

Internal Setup Utility

Overview

The [Internal Setup Utility](#) is for the use of system administrators to troubleshoot and fix database problems in SmartPlant Instrumentation without the need to log on to SmartPlant Instrumentation. The [Internal Setup Utility](#) is installed when you perform your network installation.

With the [Internal Setup Utility](#), you can run SQL statements for updating the database or for ODBC configuration. You can also compare and match a version of SmartPlant Instrumentation with the number available in the version information file.



Caution



- Incorrect use of this utility can damage the SmartPlant Instrumentation database. If you are unsure of any of the procedures, back-up your database, and contact Intergraph Support before starting to work with the [Internal Setup Utility](#).

Database Connection

Testing the Database Connection

This procedure allows you to troubleshoot SmartPlant Instrumentation database connection problems.

➤ To test the database connection

1. On the **File** menu, click **Database Type Test and Query**.
2. Do one of the following:
 - On the **Actions** menu, click **Use INI File**.
 - On the toolbar, click  to retrieve the parameters from the INTOOLS.INI file for the current Admin schema.
 - In the **Database type** group box, click the type of database that you are working in. Type the appropriate database parameters in the fields to retrieve the parameters from a different Admin schema, or from a specified Domain schema.
3. Do one of the following:
 - On the **Actions** menu, click **Connect**.
 - Click .



Notes

- The **Result** window displays the outcome of the connection. A zero (0) indicates a successful operation. Codes that indicate a connection failure include a message explaining the reason for the failure.
- You can run this option to verify whether the database that appears in the **DBParm** text box is compatible with the selected DBMS platform.

Checking Database Version Compatibility

When you run SmartPlant Instrumentation after installation or update, if a message appears that there is an inconsistency between the database and the application, this procedure can help you to start troubleshooting the source of the problem. You can then notify Intergraph Support.

➤ To check the database and application versions

1. On the **File** menu, click **Versions**.
2. Click **Read**.



Note

- By default, the [Internal Setup Utility](#) reads data from the SmartPlant Instrumentation database and the version information file.

3. Beside **Domain**, select the domain in which you want to check the version numbers.



Notes

- If the version information file number is higher than the corresponding domain database number, the most likely reason for the incompatibility is that the database needs updating. In this case, you should open the [Administration](#) module and run the upgrade option on the domain.
- If the version information file number is lower than the corresponding domain database number, the most likely reason is that an older SmartPlant Instrumentation version was installed over an existing database. In this case, you should contact Intergraph Support.



Caution

- The next steps should be performed by Intergraph Support only.
4. Click **Copy** to copy the SmartPlant Instrumentation database parameters to the version information file.
 5. Click **Save** to save the parameters to the version information file, C_VER.PBD.



Note



- The template database does not have a version number.

Database Queries

Running SQL Statements


You can run SQL statements to retrieve data from the database, or to update values in the database.

➤ To run an SQL statement

1. On the **File** menu, click **Database Type Test and Query**.
2. Do one of the following:
 - On the **Actions** menu, click **Use INI File**.
 - On the toolbar, click  to retrieve the parameters for the current Admin schema from the INTOOLS.INI file.
3. Do one of the following:
 - On the **Actions** menu, click **Connect**.
 - On the toolbar, click .
4. Under **Query**, type the SQL statements that you want to run.



Notes

- End each statement with a ; (semicolon).
 - Do not type more than one SQL select statement.
5. Do one of the following:
 - On the **Actions** menu, click **Execute**.
 - On the toolbar, click .



Note

- If you run an SQL SELECT statement, you retrieve data at all lower levels in the data hierarchy, but not from higher levels. For example, you can retrieve <plant> data when selecting to retrieve at the <unit> level, but not the other way round.

Examples of SQL Statements

The following SQL SELECT statement retrieves the current software version and project names:

```
Select rev_software, proj_name from project;
```

The following SQL SELECT statement retrieves loop names and their associated tag names:

```
Select component.cmpnt_name, loop.loop_name from component, loop  
where component.loop_id = loop.loop_id;
```

or:

```
Select c.cmpnt_name, l.loop_name from component c, loop l where  
c.loop_id = l.loop_id;
```



Notes

- End each statement with a ; (semicolon).
- In this example, the 'where' argument ensures that the tag names appear with the correct relationship to the loops to which they belong.

Save As Options

After retrieving data, you can save it in several formats such as Excel spreadsheets, PowerSoft reports, or SQL.

➤ To save the results of an SQL statement in a specified file format

1. Run an SQL SELECT statement.
2. After the results appear, on the **File** menu, click **Save As**.
3. In the dialog box, select one of the following data formats:
 - **Original**: Saves the data in all the fields, including the ones that are not visible in the preview (for example, internal database ID numbers). This option is recommended if you want to save the file in .psr or .sql format, or if you want to re-import the data into SmartPlant Instrumentation at a later stage.
 - **Data Only**: Saves only the data that is visible in the preview. This option is recommended for easier viewing of the data, and also enables you to manipulate the headers and select the columns you want to display.
4. Click **OK** and navigate to the location where you want to save the file.
5. Select the required file format and type the name of the file, then click **Save**.



Tip

- You can save a result in the original format as an SQL file under the name of the table that you specified in the Select statement. When you run the SQL file, it creates the table. You can delete all the data except for the set of values under the first 'insert into' section and run the file as a script to insert the zero (default) line if it is missing. This is a requirement for making correct validations in Sybase Adaptive Server Anywhere.

SmartPlant Instrumentation Database Structure

The SmartPlant Instrumentation database consists of a hierarchy, where each level of the hierarchy contains tables that are specific to that level. The first level of the hierarchy is the Admin schema. On the next level of the hierarchy, there is a Domain schema for each domain you create in the database.

Within each domain, there are plant hierarchy levels. By default, each domain has three levels as follows: plant, area, and unit. When you perform an upgrade, the Admin schema is updated first, then the software upgrades each domain.

The following table shows some of the most commonly used tables at each level of the hierarchy:

Level	Tables Included	Notes
Admin	Hook-up Library, users	
Domain	Supporting tables, for example, Manufacturer, Model, I/O, default wiring Specification forms	Defaults are created for these entities
<Plant>	Wiring (Panel, Strip, Terminal, Cable, Cable Set, Wire) Custom fields	Defaults are created for these entities
<Area>	None	Used as a container only
<Unit>	Loops	The system checks uniqueness at the <unit> level
	Instrument	The system checks uniqueness at the <unit> level

There is a connection between the values of entities at each level, for example, an instrument includes data at the following levels:

COMPONENT (instrument) at the <unit> level

SUPPORTING TABLES at the domain level

PD_GENERAL (process function: pressure, flow, temperature, and so forth) at the <unit> level. Since this entity is at the same level as the tag itself, it contains no default values.

SPEC_SHEET_DATA at the <unit> level (connected to PD)

Due to this set of relationships within the hierarchy, it is recommended that you define only one <plant> per domain.

Database Relationship Example

The following example shows how data in the LOOP and COMPONENT tables is related.

The LOOP table includes the following columns:

Column	Example Value	
LOOP_ID	1000	Primary key
LOOP_NAME	F-100	
LOOP_SERV		

The COMPONENT table includes the following columns:

Column	Example Value	
CMPNT_ID	999	Primary key
CMPNT_NAME	FT-100	
LOOP_ID	1000	Foreign key
CMPNT_SERV		

A primary key is the column or a combination of columns where the values determine the uniqueness of each record. Note that both tables include the column LOOP_ID. This column is the primary key in the LOOP table. In the COMPONENT table, the LOOP_ID column uses the value of LOOP_ID in the LOOP table to associate the tag to the loop, and is therefore the foreign key in the COMPONENT table.

In the LOOP table, there is a zero row, which is used as a reference row if you create a tag that is not associated with any loop.

The following SQL SELECT statement retrieves loop and tag names with the associations between them:

```
Select component.cmpnt_name, loop.loop_name from component, loop
where component.loop_id = loop.loop_id;
```


Service Operations

Creating an ODBC Profile

You need ODBC profiles to configure database files in the Windows registry and in the INTOOLS.INI file. The [Internal Setup Utility](#) provides a convenient method of creating these profiles. For example, IN_CTLOG.DB requires the ODBC driver, IN_TEMPL.DB is used for updating the database, and INTOOLS.DB is used for backup. These files need to match to the appropriate ODBC profile.

➤ To create a new ODBC profile

1. On the **Settings** menu, click **Create ODBC Profiles** to open the **ODBC Profile Wizard**.
2. Click **Next**, and under **Profile name**, type a name for the profile, and then click **Next**.
3. Under **Database path and filename**, type the database path and filename, or click **Browse** to navigate to the database file (for example: INTOOLS.DB), and then click **Next**.
4. Under **Schema name**, type the schema name for this profile, or accept the default, and then click **Next**.
5. Under **Schema password**, type the password, or accept the default, and then click **Next**.
6. Under **Database driver path**, do one of the following:
 - Accept the displayed path and database driver.
 - Type the path and database driver.
 - Click **Browse** to navigate to the database driver.
7. Click **Next**, and under **Database start command**, accept the displayed value, or type the name and path of the database engine, and then click **Next**.
6. On the **Update INTOOLS.INI file** page, do one of the following:
 - To update the Windows registry and the [database] section of the INTOOLS.INI file, select **Modify INtools.ini file and Windows registry**.
 - To update the registry, but not the INTOOLS.INI file, select **Do not modify INtools.ini file**.

7. From the **Create Windows registry entry** list, select one of the following options:
 - **SYSTEM DSN** — to create a database profile on the registry entry level of the local machine so that every user working on this machine can connect to the database using this profile. The profile appears in the Windows registry in the following path:
HKEY_LOCAL_MACHINE\SOFTWARE\ODBC\ODBC.INI.
 - **USER DSN** — to create a database profile on the registry entry level of the current user so that only the current user working on this machine can connect to the database using this profile. The profile appears in the Windows registry in the following path:
HKEY_CURRENT_USER\SOFTWARE\ODBC\ODBC.INI.
 - **USER DSN and SYSTEM DSN** — to create a profile both registry entry levels.
8. Click **Next**, and then click **Finish**.



Notes

- You can also configure the profile using the Windows **Control Panel**, **Data Sources** option, on the **System DSN** tab or **User DSN** tab.
- If a database profile row already exists in the INTOOLS.INI file, the [Internal Setup Utility](#) adds the profile as a new row, for example:
`DBParm=ConnectionString='DSN=TEST;UID=IN_DBAMN;PWD=IN_DBAMN'`
- If there are more than three rows representing database profiles, the [Internal Setup Utility](#) deletes the first row from the INTOOLS.INI file when adding the new row.

Rebuilding Stored Procedures and Triggers

This procedure allows you to identify abnormal database behavior and correct it by rebuilding the stored procedures and triggers in the database. Stored procedures and triggers are portions of code that run automatically when you perform certain actions that affect the database. For example, when you delete a loop, there is a stored procedure that runs to delete an associated loop drawing. You can recreate stored procedures either for the Admin schema or for the Domain schema.

The System Administrator should perform this procedure if inappropriate SQL messages are displayed in SmartPlant Instrumentation.



Note

- Normally, you should run this option from the [Administration](#) module; however, you can run it from the [Internal Setup Utility](#) if for some reason you cannot log on to the [Administration](#) module.

➤ To rebuild the stored procedures and triggers of your database

1. On the **Settings** menu, click **Rebuild Procedures and Triggers**.
2. Select the required schema.
3. Click **OK**.
4. In the **Database Upgrade** dialog box, type an alternative log file name, if required, and click **OK** to start updating the stored procedures and triggers of the selected schema. At the end of the upgrade process, the software displays an appropriate message and generates LOGERROR.TXT and LOG.TXT files.



Notes

- If the upgrade stops for any reason (for example, insufficient memory), you can restart the process and the upgrade will continue from where it stopped. If you get an error that cannot be corrected, contact Intergraph Support with the error description. It is recommended that you have your log file available when contacting Intergraph Support.
- The LOGERROR.TXT file lists any errors that may have occurred. The data in this file is incremented between sessions. Please send the LOGERROR.TXT file (if created) to Intergraph after you complete the upgrade process.

Testing the Printer

This option allows you to troubleshoot printing problems by testing the default printer and displaying current settings.

➤ To test the printer

1. On the **File** menu, click **Printer Diagnostics**.
2. Click **Test** to display the parameters for the default printer.



Note

- If necessary, click **Setup** to change settings for the default printer.
3. Under **Printing orientation**, select **Portrait** or **Landscape**.
 4. Click **Close** to return to the main the [Internal Setup Utility](#) window.