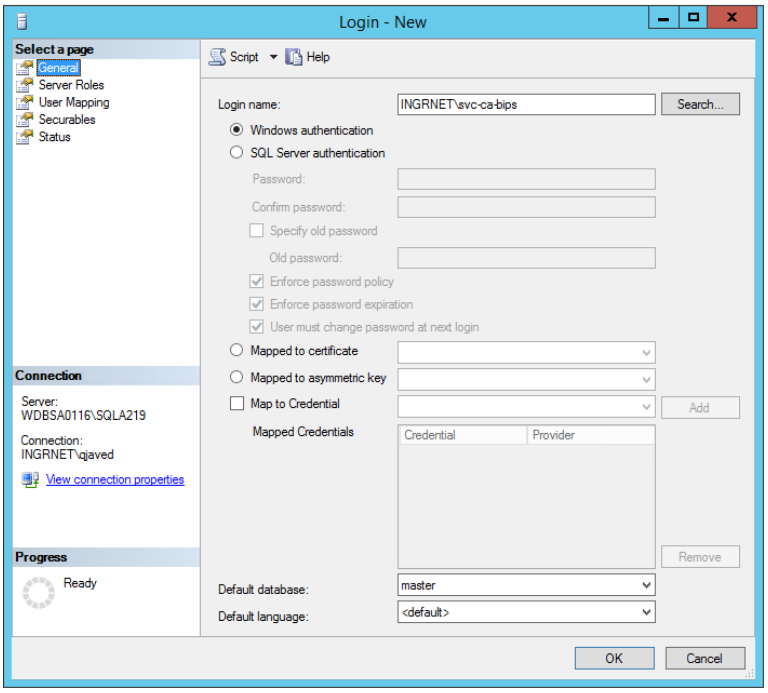
Before the SQL Server Agent runs a job step that uses a proxy, the SQL Server Agent impersonates the credentials defined in the proxy, and then runs the job step by using that security context.

The steps below create a domain user account and the credentials that are used to create and run the job.

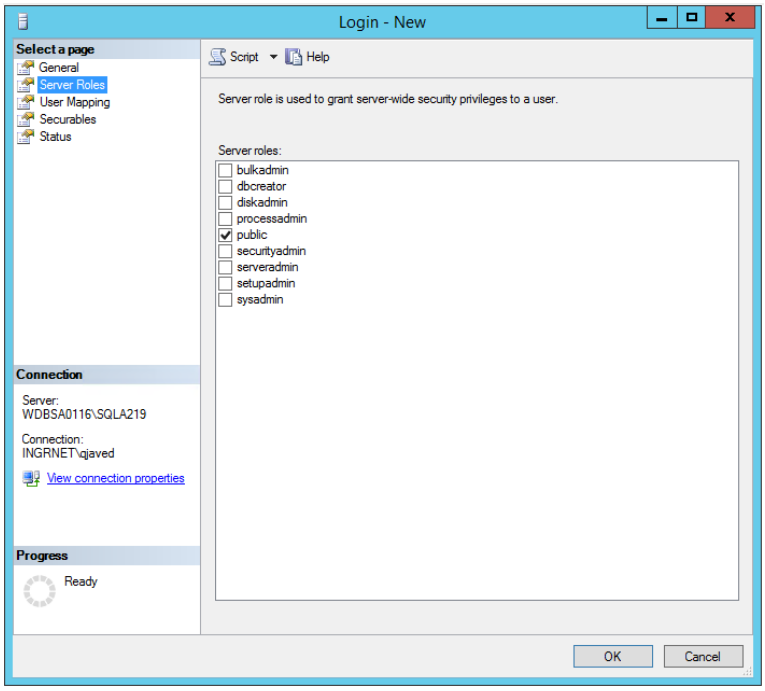
1. Open **SQL Server Management Studio**.
2. Open a connection to the Database Engine.
3. Navigate to **Security > Logins**.
4. Right-click and select **New Login.**



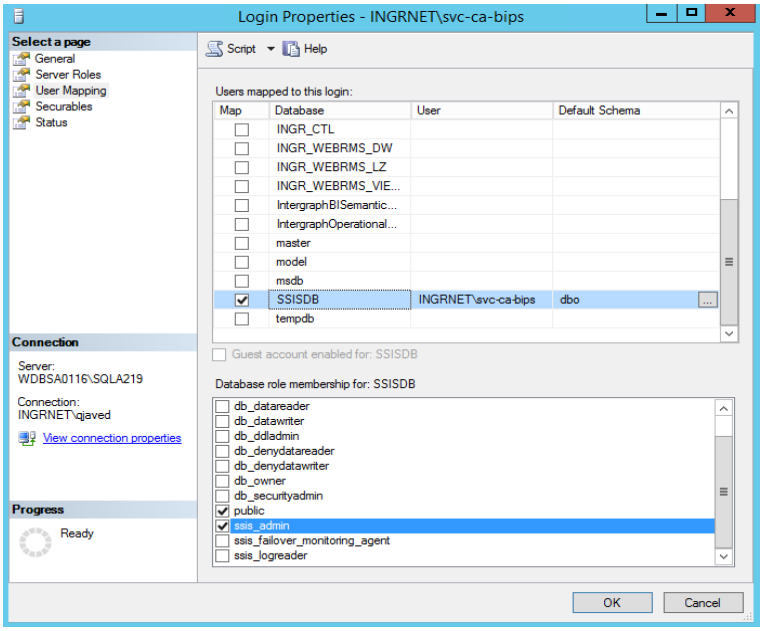
5. Under the **General** tab, pick a **Windows AD** user for **Login Name**.



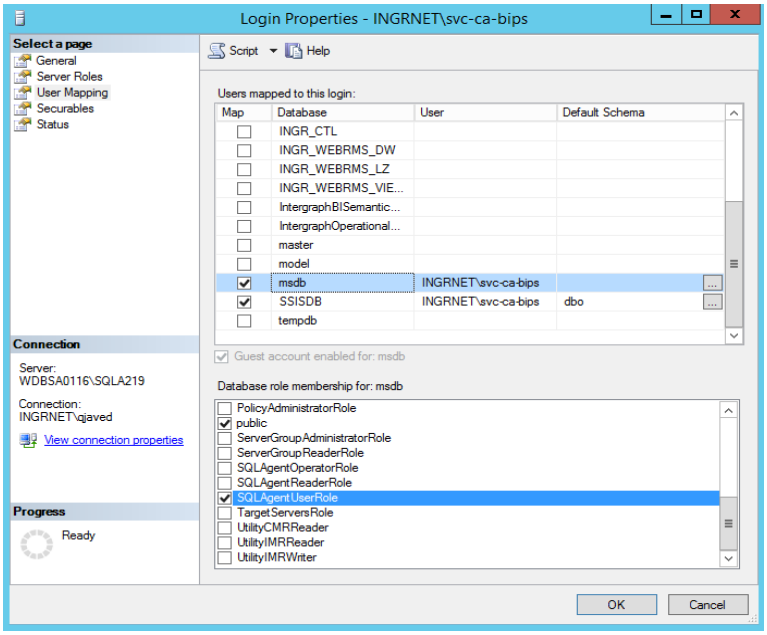
6. On **Server Roles**, select **public**.



7. Under **User Mapping**, select the **SSISDB** database and then select the **ssis_admin** role.

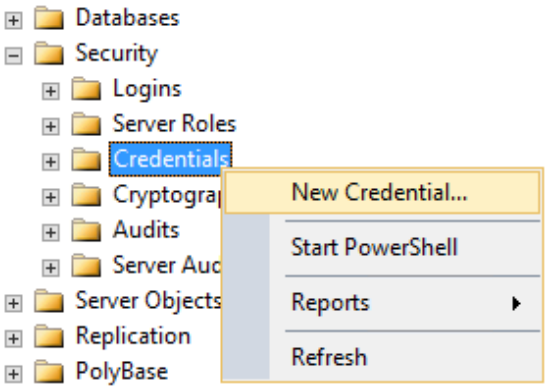


8. Select the **msdb** database and check **SQLAgentUserRole.**

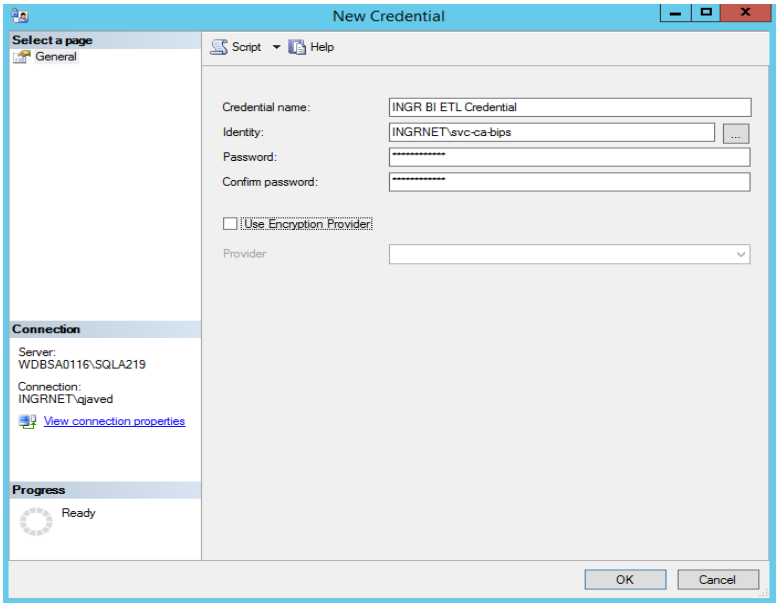


9. Click **OK** to close the **Login Properties** dialog.

10. Verify that you see a new Login created.
11. Navigate to **Security > Credentials**.
12. Right-click and select **New Credential.**

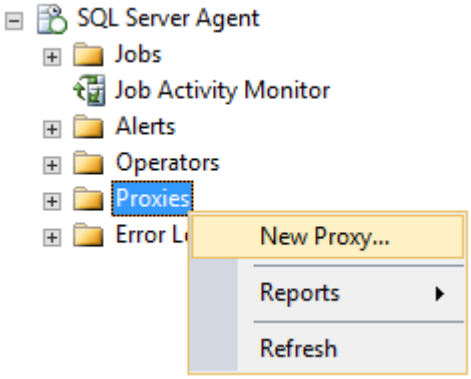


13. In the **New Credential** window, enter the following:
    - Credential name = INGR BI ETL Credential
    - Identity = <Windows AD login created in Step 5>
    - Password = <password for Windows AD login>
    - Confirm password = <password for Windows AD login>

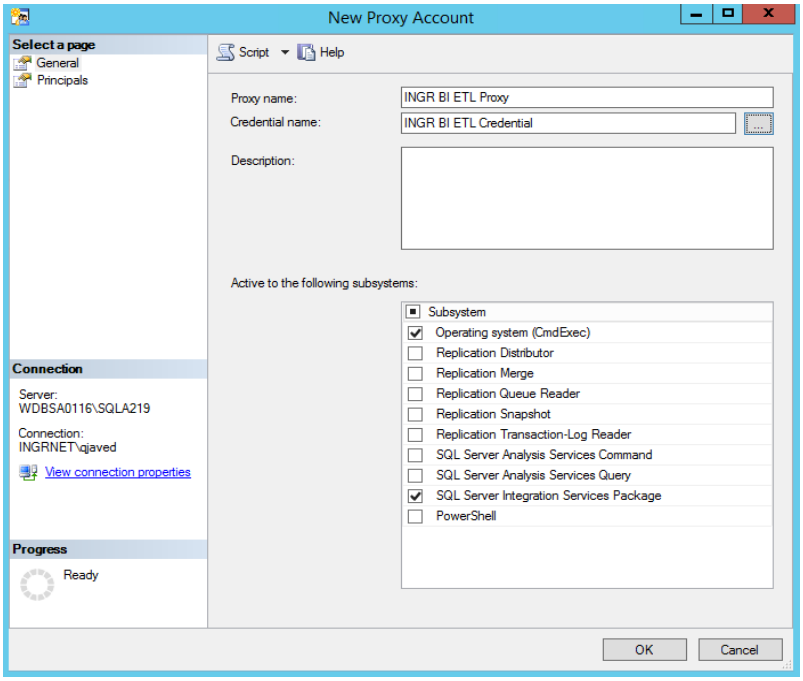


14. Select **OK**.
15. Verify that you see a new Credential created.

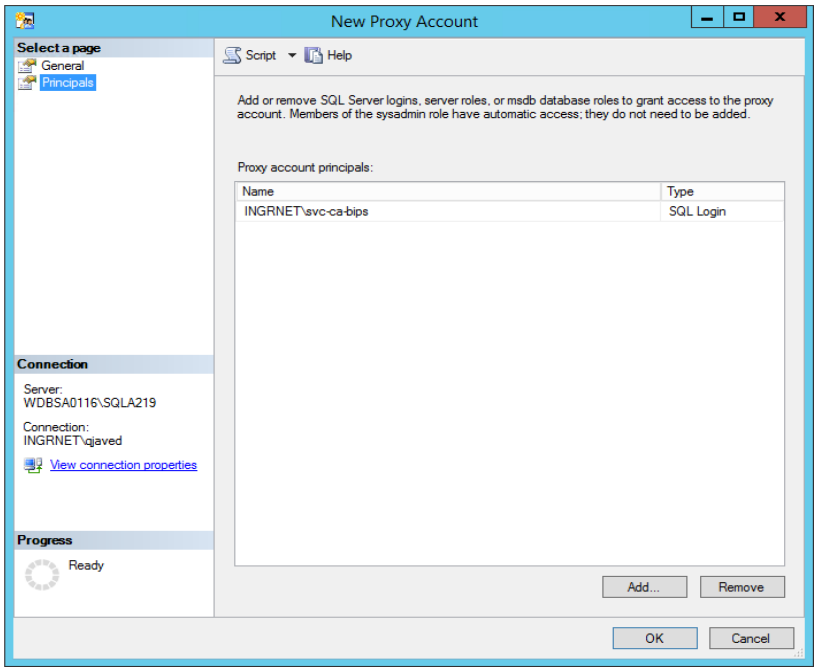
16. Navigate to **SQL Server Agent > Proxies**.
17. Right-click and select **New Proxy.**



18. On the **New Proxy Account** dialog box, from the **General** tab, enter the following:
    - Proxy Name = INGR BI ETL Proxy
    - Credential Name = INGR BI ETL Credential
    - Check the following options:Operating system (CmdExec)
    - SQL Server Integration Services Package



19. On the **New Proxy Account** dialog box, from the **Principals** tab, click **Add** and select the **Windows AD login** we used for the Credential. Click **OK** to create new proxy.

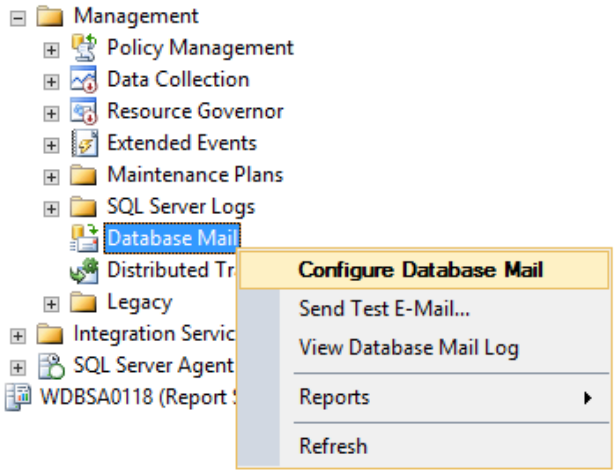


20.

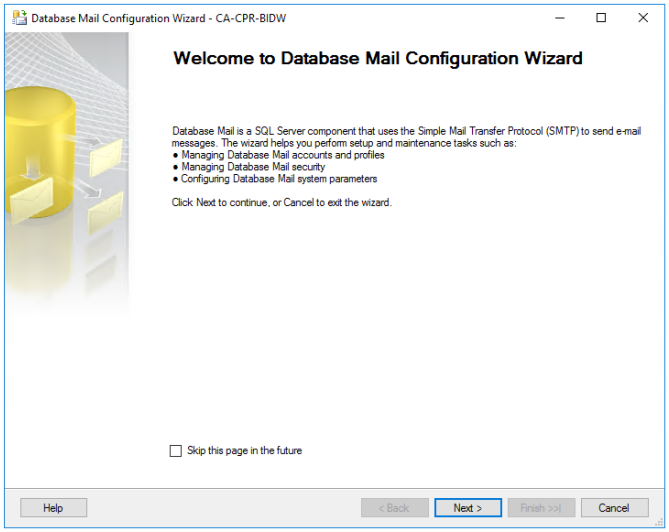
SETUP DATABASE MAIL

This section allows the setup of an email account to use to send out messages about the ETL or receive emails when the ETL fails.

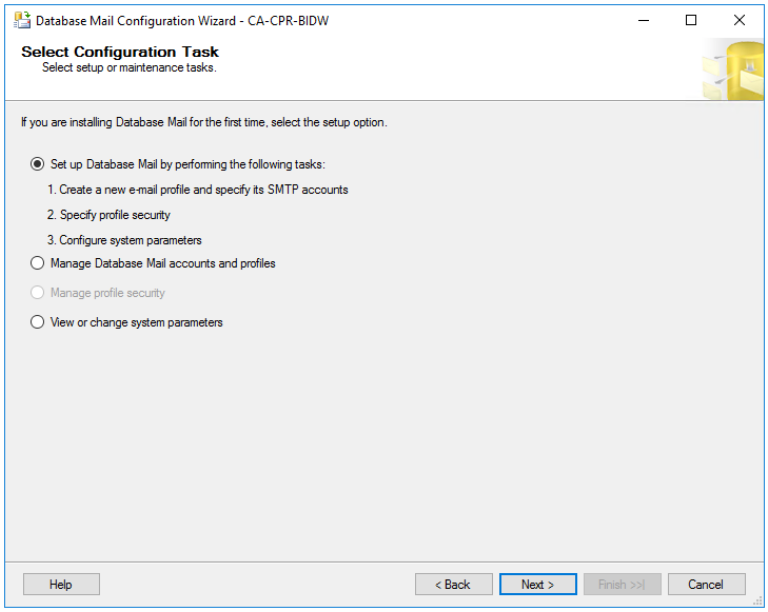
1. Open **SQL Server Management Studio**.
2. Create a connection to the Database Engine.
3. Navigate to **Management > Database Mail**.
4. Right-click on **Database Mail** and select **Configure Database Mail.**



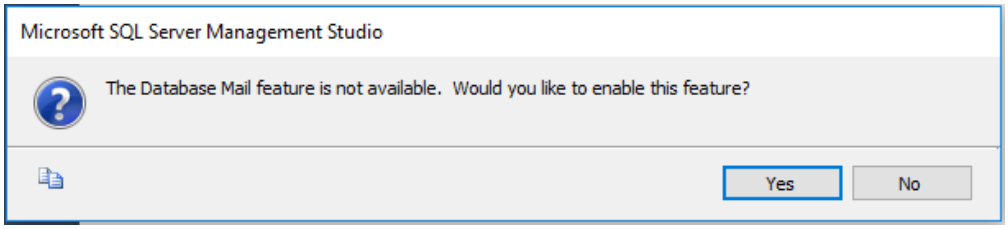
5. This will open the **Database Mail Configuration Wizard.**



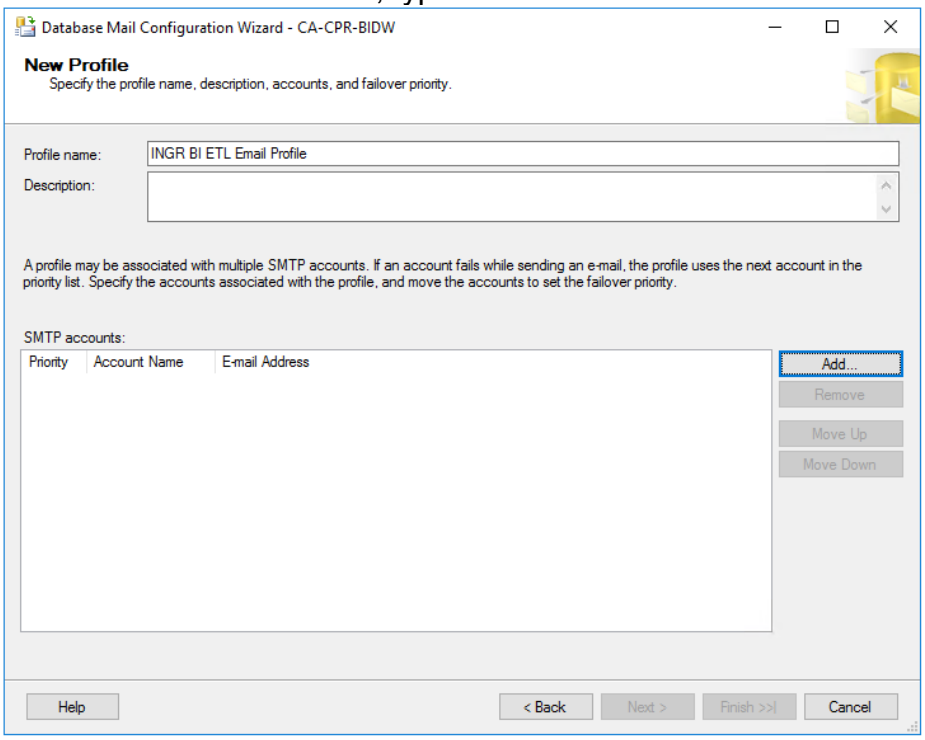
6. On the Wizard screen, select **Next**.
7. On the Select Configuration Task screen, ensure that **Set up Database Mail by performing the following tasks** is selected.



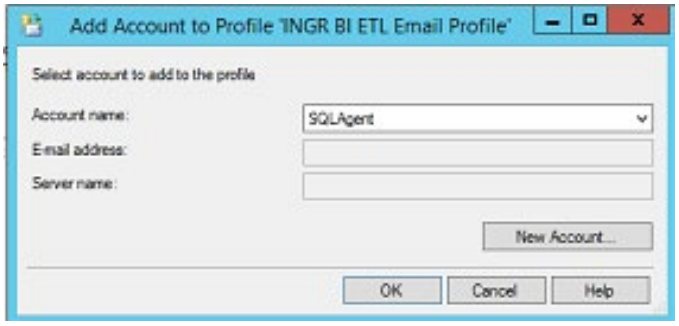
8. Select **Next**.
9. If a prompt displays, click on **Yes** to enable the Database Email feature.



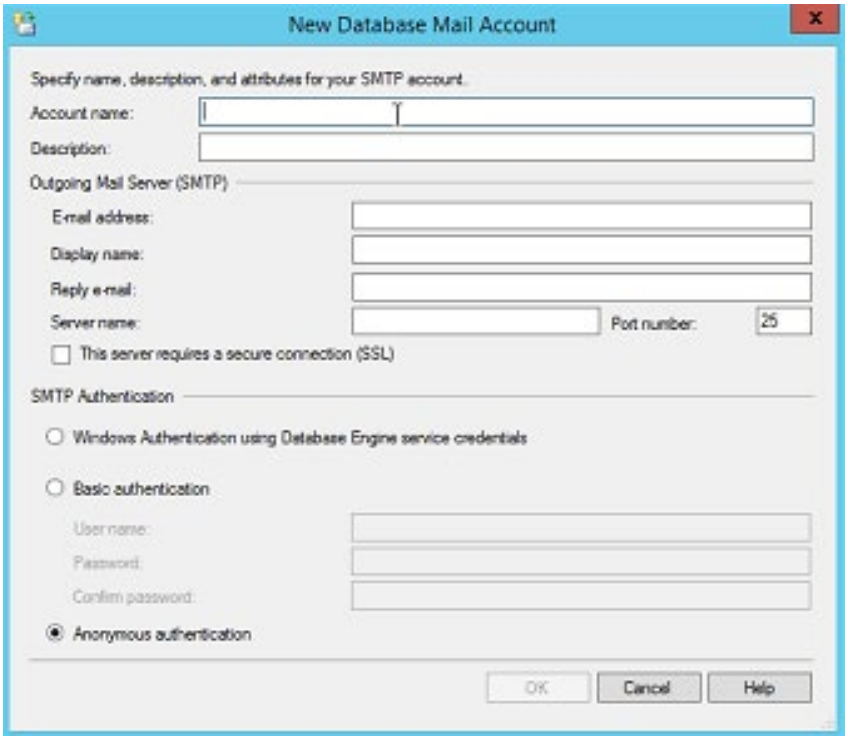
On the New Profile screen, type **INGR BI ETL Email Profile** for the Profile Name.



10. On the New Profile screen, click **Add**.



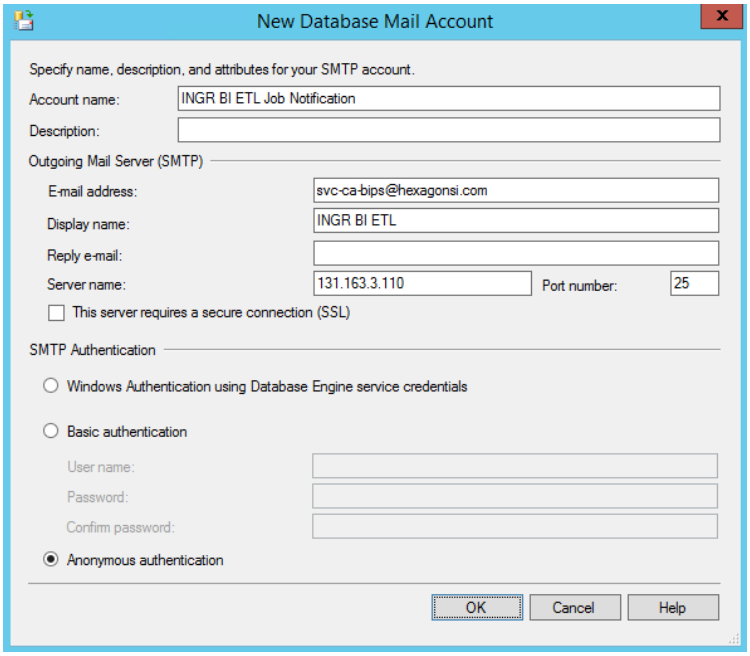
11. On the Add Account to Profile screen, select the **New Account** button.



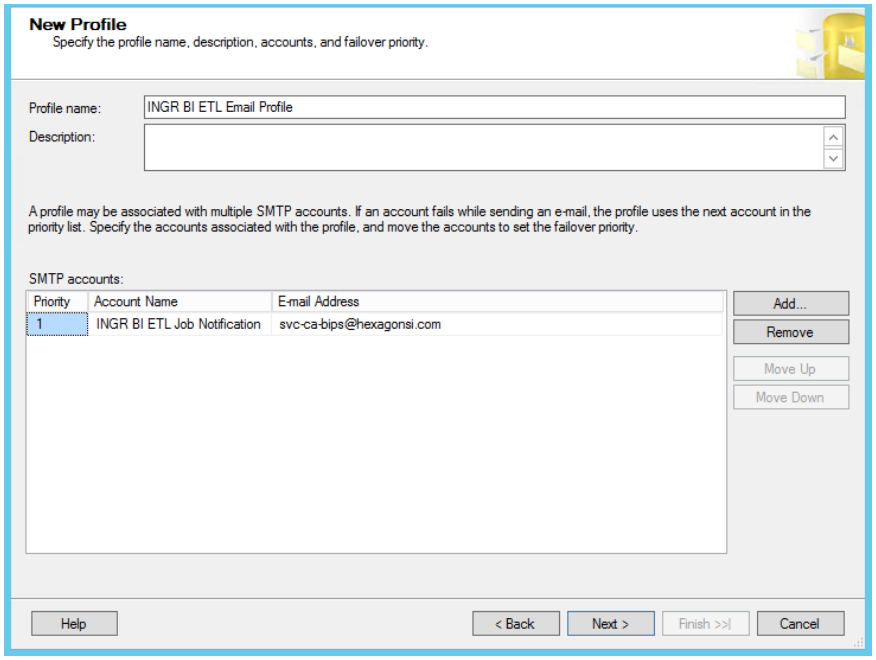
12. On the **New Database Mail Account** dialog box, fill out **SMTP** details as follows:

- Account Name = INGR BI ETL Job Notification
- Email Address = <email address that is used to send out notifications>
- Display Name = <the name you want it to be called>
- Reply email = <email address that is used for replies>
- Server Name = <name of the SMTP email server>

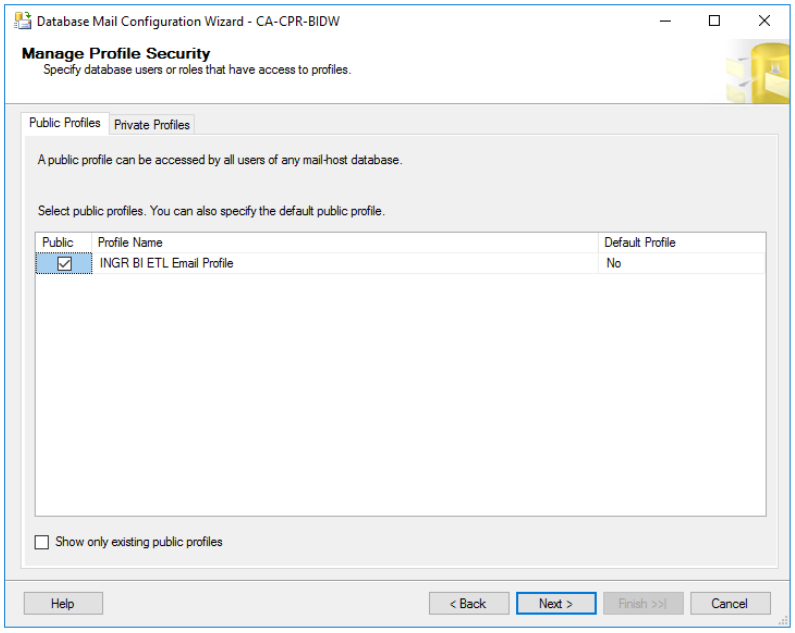
13. An example is given below for reference only:



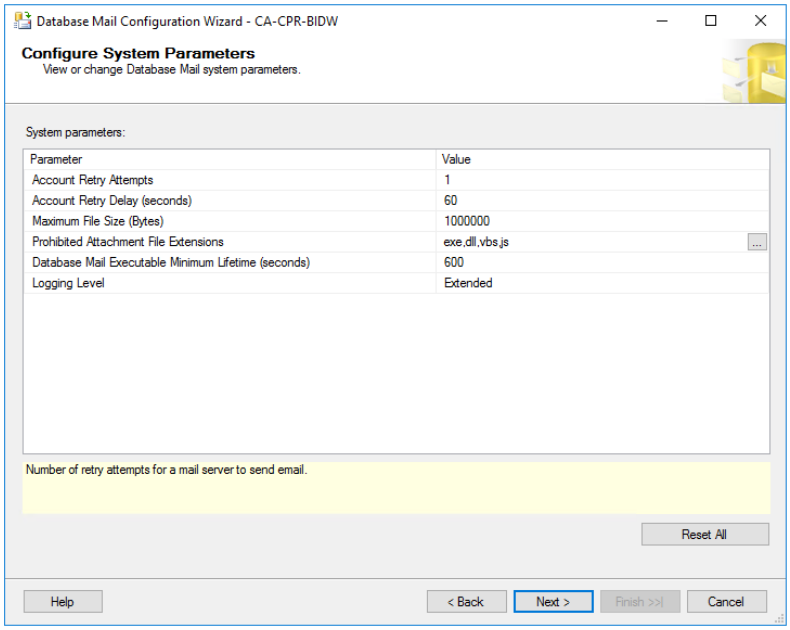
14. Select **OK** on the **New Database Mail Account** dialog box. This returns you back to the New Profile screen.



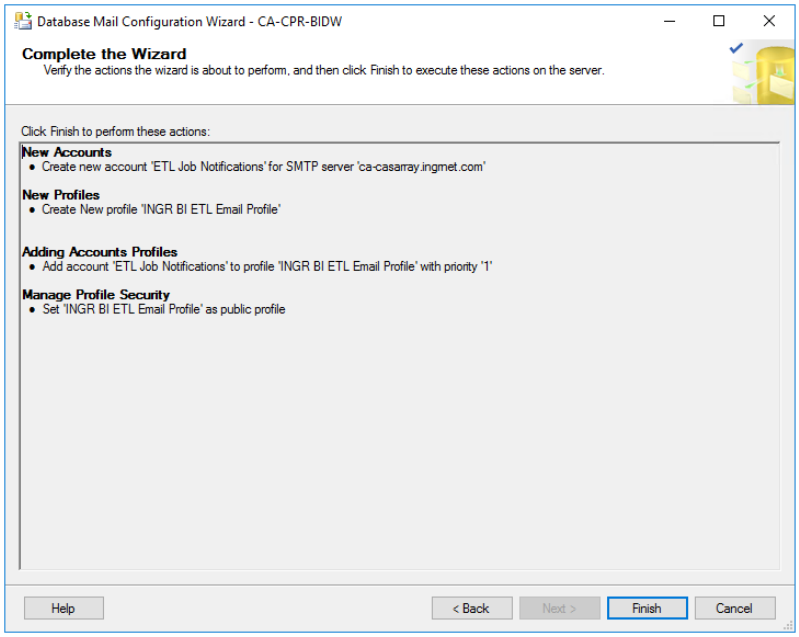
15. On the New Profile screen, select **Next**.
16. On the Manage Profile Security screen, check the **Public** box next to the newly created profile.



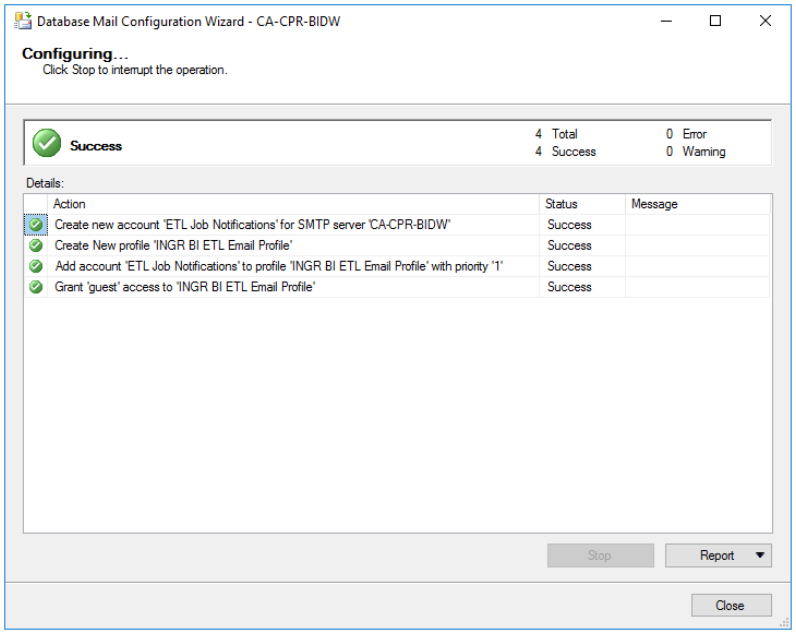
17. Select **Next**. On the Configure System Parameters screen, review the details.



18. Select **Next**.
19. On the Complete the Wizard screen, review the details and select **Finish**.



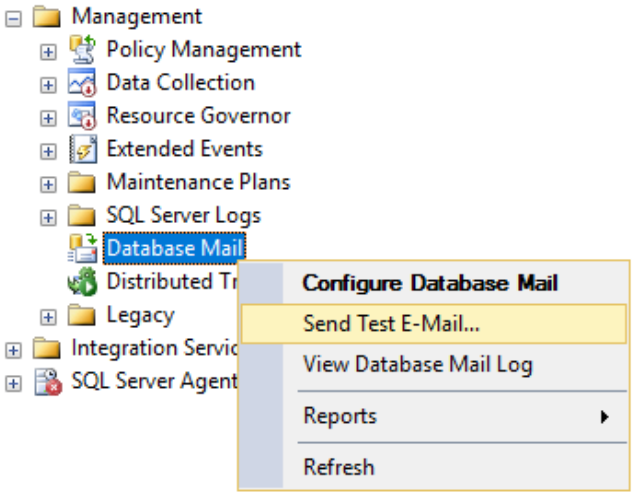
20. Once completed, ensure it was successful.



21. Select **Close**.

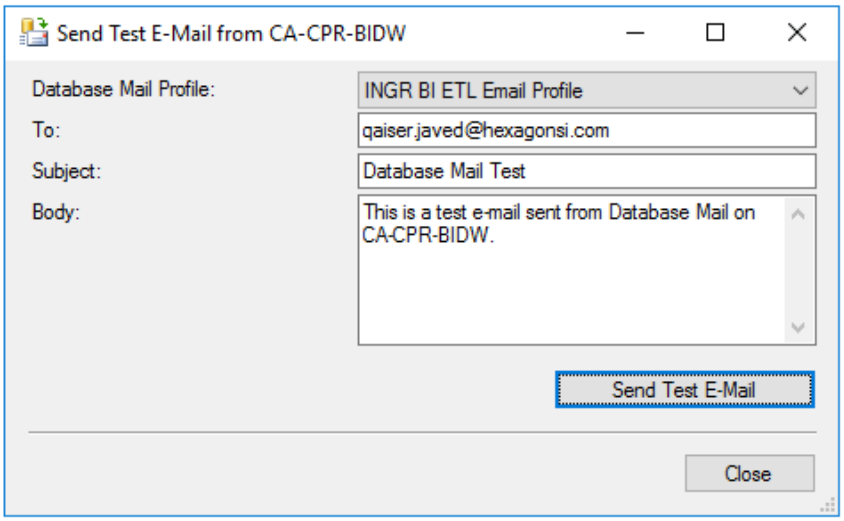
SEND TEST EMAIL

1. Open **SQL Server Management Studio**.
2. Create a connection to the Database Engine.
3. Navigate to **Management > Database Mail.**
4. Right-click on **Database Mail** and select **Send Test E-mail.**

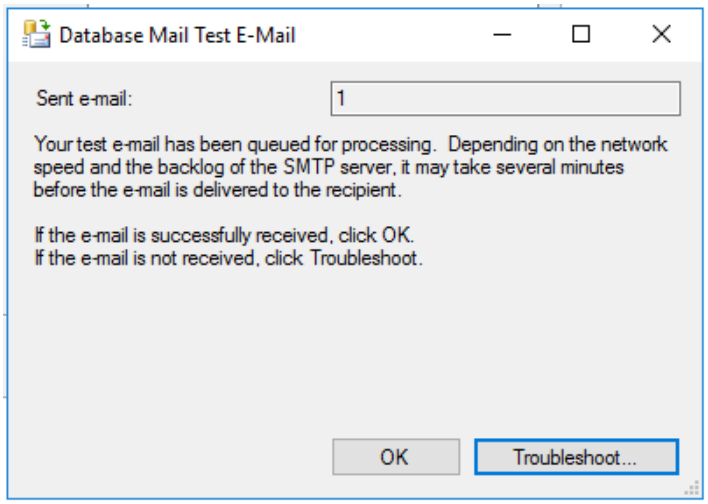


5. In the Send Test E-mail screen, select **INGR BI ETL Email Profile** from the drop-down to populate the **Database Mail Profile** field.
6. In the Send Test E-mail screen, enter a valid email address into the **To** field.

<INSERT NOTE IMAGE HERE> Multiple email addresses can be entered in this field.



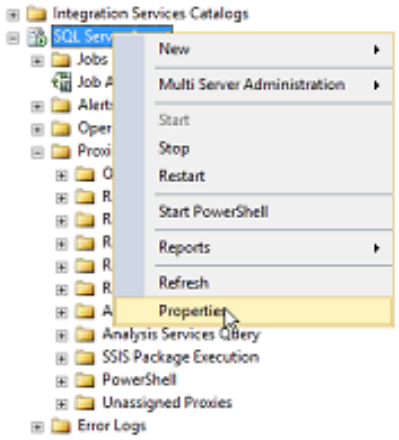
7. Select **Send Test E-Mail**. An email is sent to the address provided.
8. If you received the email, then click **OK** on the **Database Mail Test Email** screen. Otherwise, retrace all the steps and look for any mistakes.



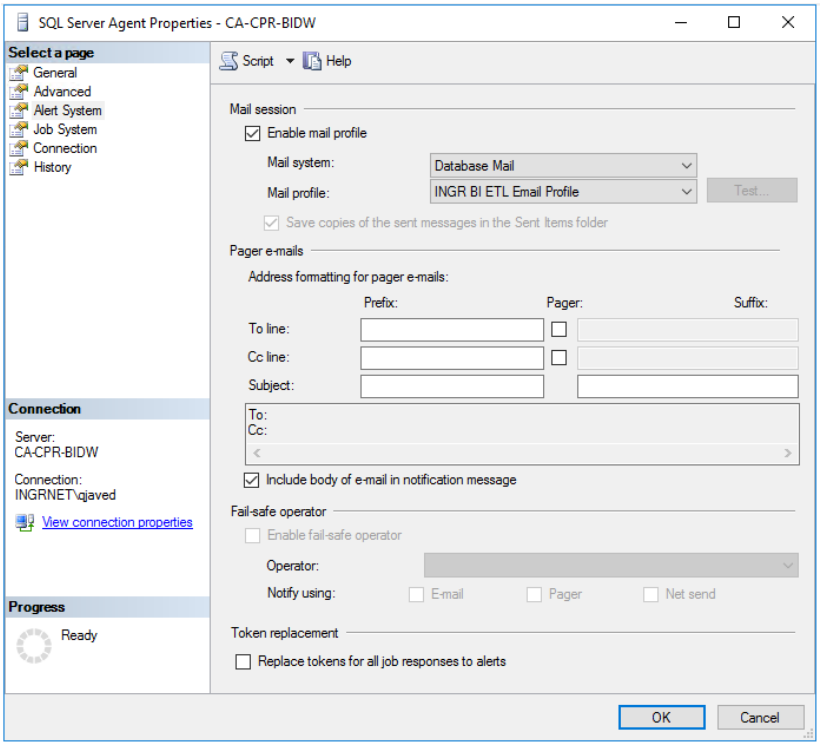
9. Select **OK**.

ENABLE SQL SERVER AGENT ALERT SYSTEM

1. Open **SQL Server Management Studio**.
2. Create a connection to the Database Engine.
3. Navigate to **SQL Server Agent.**
4. Right-click on **SLQ Server Agent** and select **Properties**. This opens the SQL Server Agent Properties.



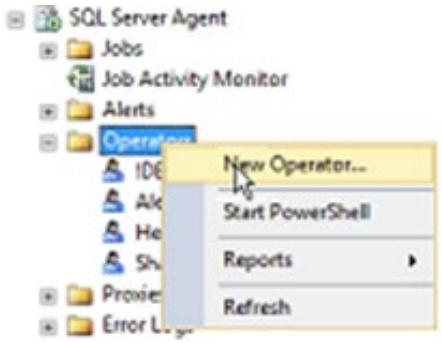
5. On the SQL Server Agent Properties window, select the **Alert System** tab on the left.
6. Check **Enable mail profile**.
7. For **Mail Profile**, select the **INGR BI ETL Email Profile** that was just created.



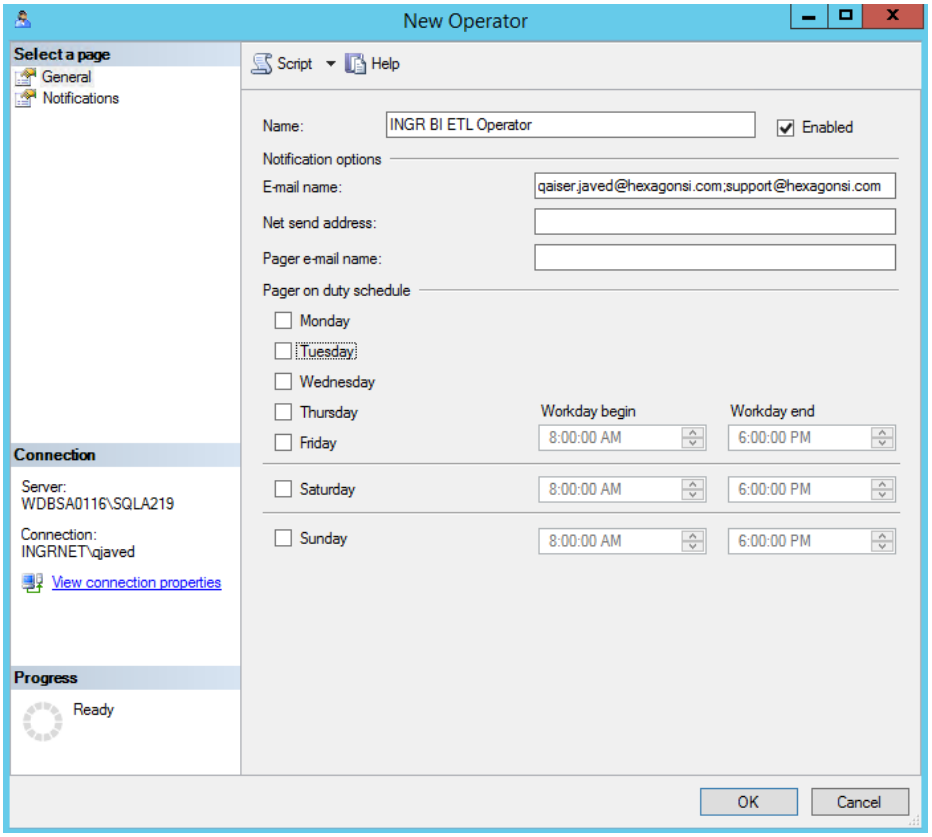
8. Leave everything else as is.
9. Select **OK**.

CREATE AN OPERATOR

1. Open **SQL Server Management Studio**.
2. Create a connection to the Database Engine.
3. Navigate to **SQL Server Agent > Operators.**



4. Right-click on **Operators** and select **New Operator**.
5. On the **New Operator** form, enter **INGR BI ETL Operator** for the **Name**.
6. On the **New Operator** form, provide an email address that receives the ETL job notifications when the job completes. Emails can be separated using a semi-colon (;).



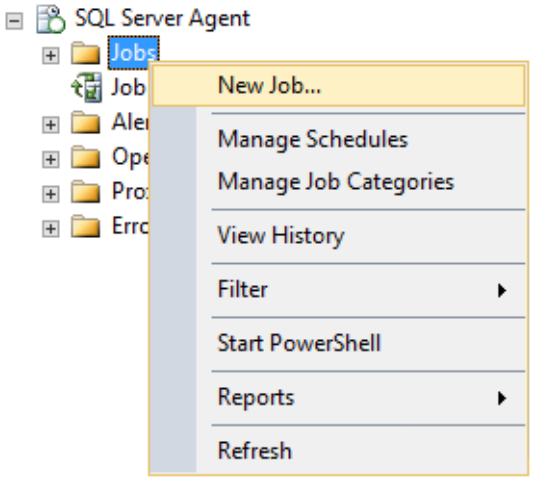
7. Click **OK**.
8. Verify if the new operator is created.

SETUP SQL SERVER AGENT JOB FOR INGR_BI_CAD

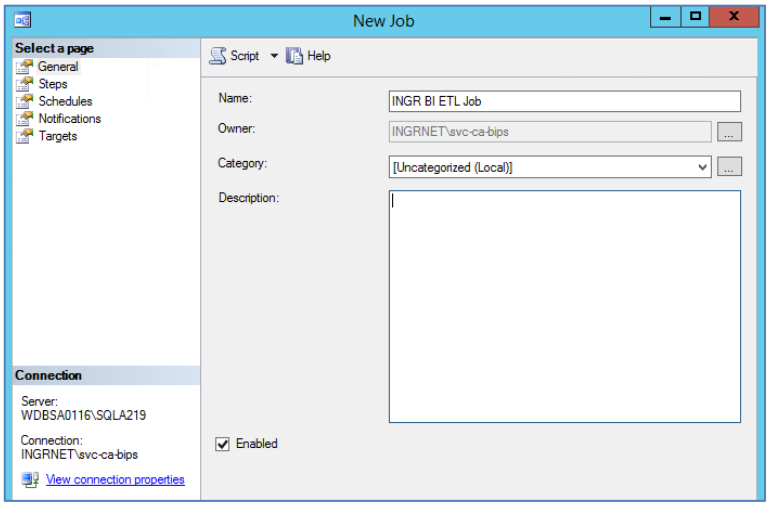
This section is to create a job that reads the ETL and runs it. The job should be created under the ETL user that was just created to run this job.

1. Open another instance of SQL Server Management Studio, using the credentials of the service account created earlier. For rxample: INGRENT\svc-ca-bips. This can be done by right-clicking on **Microsoft SQL Server Management Studio** and selecting **Run as different user**.

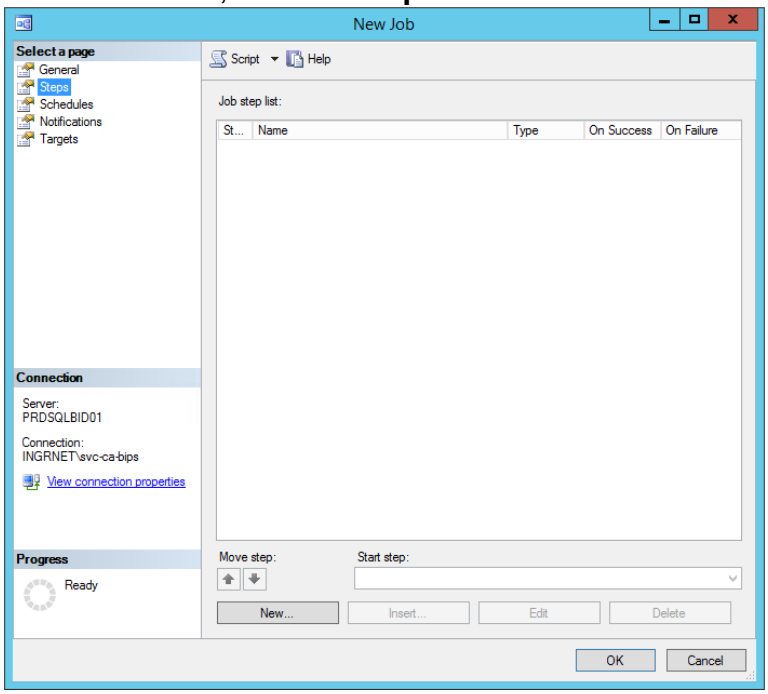
2. Open a connection to the Database Engine.
3. Navigate to **SQL Server Agent > Jobs**.
4. Right-click on **Jobs** and select **New Job**. This activates the **New Job** window.



5. In the New Job window, from the **General** tab, type **INGR BI ETL Job** for the **Name**.



6. On the left side, click on **Steps**.

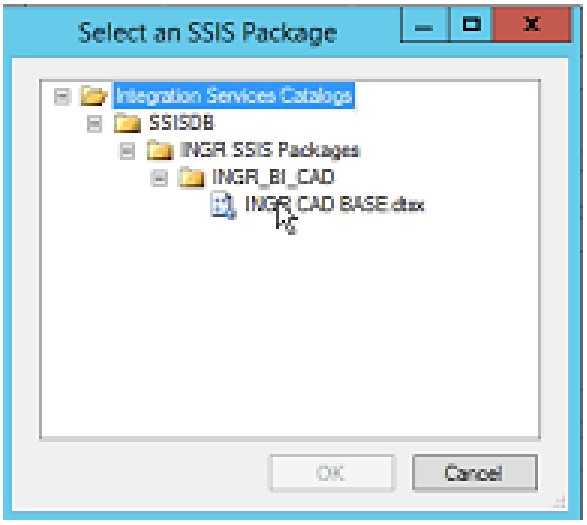


7. At the bottom of the page, select **New**. This activates the New Job Step window.
8. In the New Job Step window, enter the following:
   - Step Name = **INGR CAD BASE**
   - Type = **SQL Server Integration Services Package**
   - Run as = **INGR BI ETL Proxy**

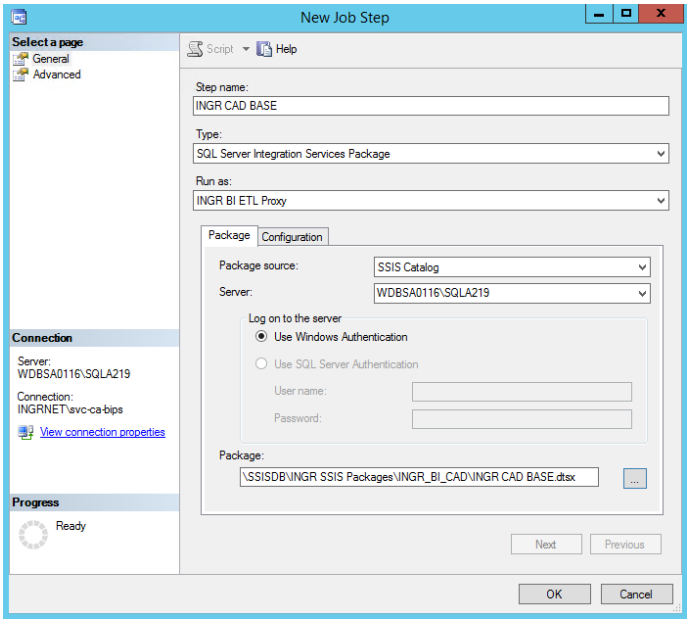
- Package
- Package source = **SSIS Catalog**
- Server = **<database server name>**

9. Check **Use Windows Authentication**.

   Navigate to**\SSISDB\INGR SSIS Packages\INGR_BI_CAD\INGR_CAD_BASE.dtsx.**



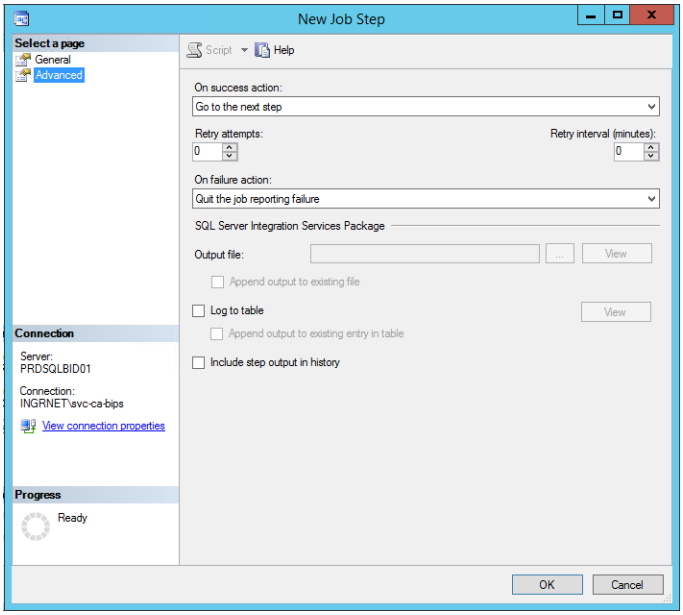
10. Verify the details on the New Job Step screen.



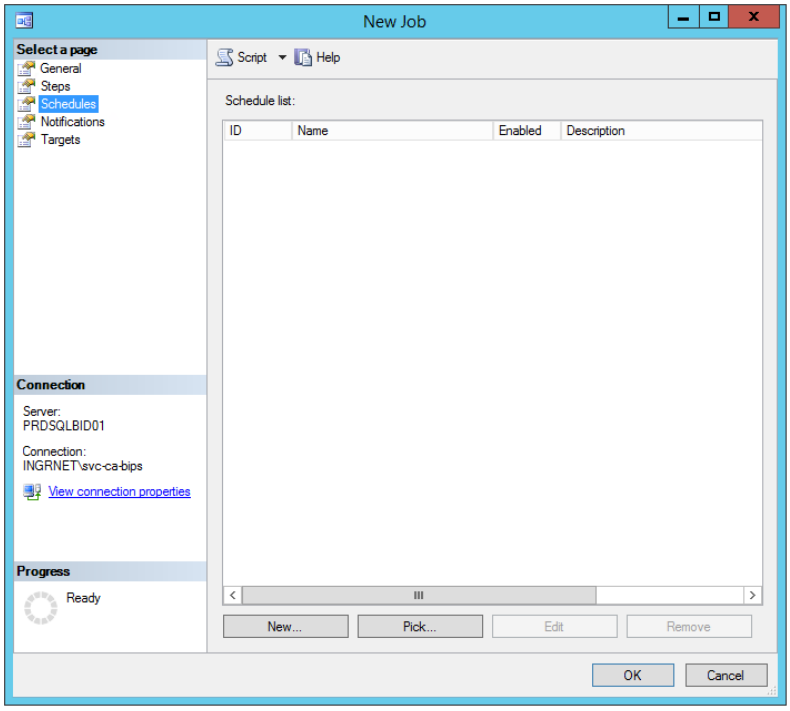
11.

12. In the middle of the page, select the **Configuration** tab and confirm that all parameters have a value under **Parameters** tab. If not, then verify that the INGR_BI_CAD package had been configured in the previous section.

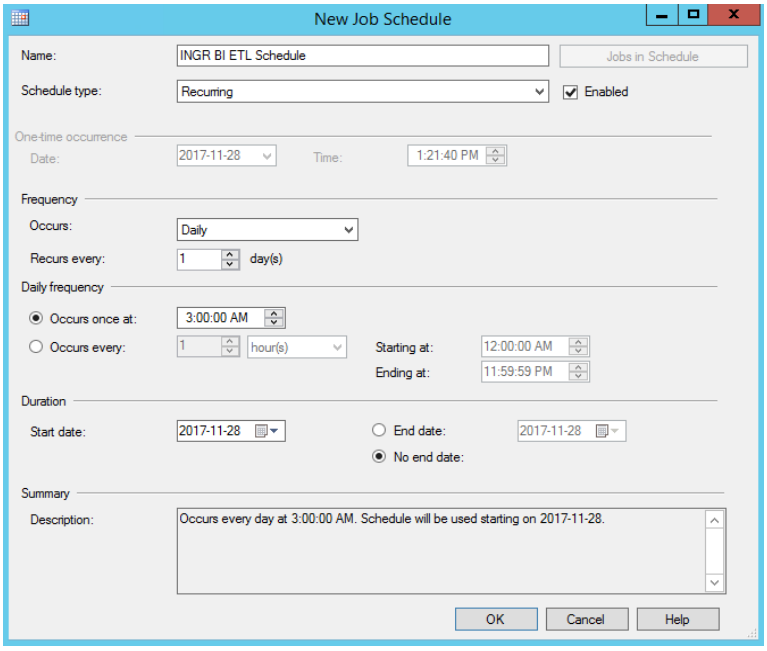
13. On the left side select the **Advanced** tab. Keep all default options.



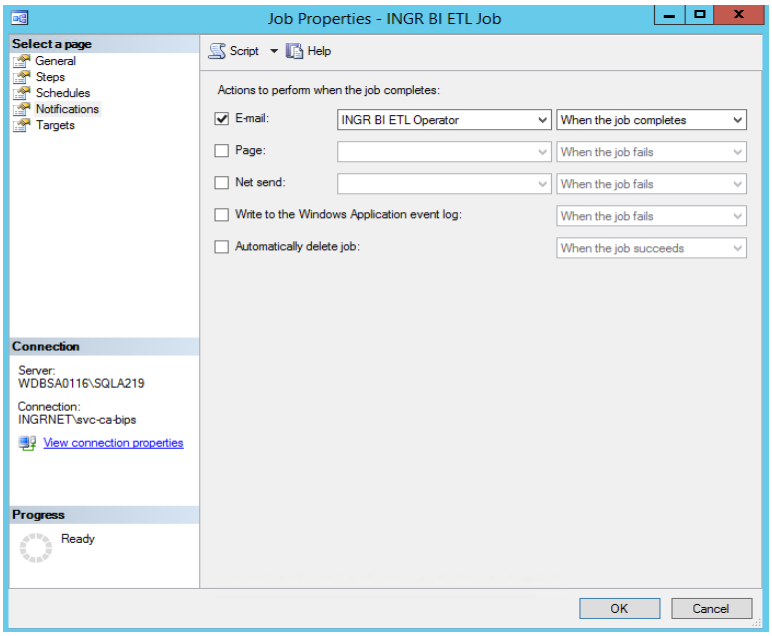
14. Click **OK**.
15. Back on the New Job screen, select the **Schedules** tab.
16.



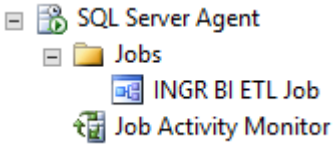
17. At the bottom of the window, select **New**.
18. For the New Job Schedule, enter the following:
    - Name = INGR BI ETL Schedule
    - Schedule Type = Recurring
    - Occurs = Daily
    - Occurs once at = 3:00am
    - Check the Enabled box



19. Select **OK**.
20. Back on the new job screen, select the **Notifications** tab.
21. Check the **E-mail** option and then from the drop-down select **INGR BI ETL Operator**. Set the option **When the job completes**.



22. Click **OK**.
23. To verify, you should see INGR BI ETL Job under **SQL Server Agent > Jobs.**



SETUP SQL SERVER AGENT JOB FOR INGR_AVL

In this section, INGR_AVL ETL is added to the existing INGR BI ETL job. The job should be created under the ETL user that was created to run this job.

When the first job, INGR CAD BASE, is run,, and it is successful, it runs the second job which is INGR AVL. Once the second job completes, exit the job.

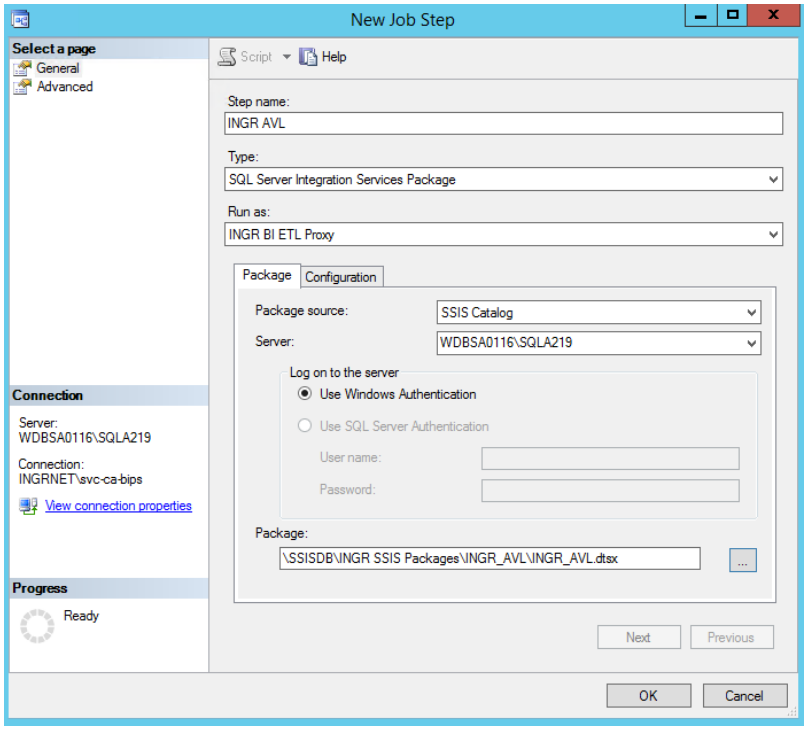
1. Right-click on **INGR BI ETL Job**, which was created in the previous section, and select **Properties**. This opens the **Job Properties** screen.
2. On the left side, click on **Steps**.
3. At the bottom of the page, select **New**. This opens the New Job Step window.
4. In the New Job Step window, enter the following:
   - Step Name = **INGR AVL**
   - Type = **SQL Server Integration Services Package**
   - Run as = **INGR BI ETL Proxy**
   - Package

     Package source = **SSIS Catalog**

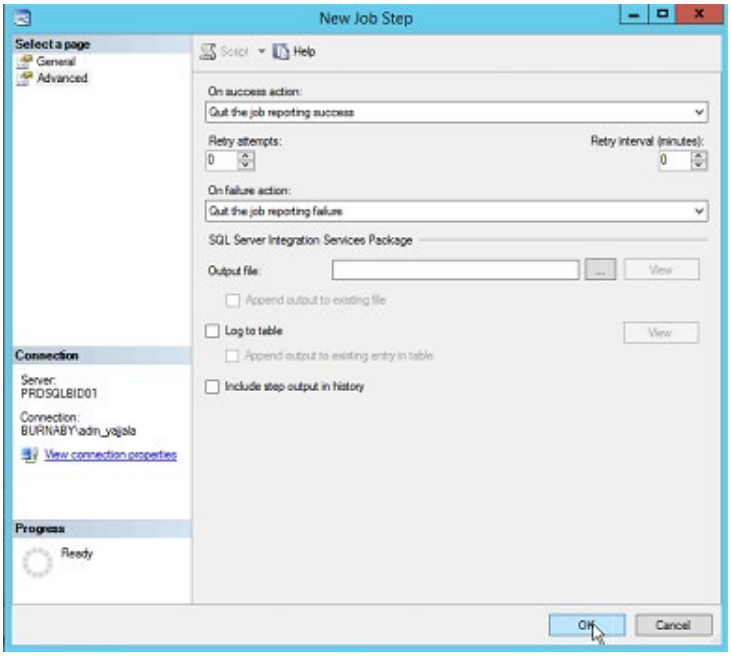
     Server = **<database server name>**

     Check **Use Windows Authentication**

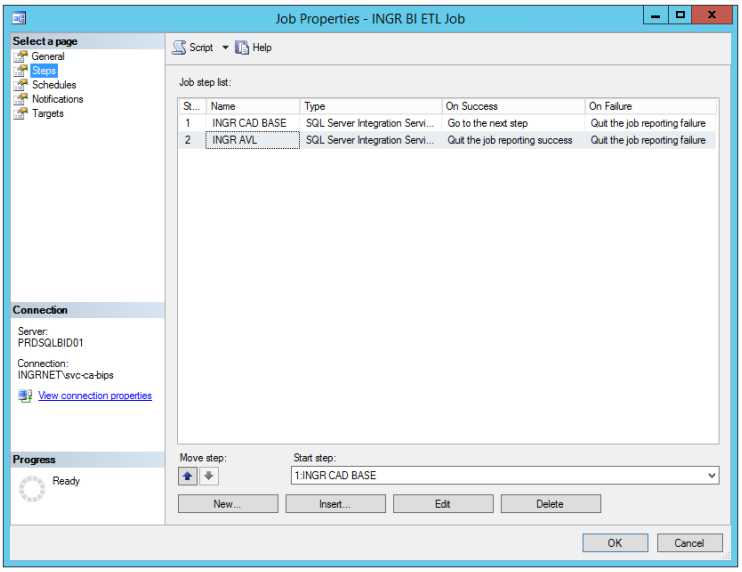
     Navigate to
     \SSISDB\INGR SSIS Packages\INGR_BI_CAD\INGR_AVL.dtsx.



5. Verify the details on the New Job Step screen.
6. In the middle of the page, select the **Configuration** tab and confirm that all parameters have a value under **Parameters** tab. If not, then verify that the INGR_AVL package had been configured in the previous section.
7. On the left side select the **Advanced** tab.
8. Change the following:
   - On Success Action to = Quit the job reporting success
   - Retry attempts to = 0
     Retry interval (minutes) to = 0
   - On failure action to = Quit the job reporting failure

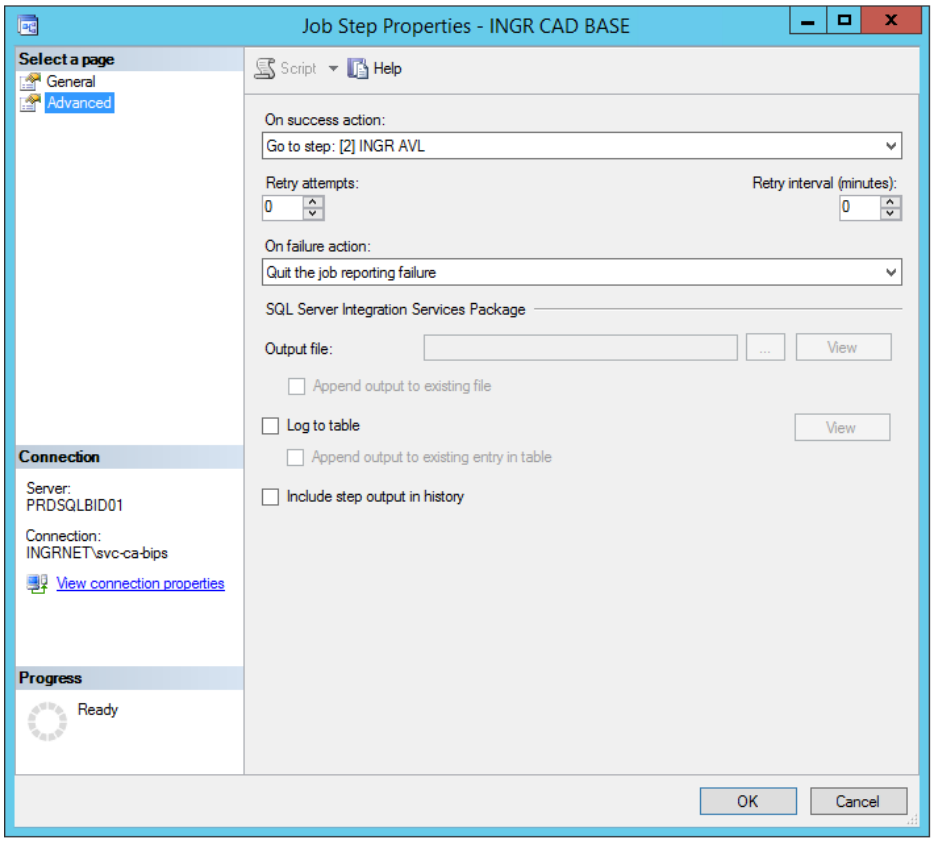


9. Click **OK**. You now see the second step in this job for AVL.



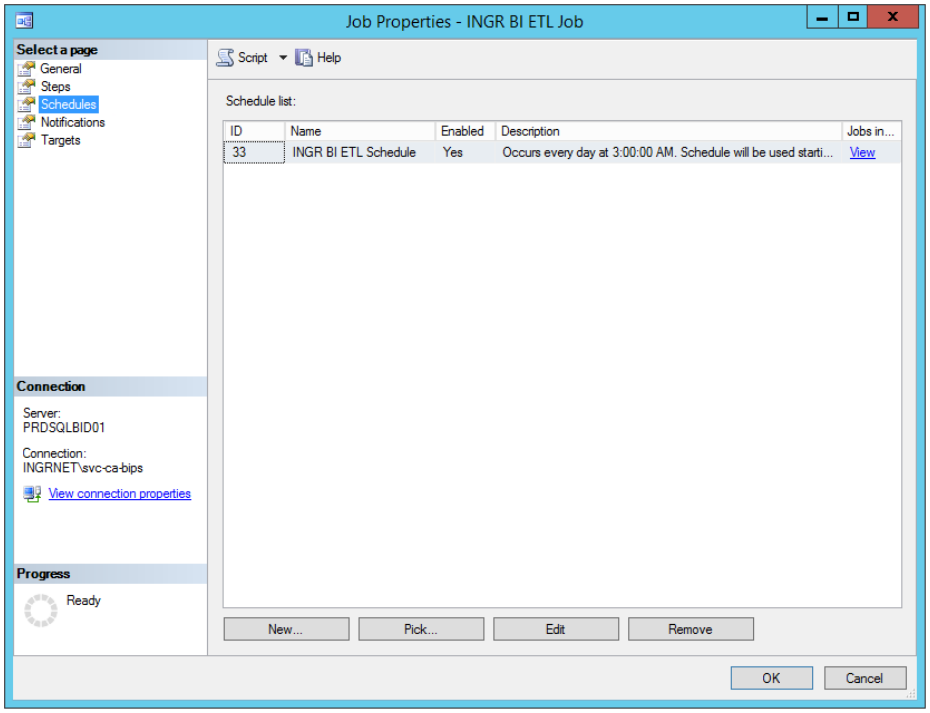
Since the second step has been added, you need to change our INGR CAD BASE step to recognize this new step.

10. For the Job Step List, double-click **INGR CAD BASE** to open the **Properties** screen.
11. On the left side select the **Advanced** tab.
12. On this screen, change the following:
    - On Success Action to = Go to step [2] INGR AVL
    - Retry attempts to = 0
      Retry interval (minutes) to = 0
    - On failure action to = Quit the job reporting failure



13. On the Job Step Properties screen, select **OK**. This returns you to the Job Properties screen.

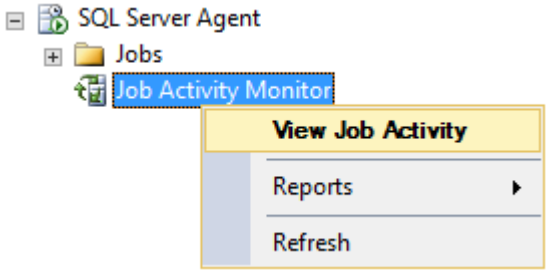
14. Back on the new job screen, select the **Schedules** tab. You see that the job has already been scheduled, as per the previous section.



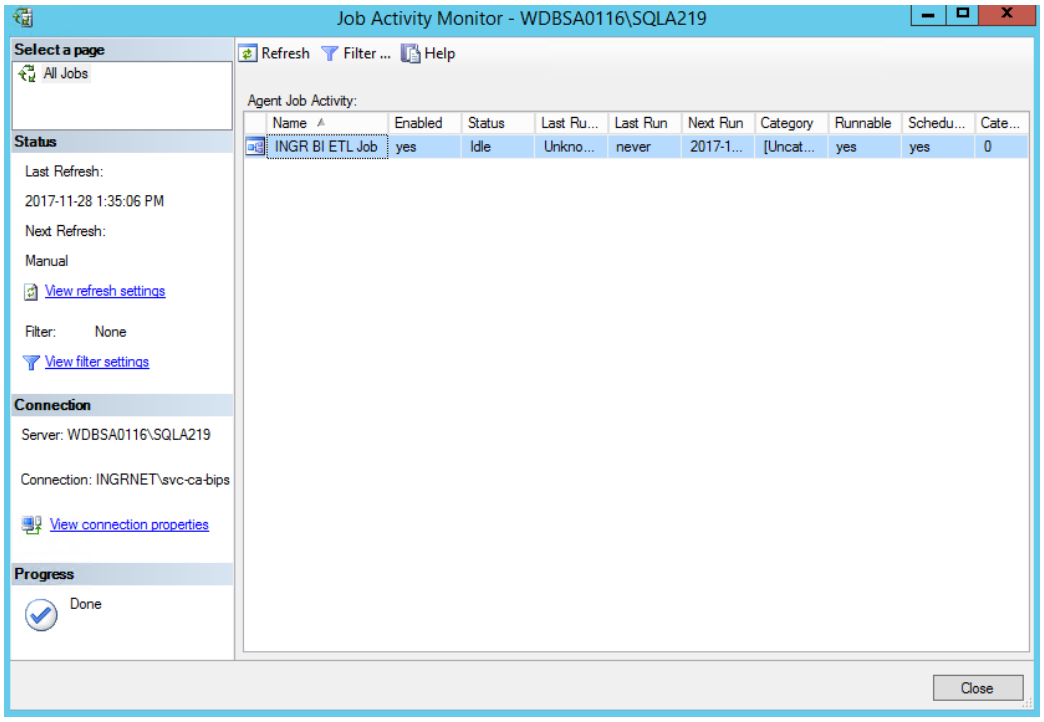
15. Ensure that the job is enabled.

MONITOR JOB STATUS

1. Open **SQL Server Management Studio.**
2. Connect to the Database Engine.
3. Navigate to **SQL Server Agent > Jobs > Job Activity Monitor**.
4. Right-click on **Job Activity Monitor** and select **View Job Activity**.



5. Note the status of the jobs in the Status column.



You now have history created in this table, as the jobs run each day.

EXECUTION OF ETL PACKAGES

The ETL can be run using three methods:

- Directly from the Visual Studio 2015
- Using Integration Services Catalogs through SQL Server Management Studio
- Using SQL Server Agent Jobs through SQL Server Management Studio

The initial ETL runs using SQL Server Agent. The job can be run remotely from any machine by anyone who has access to SQL Server Management Studio.

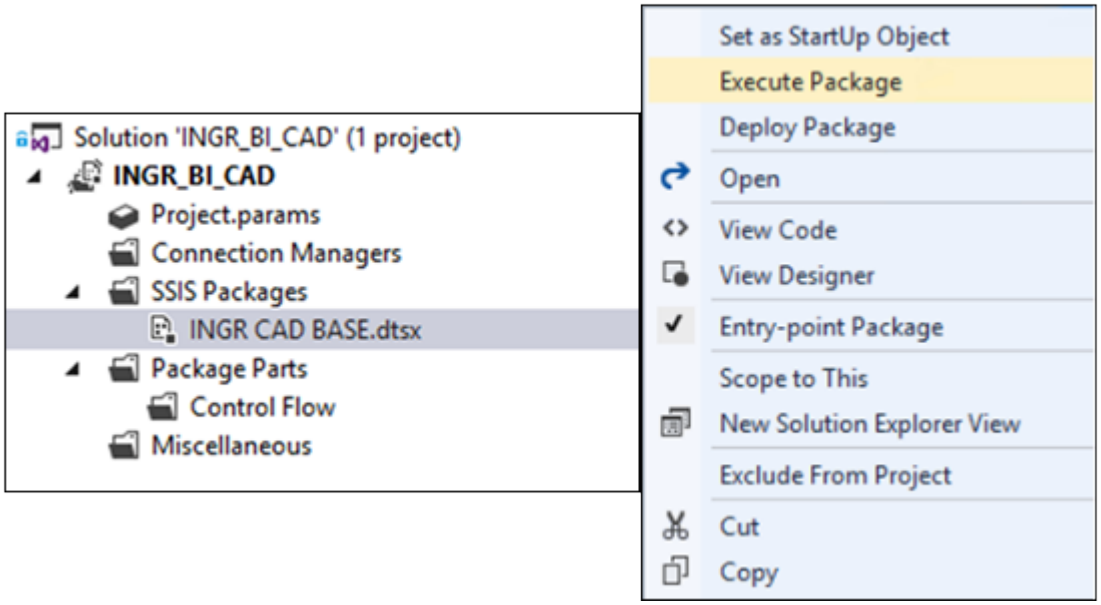
The job to run the ETL packages is not installed by the Intergraph DW\ETL installer. The steps to create a SQL Serve job are described in the InSightAdvantageforICAD_Installation_Guide.pdf.

After a successful initial ETL run, an administrator can configure and schedule a job to run nightly at a specific time for incremental data.

RUNNING THE ETL USING VISUAL STUDIO 2015

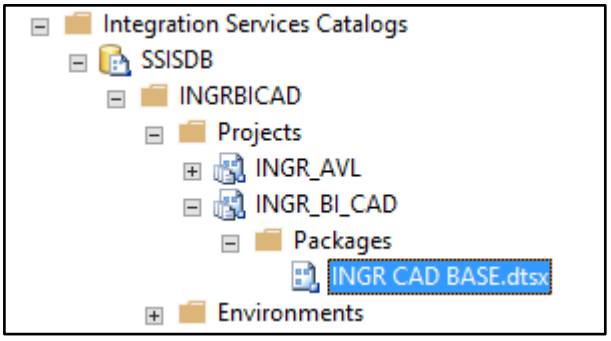
This requires a remote login to the ETL machine as a local administrator.

1. Launch Visual Studio 2015 runtime and open the Project/Solution INGR_BI_CAD.sln file.
2. Once the solution is open, navigate to **Project > INGR_BI_CAD Properties**.
3. Select **Debugging > Run64BitRuntime**.
4. Change the value to **True**.
5. Expand **INGR_BI_CAD > SSIS Packages**.
6. Right-click and select **INGR CAD BASE.dtsx > Execute Package**.

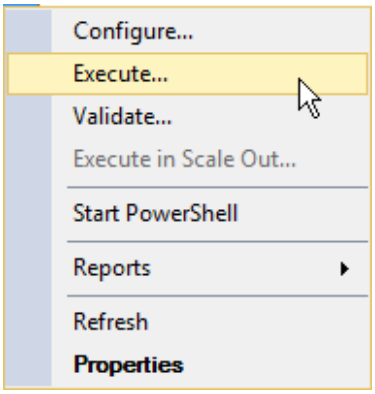

Solution 'INGR_BI_CAD' (1 project)
INGR_BI_CAD
Project.params
Connection Managers
SSIS Packages
INGR CAD BASE.dtsx
Package Parts
Control Flow
Miscellaneous
Set as StartUp Object
Execute Package
Deploy Package
Open
View Code
View Designer
Entry-point Package
Scope to This
New Solution Explorer View
Exclude From Project
Cut
Copy

RUNNING THE ETL USING INTEGRATION SERVICES CATALOGS

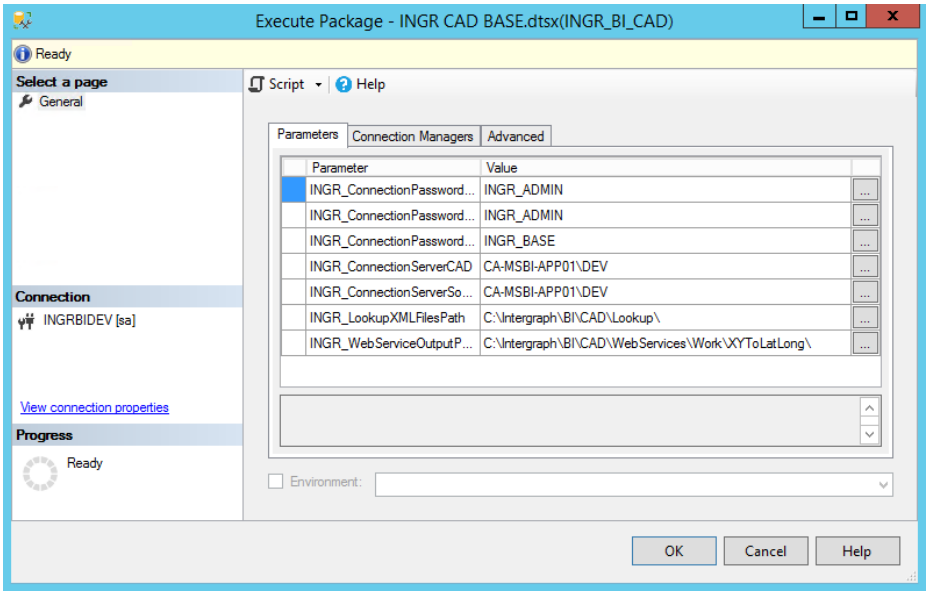
1. Launch **SQL Server Management Studio** and connect to the BI Data Warehouse server.
2. Expand **Integration Services Catalogs > SSISDB > INRGBICAD > Projects**
3. Locate the INGR CAD BASE.dtsx package or the INGR AVL.dtsx package.



4. To run the package, right click and select **Execute**.

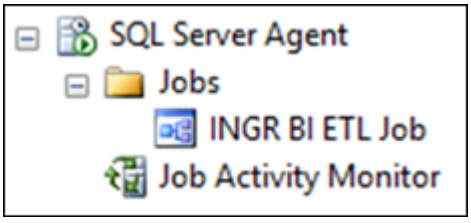


5. Click **OK** on the next window (Execute Package).

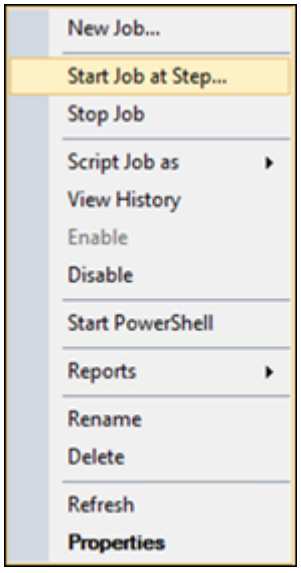


RUNNING THE ETL USING SQL SERVER AGENT

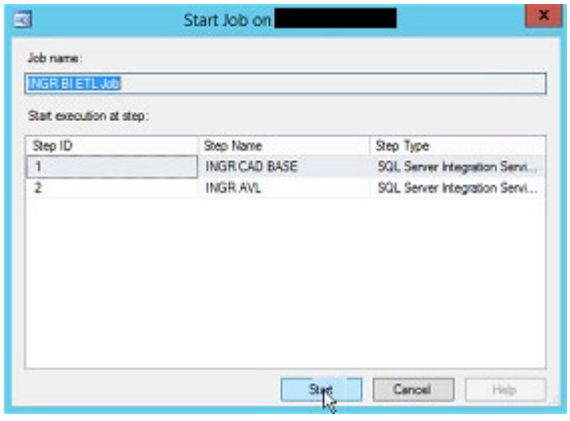
1. Launch **SQL Server Management Studio** and connect to the BI Data Warehouse server.
2. Expand **SQL Server Agent > Jobs**



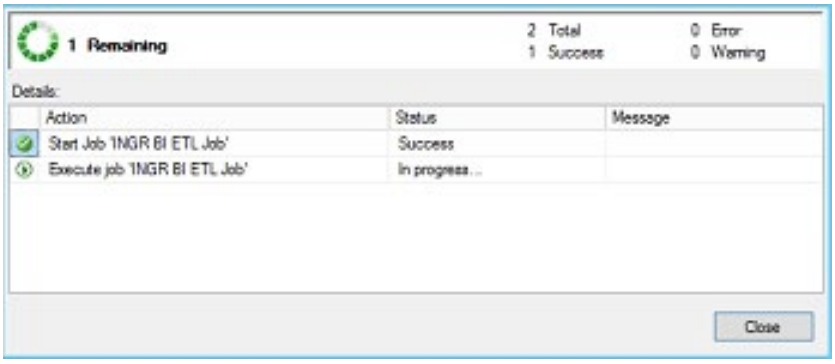
3. Locate the **INGR BI ETL Job**.
4. To run the job, right-click and select **Start Job at Step**.



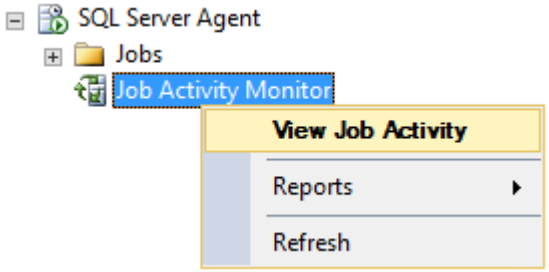
5. If there is only one step, the job begins. However, if there is a second step, for INGR AVL, you are prompted with a list of steps. Select Step 1.



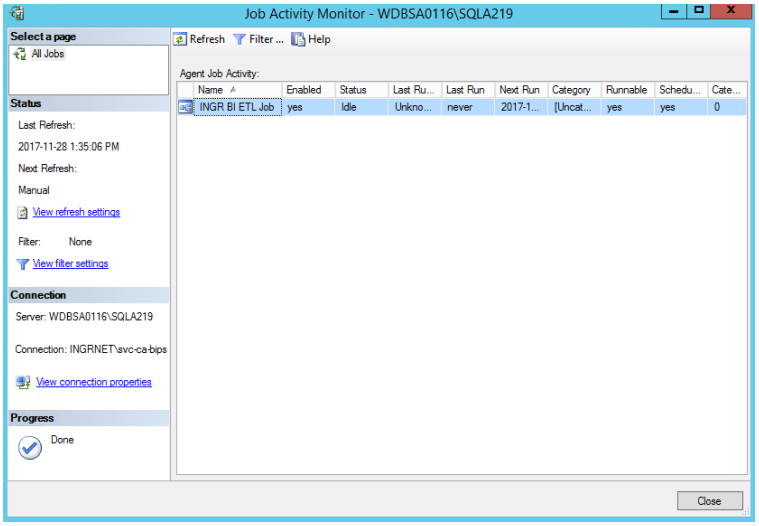
6. The job starts processing.



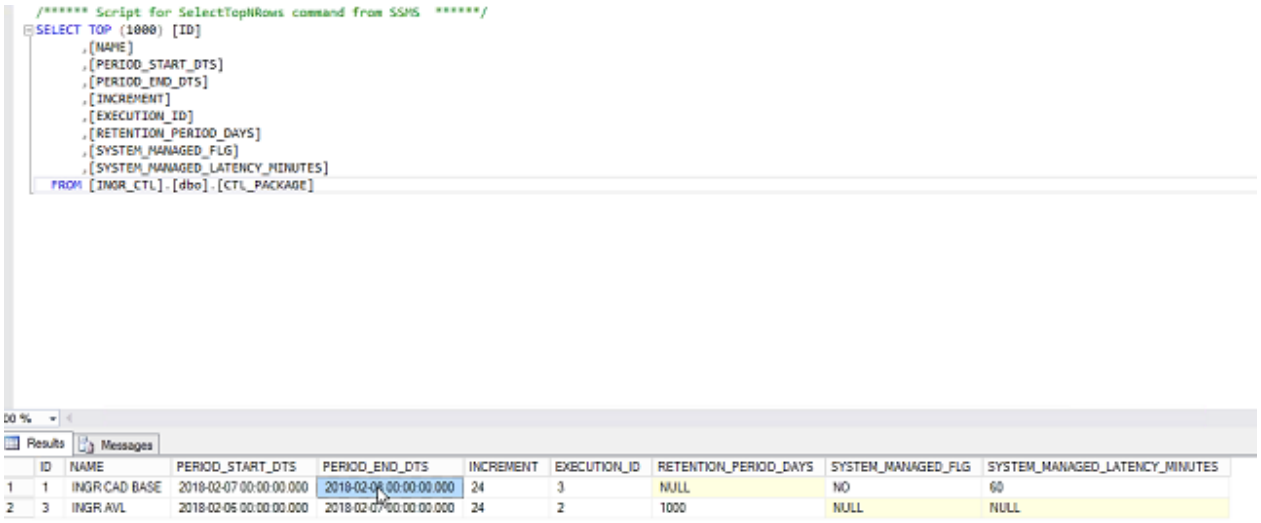
7. To see what is happening, right-click on **Job Activity Monitor** and select **View Job Activity**.



8. Note the status of the jobs in the Status column.



9. To see the change of the dates for the incremental ETL, open the **INGR_CTL** database and select all records from the **CTL_PACKAGE** table. The PERIOD_START_DTS and PERIOD_END_DTS tables tell you the date period for which the ETL is run.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT TOP (1000) [ID]
      ,[NAME]
      ,[PERIOD_START_DTS]
      ,[PERIOD_END_DTS]
      ,[INCREMENT]
      ,[EXECUTION_ID]
      ,[RETENTION_PERIOD_DAYS]
      ,[SYSTEM_MANAGED_FLG]
      ,[SYSTEM_MANAGED_LATENCY_MINUTES]
  FROM [INGR_CTL].[dbo].[CTL_PACKAGE]
```

Results | Messages

	ID	NAME	PERIOD_START_DTS	PERIOD_END_DTS	INCREMENT	EXECUTION_ID	RETENTION_PERIOD_DAYS	SYSTEM_MANAGED_FLG	SYSTEM_MANAGED_LATENCY_MINUTES
1	1	INGR CAD BASE	2018-02-07 00:00:00.000	2018-02-08 00:00:00.000	24	3	NULL	NO	60
2	3	INGR AVL	2018-02-05 00:00:00.000	2018-02-07 00:00:00.000	24	2	1000	NULL	NULL

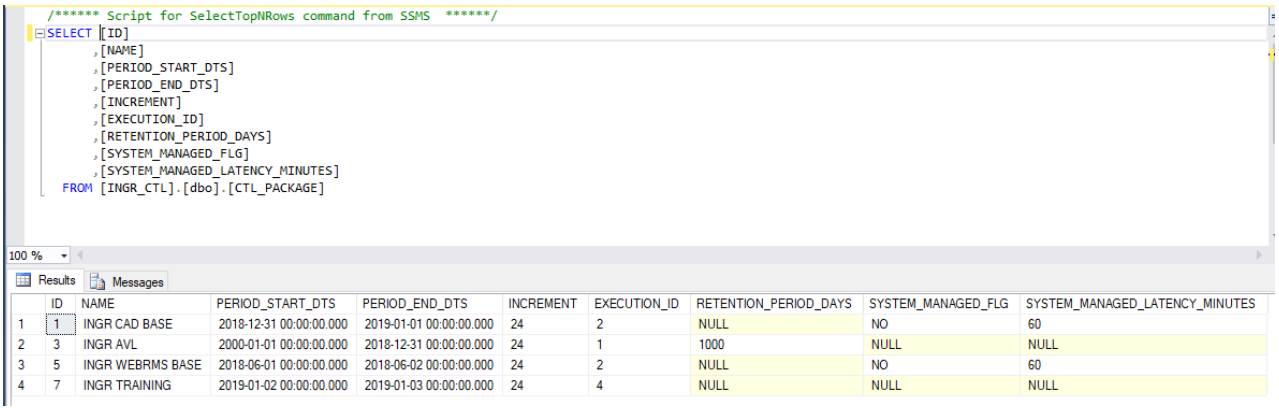
INCREMENTAL ETL

There are two options to control the date range when running the ETL. The first option is to define a start date and end date. The second option is to define a start date and leave the end date to be managed by the system. With this option the end date is equal to the date and time when the ETL starts minus a defined latency. The latency ensures all data has been written to the source before the ETL is run; for example, if the agency is using DB Copy to populate the source database, the latency defined must be greater than the latency introduced by DB Copy.

These configuration parameters are defined in the CTL_PACKAGE table in the INGR_CTL database. This table has one record for each SSIS package. The parameters are stored in columns in this table. Some of these columns are described below:

PERIOD_START_DTS – Start date and time.
PERIOD_END_DTS – End date and time (is not used if the SYSTEM_MANAGED_FLG attribute is equal to YES).
RETENTION_PERIOD_DAY – Attribute used by the AVL package to define the amount

of data (in days) maintained in the database.
SYSTEM_MANAGED_FLG – Attribute used by CAD and RMS SSIS packages. If SYSTEM_MANAGED_FLG = YES, then the PERIOD_END_DTS is defined as the date and time when the package started minus the latency.
SYSTEM_MANAGED_LATENCY_IN_MINUTES – Attribute used by CAD and RMS SSIS packages to define the latency in minutes to define the PERIOD_END_DTS.



```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT [ID]
      ,[NAME]
      ,[PERIOD_START_DTS]
      ,[PERIOD_END_DTS]
      ,[INCREMENT]
      ,[EXECUTION_ID]
      ,[RETENTION_PERIOD_DAYS]
      ,[SYSTEM_MANAGED_FLG]
      ,[SYSTEM_MANAGED_LATENCY_MINUTES]
  FROM [INGR_CTL].[dbo].[CTL_PACKAGE]
```

100 %

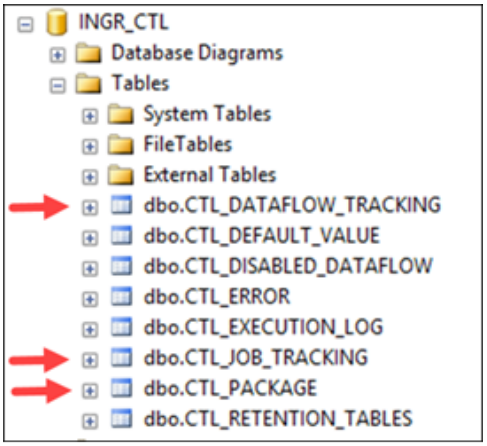
Results | Messages

	ID	NAME	PERIOD_START_DTS	PERIOD_END_DTS	INCREMENT	EXECUTION_ID	RETENTION_PERIOD_DAYS	SYSTEM_MANAGED_FLG	SYSTEM_MANAGED_LATENCY_MINUTES
1	1	INGR CAD BASE	2018-12-31 00:00:00.000	2019-01-01 00:00:00.000	24	2	NULL	NO	60
2	3	INGR AVL	2000-01-01 00:00:00.000	2018-12-31 00:00:00.000	24	1	1000	NULL	NULL
3	5	INGR WEBRMS BASE	2018-06-01 00:00:00.000	2018-06-02 00:00:00.000	24	2	NULL	NO	60
4	7	INGR TRAINING	2019-01-02 00:00:00.000	2019-01-03 00:00:00.000	24	4	NULL	NULL	NULL

MONITORING TABLES

The ETL execution is controlled and monitored using three tables located inside INGR_CTL database:

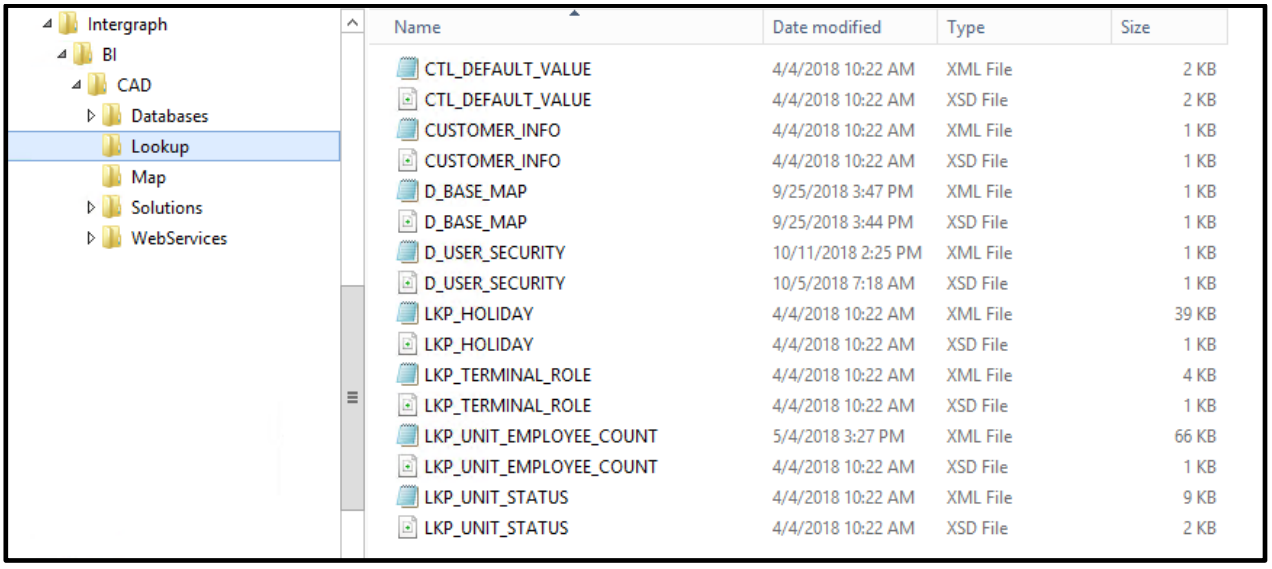
- CTL_PACKAGE: Controls the execution of an ETL by providing dates for data extraction and the value for next incremental run.
- CTL_JOB_TRACKING: Monitors the execution of an ETL at the job level.
- CTL_DATAFLOW_TRACKING: Monitors the execution of an ETL at the data-flow level.



LOOKUP SHEETS

The CAD ETL requires certain site-specific data, such as holidays, unit status codes, terminal roles, and unit employee count.

This data is made available using XML files named accordingly with the database tables. By default, these files are located at C:\Intergraph\BI\CAD\Lookup.



The definitions of each XML files are described below:

- **CTL_DEFAULT_VALUE:** Default values for different attribute types.
- **CUSTOMER_INFO**: Agency name and logo.
- **D_BASE_MAP:** Base map name and URL to the tile map server.
- **D_USER_SECURITY:** Group or users associated with a CAD agency.
- **LKP_HOLIDAY**: National and provincial holidays, which can include non-holiday special events such as Halloween and St. Patrick's Day.

  This lookup is used for parsing data based on a specific date or event.
- **LKP_TERMINAL_ROLE**: Default role of each terminal (for example, dispatcher, mobile, or call taker).
- **LKP_UNIT_EMPLOYEE_COUNT**: Default number of employees on a unit during an event.
- **LKP_UNIT_STATUS**: All unit status codes used by the CAD, including custom and overrides.

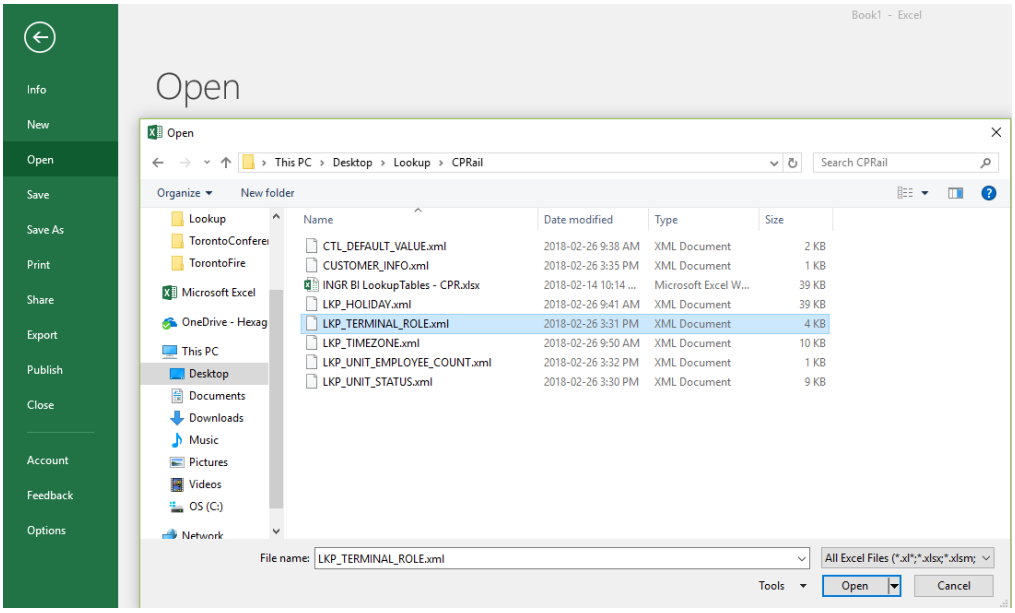
Part of the implementation process consists of populating these files with customer/site specific information. The XML files can be edited using an XML editor. Hexagon recommends using Microsoft Excel is recommended for this purpose.

Example: Editing LKP_TERMINAL_ROLE.xml file using a text editor.

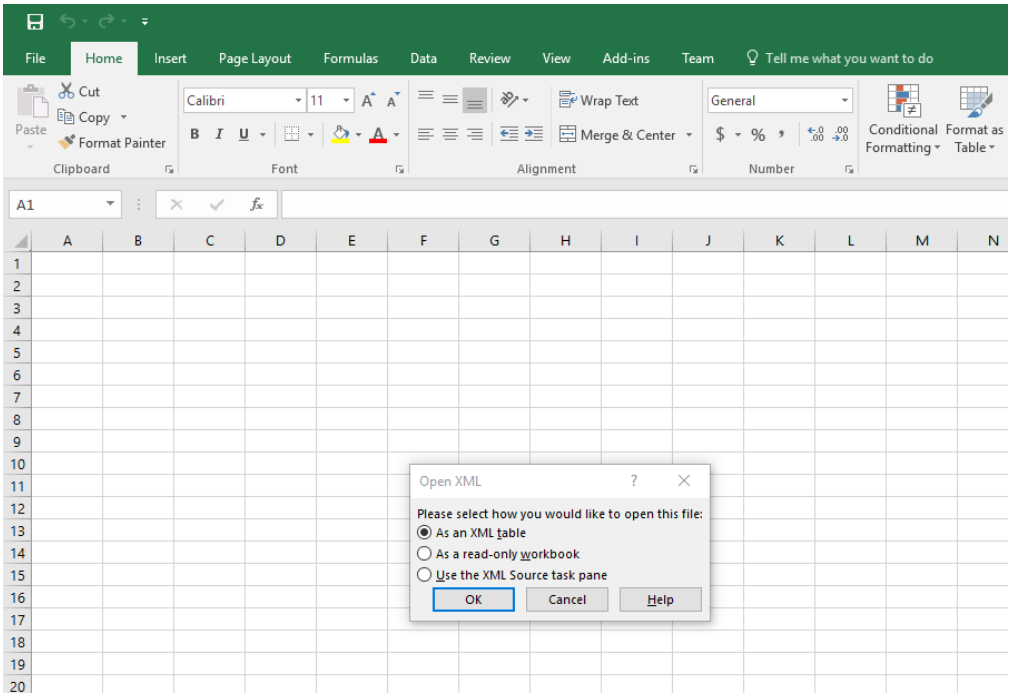
```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<lkpterminalrole xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance">
    <terminalroleData>
        <terminal_id>1001396</terminal_id>
        <role>User ID</role>
    </terminalroleData>
    <terminalroleData>
        <terminal_id>1002662</terminal_id>
        <role>User ID</role>
    </terminalroleData>
    <terminalroleData>
        <terminal_id>1003557</terminal_id>
        <role>User ID</role>
    </terminalroleData>
    <terminalroleData>
        <terminal_id>1004004</terminal_id>
        <role>User ID</role>
    </terminalroleData>
    <terminalroleData>
        <terminal_id>1q1t942</terminal_id>
        <role>Dispatcher</role>
    </terminalroleData>
    <terminalroleData>
        <terminal_id>1qhv942</terminal_id>
        <role>Not Used Anymore</role>
    </terminalroleData>
    <terminalroleData>
        <terminal_id>215</terminal_id>
        <role>User ID</role>
    </terminalroleData>
```

To populate the XML files with site specific data:

1. Open **Microsoft Excel** and navigate to the folder where the XML files are stored.
2. Choose an XML file and open.



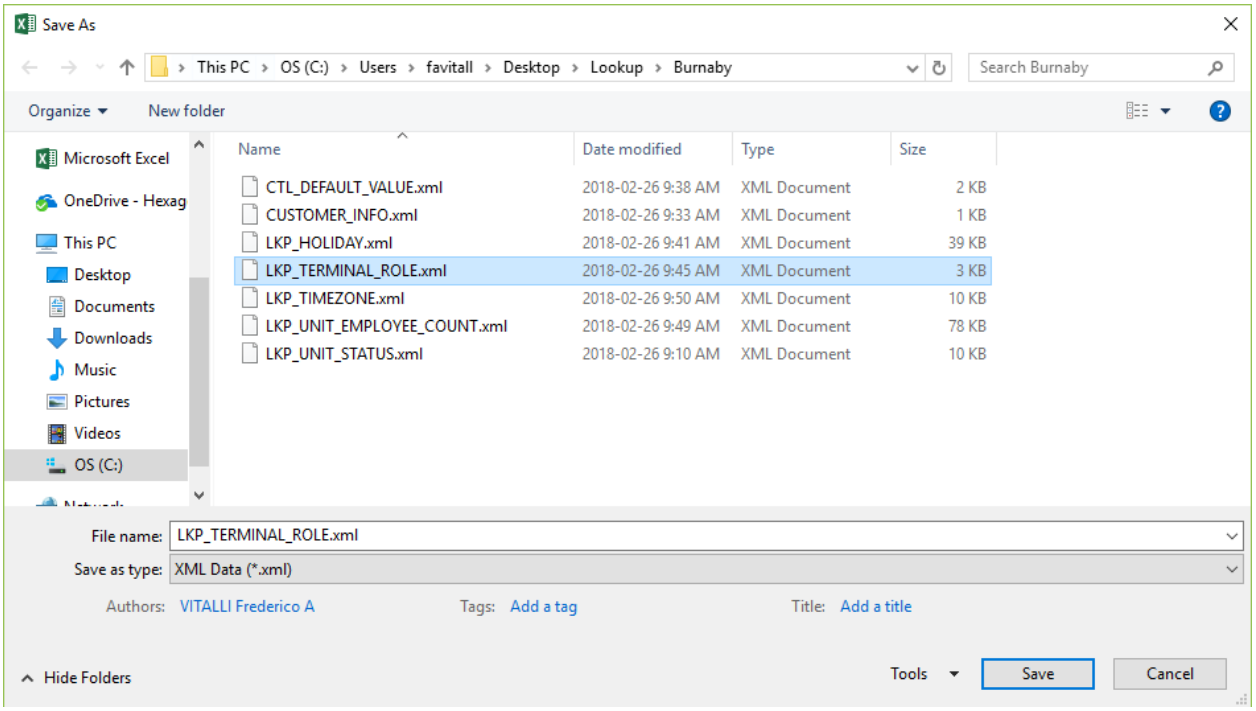
3. Select to open the files **As an XML table**.



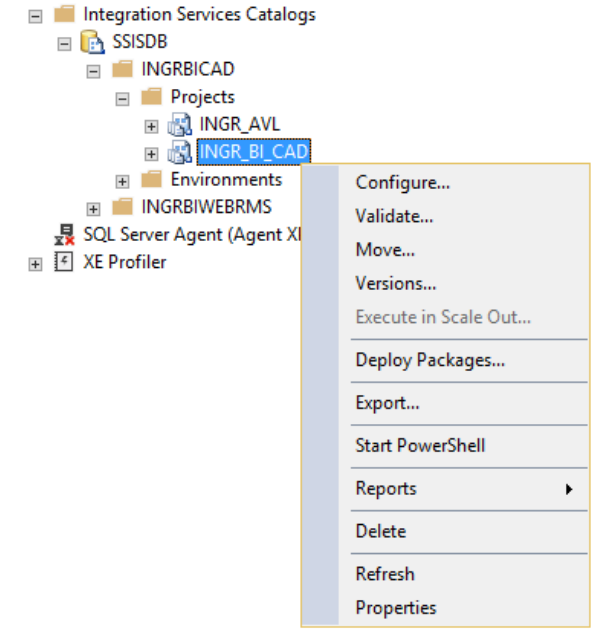
4. Edit the LKP_UNIT_STATUS.xml file, where necessary.

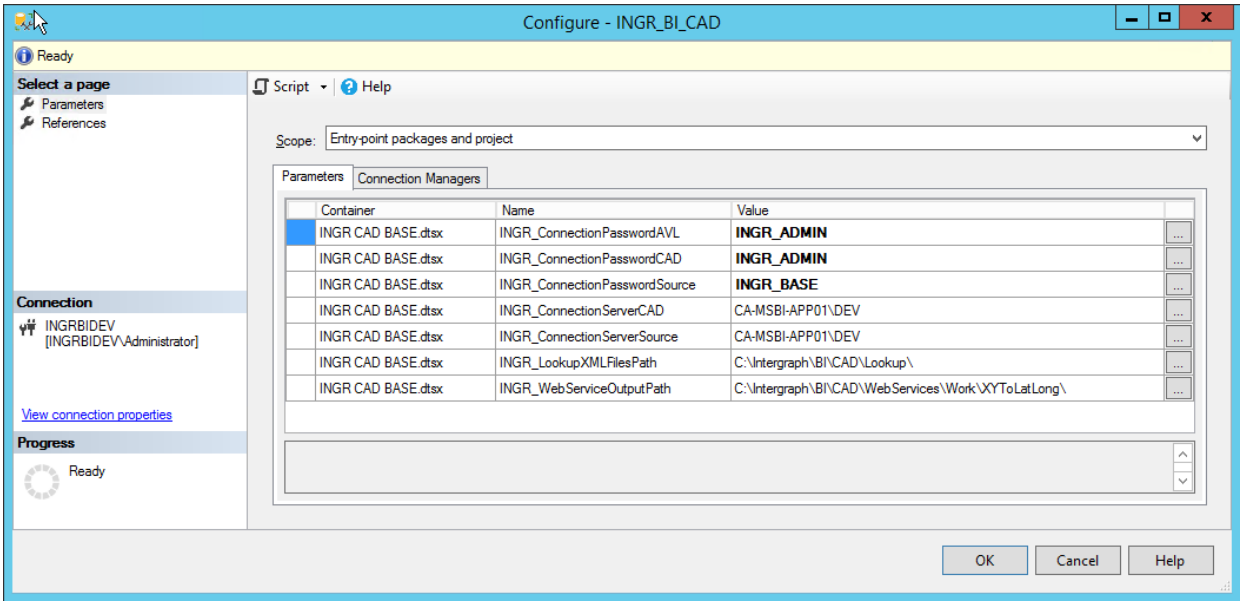
	A	B	C	D	E	F	G	H
1	unit_status_id	unit_status_cd	unit_status_desc	unit_status_class	unit_status_class_cd	unit_on_event_flg	configuration	unit_order
2	0	AV	Available	Available	AVL	No	default	1
3	1	AQ	Available at Quarters	Available	AVL	No	default	1
4	2	AP	Available on Pager	Available	AVL	No	default	1
5	3	AF	Available on Foot	Available	AVL	No	default	1
6	4	AM	Available on MDT	Available	AVL	No	default	1
7	5	EO	Available on emergency	Available	AVL	No	default	1
8	6	1R	Available First Response	Available	AVL	No	default	1
9	7	DP	Dispatched	Assigned	ASG	Yes	default	4
10	8	ER	Enroute	Assigned	ASG	Yes	default	6
11	9	AR	Arrived	Assigned	ASG	Yes	default	7
12	10	AD	Arrive Danger	Assigned	ASG	Yes	default	8
13	11	LA	Arrive Long	Assigned	ASG	Yes	default	9
14	12	OS	Out of Service	Out of Service	OOS	No	default	12
15	13	TR	Transport	Assigned	ASG	Yes	default	10
16	14	TA	Transport Arrive	Assigned	ASG	Yes	default	11
17	15	AK	Acknowledge	Assigned	ASG	Yes	default	5
18	16	GP	Group Member	Available	AVL	No	default	13
19	18	PA	Pending Available	Available	AVL	No	default	2
20	19	PM	Pending Mobile	Available	AVL	No	default	3
21	20	AN	Available On Neighborhood Patrol	Available	AVL	No	default	13
22	25	NR	Not Recommendable	Available	AVL	No	default	13
23	26	RE	Relocate Back	Available	AVL	No	default	13
24	90	EE	Enroute Event	Assigned	ASG	Yes	default	13
25								

5. Save your changes as XML data.



The default location for lookup XML files can be changed by updating the INGR_LookupXMLFilesPath parameter in INGR_BI_CAD solution. This task can be accomplished by right clicking on the project name and selecting **Configure**.

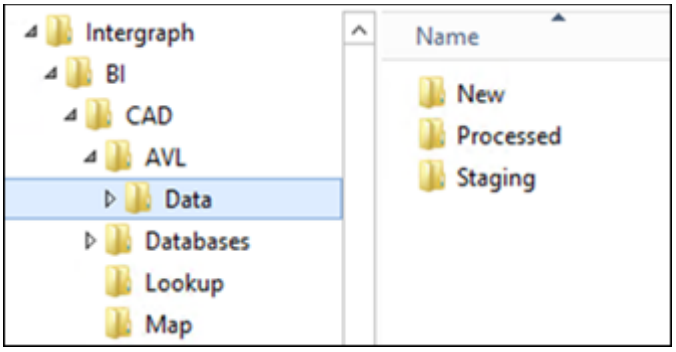




TRACKER DATA FILES (.TRK)

The AVL ETL looks for tracker data files (.trk) and loads them into the AVL data warehouse, INGRL_AVL_DW, after processing.

By default, the AVL ETL looks for these files under C:\Intergraph\BI\CAD\AVL\Data.

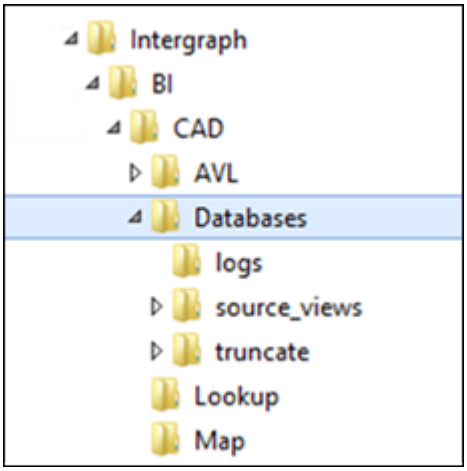


- All tracker data files (.trk) must be placed in the \New folder to be picked up by the ETL. When the ETL begins processing the tracker file, it moves the file from the \New folder to the \Staging folder. Once the ETL has completed processing the file, it moves the file from the \Staging folder to the \Processed folder.
- <INSERT NOTE IMAGE HERE> The files in the \Processed folder should be cleared on a regular basis as these files can be large and can use up available storage space quickly.

DATABASE SCRIPTS (.SQL)

The BI databases can be dropped and re-created using SQL scripts delivered with the product. These scripts can be used when building a new server, migrating to a different server, or truncating data after an ETL run that produces errors.

By default, the database scripts are delivered at C:\Intergraph\BI\CAD\Databases.



The truncate scripts are located under the \truncate folder. Update the variables in the SETENV.bat file and run the scripts using Command Prompt.

CAD MAP FILE UPDATES

The map file used by InSight must match the active CAD map file. Therefore, if the map file changes in CAD, you must also replace it in InSight.

- When replacing the map file with a new map file:
  - The file must have the same name as the old map file.
  - The file must be placed in the same folder as the old map file.

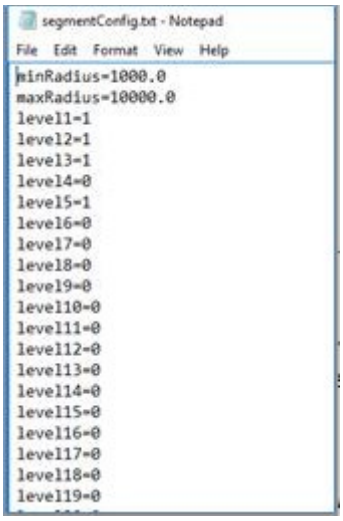
If any street segments change levels, you mustpdate the segmentConfig.txt file (see "Update Street Segment Configuration File")

The next ETL run consolidates all the map changes into the data warehouse.

UDPATE STREET SEGMENT CONFIGURATION FILE

This section defines how to update the SegmentConfig.txt file. This file is used by the AVL web service when running the AVL ETL to locate information about the street segments.

10. Open Windows Explorer. Navigate to ..\Intergraph\BI\CAD\WebServices\Work\AVLWebService.
11. Open the SegmentConfig.txt file.
12. If the levels that contain street network elements are known, edit the file and set the levels with the street network elements to 1, and all other levels to 0. In the screenshot below, levels 1, 2, 3, and 5 represent the street network.



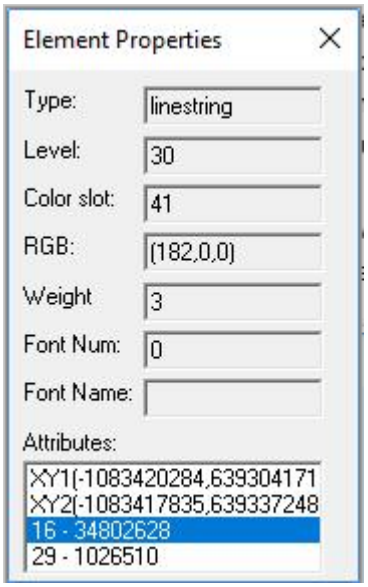
5. At the bottom of the file, there is an entry for the entity number called “entities”. Set this to the entity number used for the database linkage in the .map file.
   If the levels are not known, use the ShowMap.exe application, located in ..\Intergraph\BI\Common\Tools\ShowMap.

   Browse to the .map file and discover the required levels and entity number:

a. Turn on one level that looks like it contains roads and click **OK**.
b. If the level contains streets, then select the **Locate** tool from the menu and then select the road and double click on it to view the attributes.

c. If the attributes show only XY coordinates and nothing else, then this level can be ignored. It is not linked to database. Put 0 for this level.
d. If the attributes show XY coordinates plus an object ID and the value, then this road feature is linked to the database. The object ID is the entity number and the value is the database link. Put 1 for this level. Put the entity number at the bottom of the segmentConfig.txt file where it says entities=.
e. Repeat the steps above for remaining levels (there are 64 levels). Usually there are only a few levels that contain roads with linkages, such asprimary highways, secondary highways, city streets, and so forth.
f. Modify the segmentConfig.txt file to reflect the levels and the entity number.

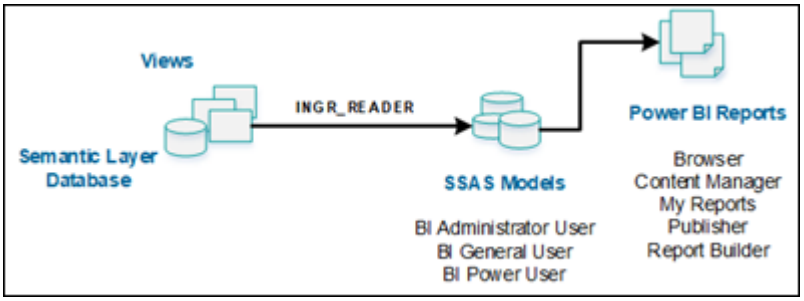
In the following example, entities=16



SECURITY: A MULTI-LAYER SECURITY MODEL

Security in InSight is managed at multiple levels:

- Agency Level Security
  - Managed by InSight System Administrators
- Report Server Security
  - Managed by InSight System Administrators
  - Five default roles are provided: Browser, Content Manager, My Reports, Publisher, and Report Builder
- Database Security
  - Managed and controlled by a DBA
  - Three default SQL logins are provided: INGR_BASE, INGR_READER, and INGR_ADMIN
  - Authentication uses Windows Integrated Security using a Windows service account to run applications and windows users with privileges to create reports.
- Semantic Models Security (SSAS)
  - Managed by InSight System Administrators
  - Three default roles are provided: BI Administrator User, BI General User, and BI Power User



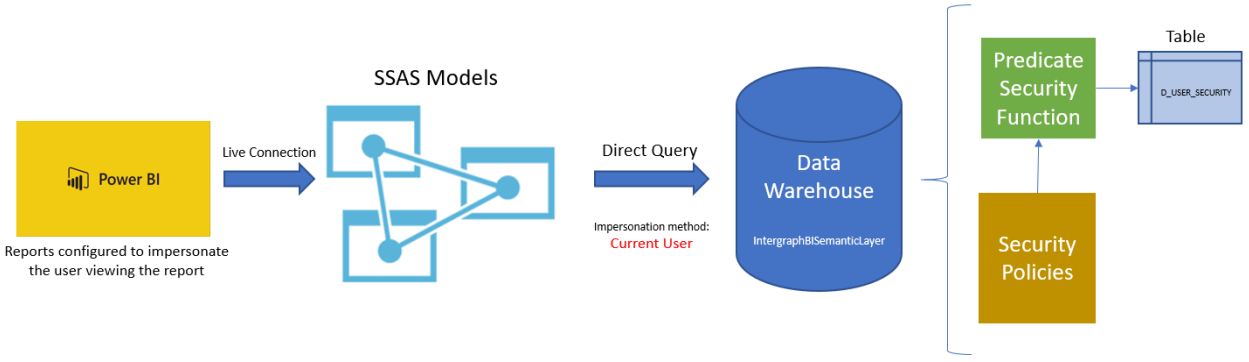
AGENCY LEVEL SECURITY

Agency level security is managed by SQL Server Row-Level Security (RLS). This enables customers to control access to rows in a database table based on the credentials of the user executing a query. In the InSight solution, I/CAD data can be secured by agency. Data from one agency cannot be accessible by another agency unless access is explicitly given.

To enable this feature, the **Enable Row Level Security** option should be checked during the installation process. Also, the BI Administrator needs to populate the D_USER_SECURITY.xml file, create the Windows User Groups, and assign the users for each group. These tasks are explained below.

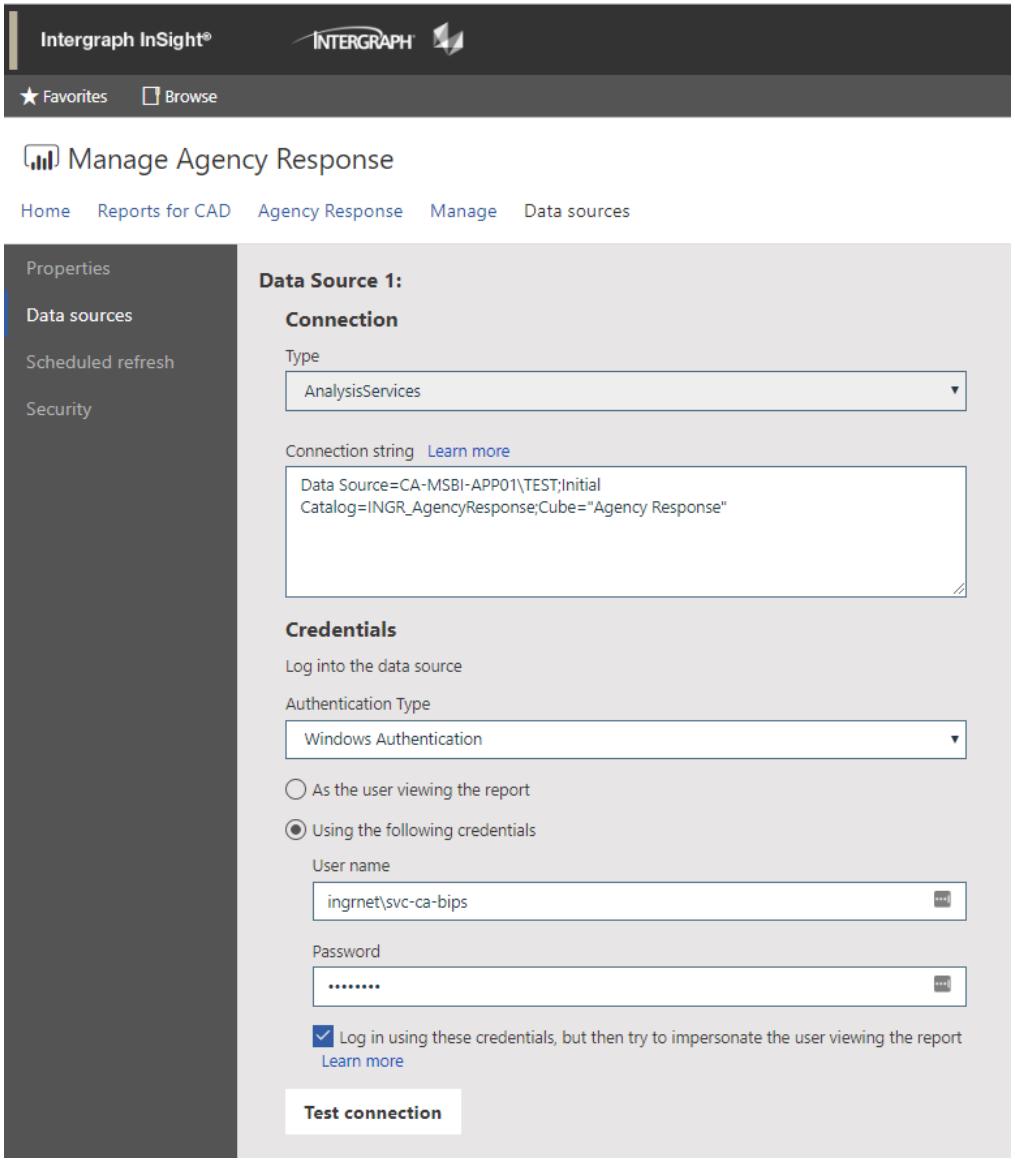
The architecture used to implement agency-level security is composed of five key elements:

- Reports connect to SSAS using a windows service account and user impersonation.
- SSAS models connect to the data warehouse impersonating the current user.
- Predicate functions based on agency identifiers in each data row, such as AGENCY_DKEY and AGENCY ID.
- Security policies for each fact table in the data warehouse.
- Metadata that maps users and groups to agencies.

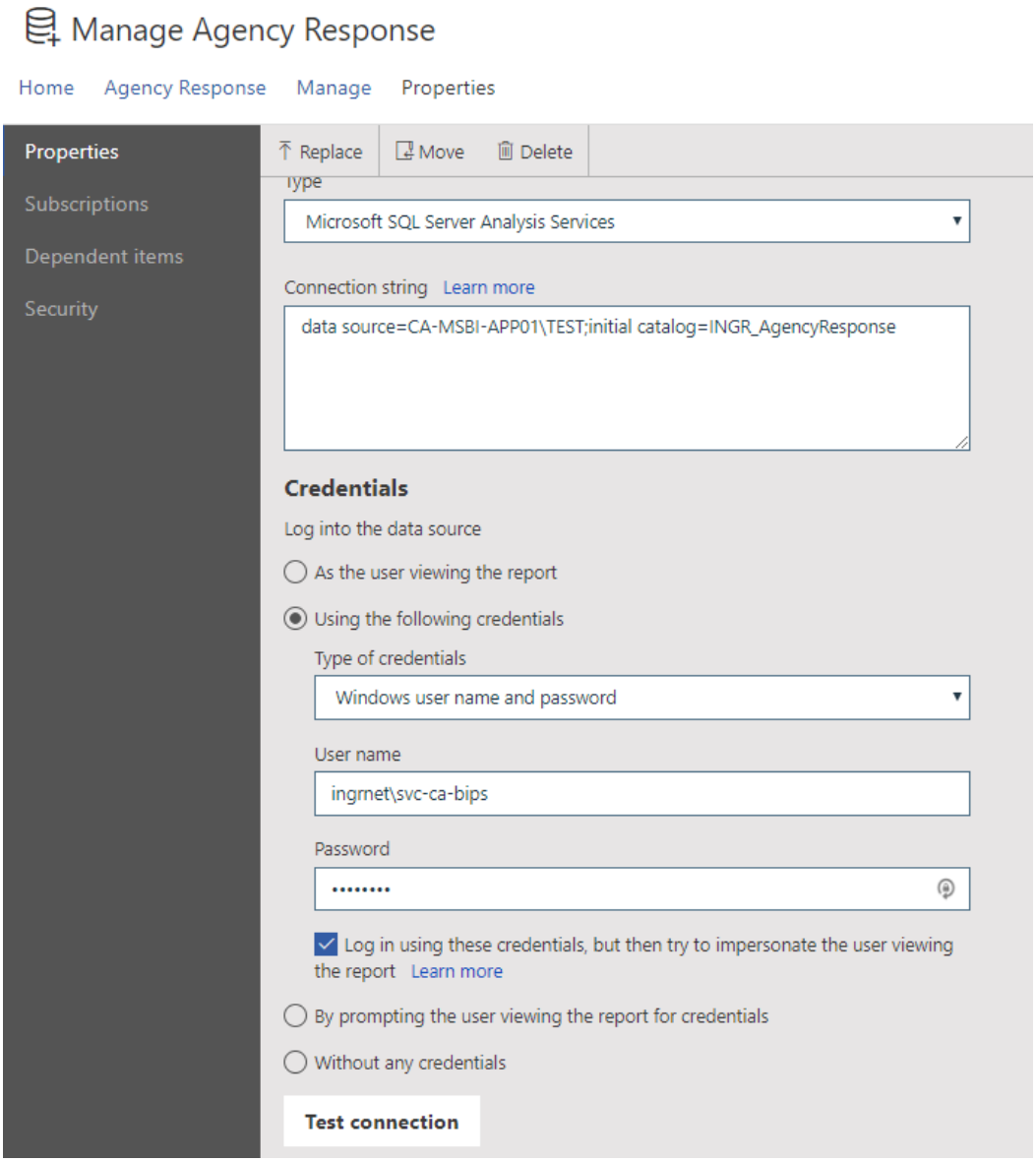


- Reports:

- The **Power BI Reports** needs to be configured to impersonate the user viewing the report, but use a service account's credential to log into the database.

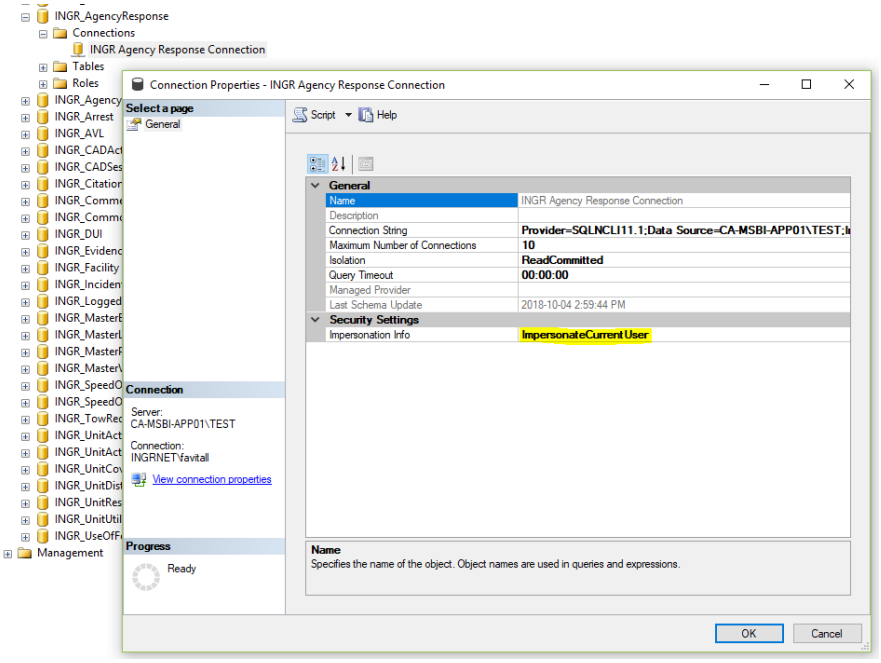


- The **Paginated Reports** need to have their data sources configured to impersonate the user viewing the report but use a service account's credentials to log into the database.

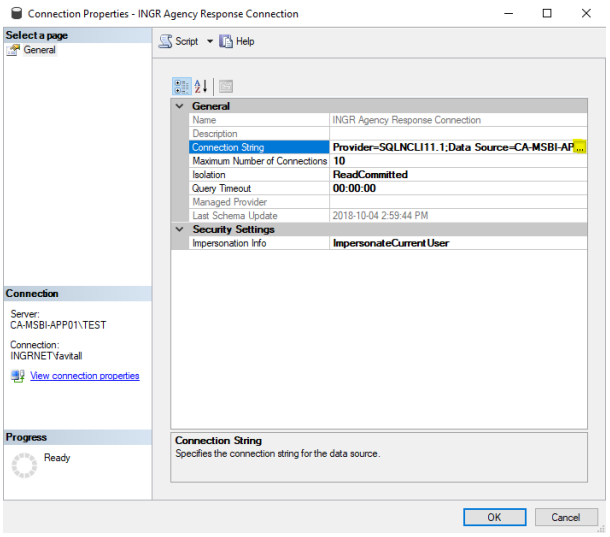


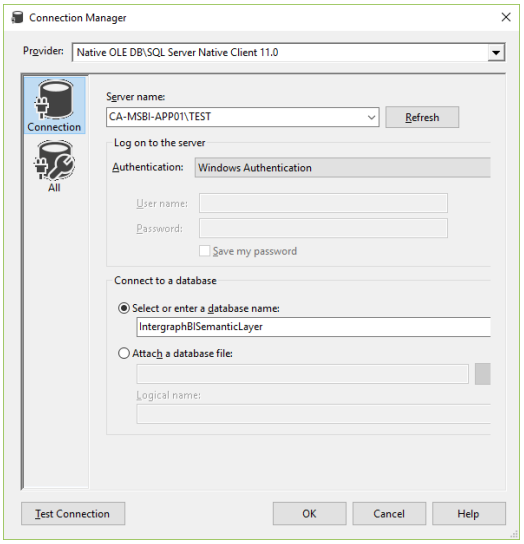
- The connections for the SSAS models need to be configured to use Windows Authentication as the authentication mode when connecting to the data warehouse. In addition, the **ImpersonateCurrentUser** option must be enabled. The administrator can verify those settings by accessing the SSAS instance, expanding a model, and right-clicking on the connection name. The

Connection Properties window appears, and the impersonation info can be verified.



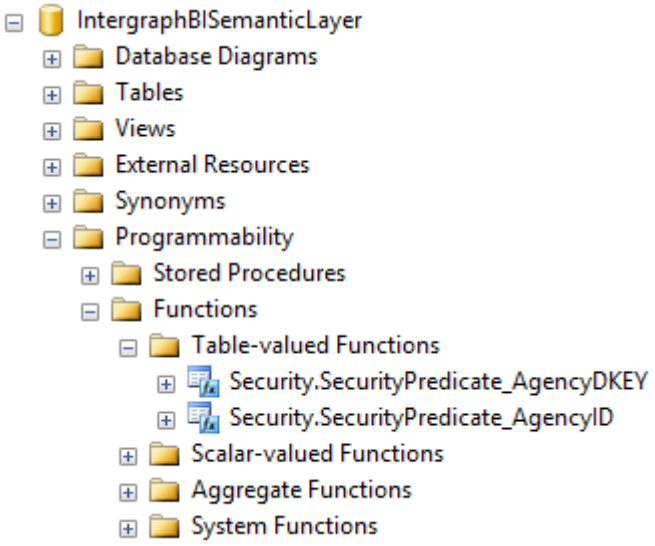
To open the **Connection Manager** windows, click on the ellipsis (…) button at the end on the **Connection String** line.



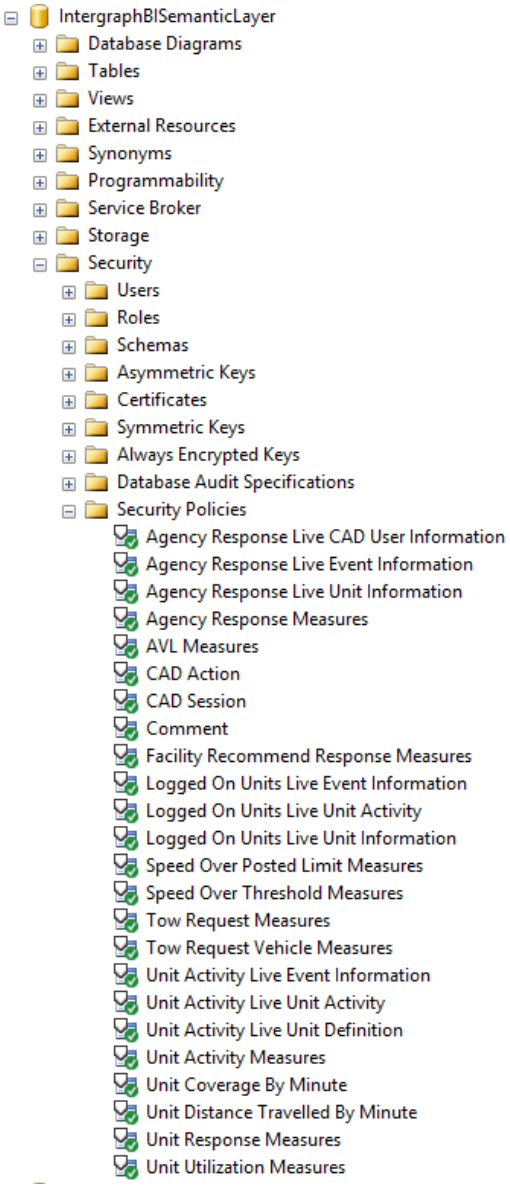


- The predicate functions are required to implement the security policies. The predicates are filters that are used to secure the data by row. There are two predicate functions:
  - **SecurityPredicate_AgencyDKEY**: This function filters all data based on the unique agency identifier AGENCY_DKEY. AGENCY_DKEY is a system generated key used by the data warehouse to uniquely identify an agency.
  - **SecurityPredicate_AgencyID**: This function filters all data based on the unique agency identifier AGENCY_ID. AGENCY_ID is the business key in the CAD database that uniquely identifies an agency.

  These functions are implemented in the **IntergraphBISemanticLayer** database. All access must be through this layer.



1. The security policies assign the predicates to the appropriate tables to secure the data. These security policies are created in the **IntergraphBISemanticLayer** database and can be founded under **Security > Security Policies**.


IntergraphBISemanticLayer
Database Diagrams
Tables
Views
External Resources
Synonyms
Programmability
Service Broker
Storage
Security
Users
Roles
Schemas
Asymmetric Keys
Certificates
Symmetric Keys
Always Encrypted Keys
Database Audit Specifications
Security Policies
Agency Response Live CAD User Information
Agency Response Live Event Information
Agency Response Live Unit Information
Agency Response Measures
AVL Measures
CAD Action
CAD Session
Comment
Facility Recommend Response Measures
Logged On Units Live Event Information
Logged On Units Live Unit Activity
Logged On Units Live Unit Information
Speed Over Posted Limit Measures
Speed Over Threshold Measures
Tow Request Measures
Tow Request Vehicle Measures
Unit Activity Live Event Information
Unit Activity Live Unit Activity
Unit Activity Live Unit Definition
Unit Activity Measures
Unit Coverage By Minute
Unit Distance Travelled By Minute
Unit Response Measures
Unit Utilization Measures

2. A mapping of users and groups is defined in the metadata table called D_USER_SECURITY. This table resides in the IntergraphBISemanticLayer database. This table maps the users and groups to the agency using unique identifiers for each agency.

```
/****** Script for SelectTopNRows command from SSMS  ******/
SELECT [USER_SECURITY_DKEY]
      ,[AGENCY_DKEY]
      ,[WINDOWS_USERS_AND_GROUPS]
      ,[AGENCY_ID]
      ,[READ_ONLY]
      ,[ETL_EXECUTION_ID]
      ,[ETL_EXECUTION_DTS]
  FROM [IntergraphBISemanticLayer].[dbo].[D_USER_SECURITY]
```

100 %

Results | Messages

	USER_SECURITY_DKEY	AGENCY_DKEY	WINDOWS_USERS_AND_GROUPS	AGENCY_ID	READ_ONLY	ETL_EXECUTION_ID	ETL_EXECUTION_DTS
1	1	1	CA-MSBI-APP01\Highview_NOLOGIN	11	Yes	1	2018-10-09 13:59:39.813
2	2	2	CA-MSBI-APP01\Worthington_NOLOGIN	18	Yes	1	2018-10-09 13:59:39.827
3	3	3	CA-MSBI-APP01\Middletown_NOLOGIN	99	Yes	1	2018-10-09 13:59:39.827
4	4	4	CA-MSBI-APP01\EMS_NOLOGIN	EMS	Yes	1	2018-10-09 13:59:39.827
5	5	5	CA-MSBI-APP01\LFD_NOLOGIN	LFD	Yes	1	2018-10-09 13:59:39.827
6	6	6	CA-MSBI-APP01\LMPD_NOLOGIN	LMPD	Yes	1	2018-10-09 13:59:39.827
7	7	7	CA-MSBI-APP01\SFD_NOLOGIN	SFD	Yes	1	2018-10-09 13:59:39.827
8	8	1	CA-MSBI-APP01\Highview	11	No	1	2018-10-09 13:59:39.827
9	9	2	CA-MSBI-APP01\Worthington	18	No	1	2018-10-09 13:59:39.827
10	10	3	CA-MSBI-APP01\Middletown	99	No	1	2018-10-09 13:59:39.827
11	11	4	CA-MSBI-APP01\EMS	EMS	No	1	2018-10-09 13:59:39.827
12	12	5	CA-MSBI-APP01\LFD	LFD	No	1	2018-10-09 13:59:39.827
13	13	6	CA-MSBI-APP01\LMPD	LMPD	No	1	2018-10-09 13:59:39.827
14	14	7	CA-MSBI-APP01\SFD	SFD	No	1	2018-10-09 13:59:39.827

This table is populated by the ETL using an XML file as a source data. The XML file can be found in the default path for the XML files. Shown below is an example of the XML file, called D_USER_SECURITY.xml.

	A	B	C
1	WINDOWS_USERS_AND_GROUPS	AGENCY_ID	READ_ONLY
2	CA-MSBI-APP01\Highview_NOLOGIN	11	Yes
3	CA-MSBI-APP01\Worthington_NOLOGIN	18	Yes
4	CA-MSBI-APP01\Middletown_NOLOGIN	99	Yes
5	CA-MSBI-APP01\EMS_NOLOGIN	EMS	Yes
6	CA-MSBI-APP01\LFD_NOLOGIN	LFD	Yes
7	CA-MSBI-APP01\LMPD_NOLOGIN	LMPD	Yes
8	CA-MSBI-APP01\SFD_NOLOGIN	SFD	Yes
9	CA-MSBI-APP01\Highview	11	No
10	CA-MSBI-APP01\Worthington	18	No
11	CA-MSBI-APP01\Middletown	99	No
12	CA-MSBI-APP01\EMS	EMS	No
13	CA-MSBI-APP01\LFD	LFD	No
14	CA-MSBI-APP01\LMPD	LMPD	No
15	CA-MSBI-APP01\SFD	SFD	No

The D_USER_SECURITY.xml file has three attributes that needs to be populated:

- WINDOWS_USERS_AND_GROUPS – Name of the group or user.
- AGENCY_ID – ID of the agency associated with the user or group.
- READ_ONLY – Flag to identify if the user or group has privileges to only run the reports (Yes) or to also create and publish reports (No).

In order to make the management of the users more effective, the system administrator should create local or domain groups and assign users for these groups.

Two windows groups per agency should be created:

- Read-only users and groups that need to access the InSight portal only to run and read reports
- Login users and groups that need to create reports and analytics.

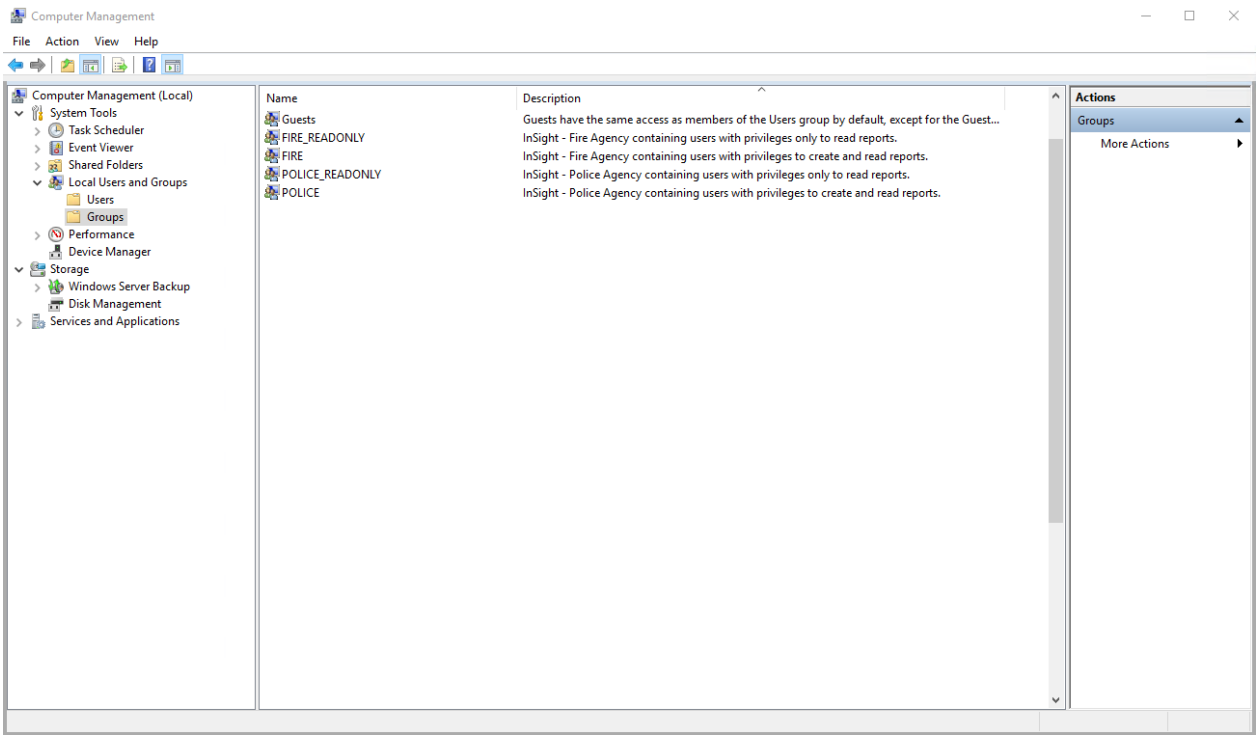
The users within the read-only group cannot login into the Server data warehouse.

- The login users have login privileges on SQL Server and these users can use Power BI Desktop to connect to the semantic models using their windows credentials to create and publish reports.

For example, if the InSight solution is being implemented for an I/CAD customer with two agencies, Fire and Police, the administrator should create four Windows Groups:

FIRE
FIRE_READONLY
POLICE
POLICE_READONLY

The image below illustrates the groups created for Police and Fire agencies.



The ETL reads the D_USER_SECURITY.xml file and populates the D_USER_SECURITY table. The ETL then creates the necessary database users.

The system administrator is responsible for creating and modifying the windows users and groups, as well as maintaining the XML file to ensure that it is up to date.

REPORT SERVER SECURITY

Report server security is divided into two levels:

- Site-level security
  - Defines who can login to the portal
  - Two roles available: system administrator and system user
- Report-level security
  - Defines who can access/modify the reports
  - Five roles available: Browser, Content Manager, My Reports, Publisher, and Report Builder

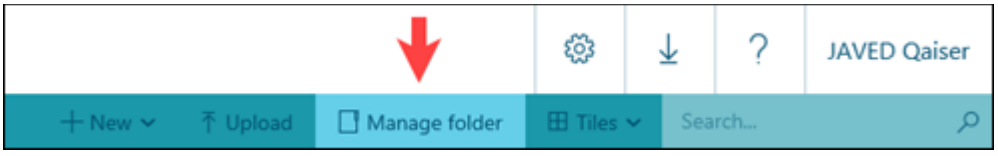
The two security levels work independently of each other.

SITE-LEVEL SECURITY

Site-level security lets you define system properties, role definitions, site branding, shared schedules, and other features. It can be accessed by navigating to **Settings > Site Settings > Security**.

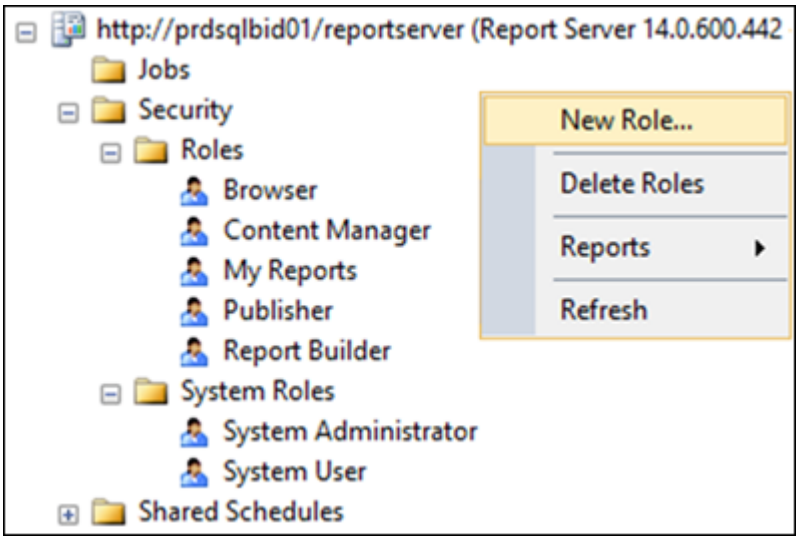
REPORT-LEVEL SECURITY

Report level security lets users create/modify reports and folders, view and publish reports, subscribe to reports, and other actions. It can be defined by clicking on the Manage Folder link available at the top of the screen.



ADJUSTING SECURITY ROLES

Security roles can be added, removed, or modified by connecting to the Report Server using SQL Server Management Studio. To add a new role, right-click on **Roles** and select **New Role.**



To modify a role, double-click on the role. A pop-up window lets you make modifications to the role.

DATABASE SECURITY

The InSight Data Warehouse instance allows direct connection for the following types of logins:

- InSight delivered SQL Logins:
  - INGR_READER: Read access privilege for all delivered BI databases.
  - INGR_ADMIN: Read/Write privilege for all delivered BI databases.
  - INGR_BASE: Owner of INGR_CAD_VIEWS database.
- Service Account: A windows domain user created by the customer to be used as a credential to connect to the data sources of the reports.
- Windows domain users: Users assigned to windows groups with privileges to create reports (Power BI Desktop Users) have SQL logins to connect to the database and

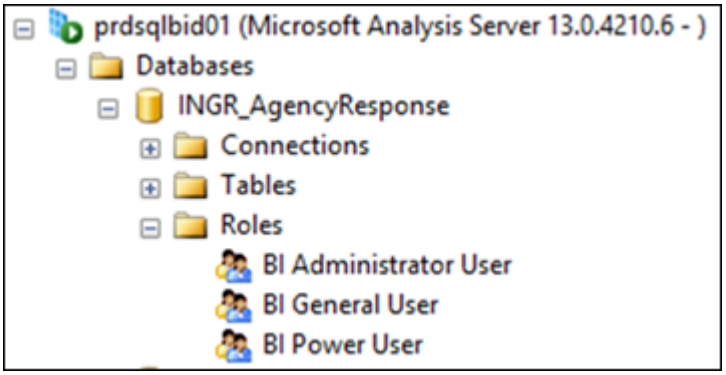
read data from IntergraphBISemanticLayer, INGR_CAD_DW, INGR_COMMON_DW and INGR_CAD_VIEWS databases.

SEMANTIC MODELS SECURITY

BI reports access data through the SSAS Semantic Layer Models.

This access is controlled using three pre-defined roles:

- **BI Administrator User**: Full Control
- **BI Power User**: Read and Process
- **BI General User**: Read-Only



Every BI user must be a member of one of these roles in order to view data in reports. By default, every user is added with both **Power User** and **General User** roles.

TROUBLESHOOTING

ETL RELATED ERROR MESSAGES – Use the following steps to identify the cause of an ETL job failure.

STEP 1: Check SQL Server Agent's **Job Activity Monitor** to discover which job has failed.

STEP 2: Connect to Integration Services Server and look under **Integration Services Catalogs**. Locate the Project that has failed, as identified in Step 1, and generate a Standard Report for All Executions.

Read the report and locate the source of the error.

- If the error is related to the environment (network, login, database, and so forth) then fix the cause and rerun the ETL job manually.
- If the error is within the ETL code, contact Hexagon Support.

REPORTING PORTAL ERROR MESSAGES **–** If a report is giving an error or not displaying data.

STEP 1: Check SQL Server Agent's **Job Activity Monitor** to make sure ETL job did not fail.

STEP 2: Login to Reports Portal and navigate to **Data Sources** for the report.

Under **Connection**, verify the Connection String and make sure the Data Source is pointing to the correct server. Under **Credentials**, make sure User Name is the service account with the correct password.

Click **Test Connection**.

- If the connection is successful, contact Hexagon Support for further investigation.
- If the connection failed, contact the BI system administrator to verify **SQL Server Analysis Services** is running on the SQL Server instance specified with the Connection String above.

  Once verified, test the connection again.
  - If it still fails, contact Hexagon Support.
  - If connection is successful, run the report again and see if you are getting results.

LOGIN RELATED ERROR MESSAGES **–** If a user cannot access a folder or report

STEP 1: Login to **Reports Portal** and navigate to the folder or report which is not displaying

STEP 2: Click on Manage Folder link, if it is a folder, or Manage link, if it is a report, and navigate to **Security** tab. Verify that the user name is in the list.

- If user name is missing from the list, add the user with appropriate role
- If the user still cannot see the report or folder, contact Hexagon Support.

INDEX