BeyondTrust

Remote Support
Integration Client

TC:4/5/2019

Table of Contents

Integration Client Guide

The BeyondTrust Integration Client is used to transfer session logs and recordings from the Secure Remote Access Appliance to an external system. Two external systems are currently supported: Microsoft SQL Server and Windows-based file systems.

The BeyondTrust Integration Client supports plugins for these systems. A plugin defines the transfer details, such as the destination directory/file name or database to use. Plugin details and the standard SQL Server Schema are defined later in this guide.

This guide walks you through the installation and configuration process for the BeyondTrust Integration Client. To begin using the integration client initially, you should:

1. Ensure prerequisites are met. See "Integration Client Prerequisites" on page 4.
2. Download the BeyondTrust Integration Client installation package from the BeyondTrust Self Service Center at **ssc.bomgar.com**.
3. Install the integration client package. See "Install the BeyondTrust Integration Client" on page 5.
4. Configure the settings database. See "Create the Settings Database" on page 7.
5. Configure the BeyondTrust site. See "Configure the BeyondTrust Site" on page 9.
6. Configure the File System or SQL Server plugin. See "Configure the File System or SQL Server Plugins " on page 10.
7. Review the SQL Server Storage Database. See "SQL Server Storage Database" on page 14.
8. Set the plugin schedule. See "Set the Integration Client Schedule" on page 20.
9. Optional: test the integration client specific plugin. See "Test the Integration Client" on page 22.
10. Review the BeyondTrust Integration Client Toolset. See "Integration Client Tools" on page 24.

Integration Client Prerequisites

There are several prerequisites to run the integration client:

- Prior to BeyondTrust 16.1, credentials for a BeyondTrust account require permissions to view reports and make backups. For BeyondTrust versions 16.1 and above, an API account with reporting and backup permissions is required.
- A Windows machine (7 or later) with access to both the external system to which data is to be transferred and the BeyondTrust site from which data is to be extracted
- Storage space sufficient to retain all desired recordings, session data, and backups
- Microsoft .NET Framework 4.5 or later
- On the host system, an enabled cipher suite matching one enabled on **/appliance > Security > SSL/TLS Configuration**
- The XML API interface enabled from the **/login > Management > API Configuration** page

The SQL Server has an additional set of requirements:

- Microsoft SQL Server Database 2008, 2008 R2, or 2012 Standard Edition or above. If you are running an earlier version of SQL Server, contact BeyondTrust Technical Support to determine if the database can be configured for your environment. BeyondTrust suggests a size of 100 GBs for the server database.
- Permission to create tables and execute **INSERT**, **UPDATE**, **DELETE** and **SELECT** statements in a specific database. Two tables are required on the SQL Server.

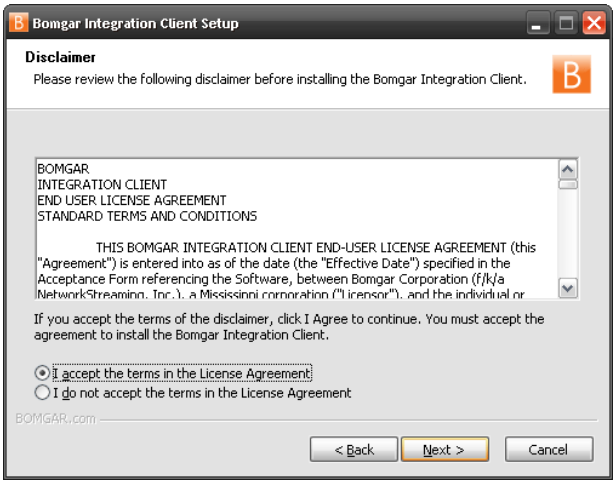
You may download the integration client installer from the "Patches and Utilities" page in the BeyondTrust Self-Service Center, or request it from BeyondTrust Technical Support: help.bomgar.com.

Install the BeyondTrust Integration Client

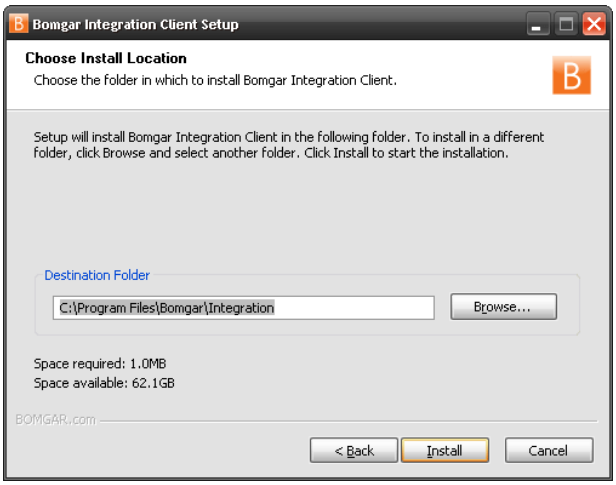
Once you have met the prerequisites and received the integration client installation package from BeyondTrust Technical Support, you are ready to install.

1. To access the integration client installer, you must log into the **BeyondTrust Self-Service Center** at ssc.bomgar.com.
2. Once authenticated, click on **Patches and Utilities** from the side menu.
3. From the list, locate the integration client compatible with your BeyondTrust site.
4. Download the **bomgar-ic-setup.exe** file to your Windows system and then run it.
5. From the installation wizard, click **Next**.

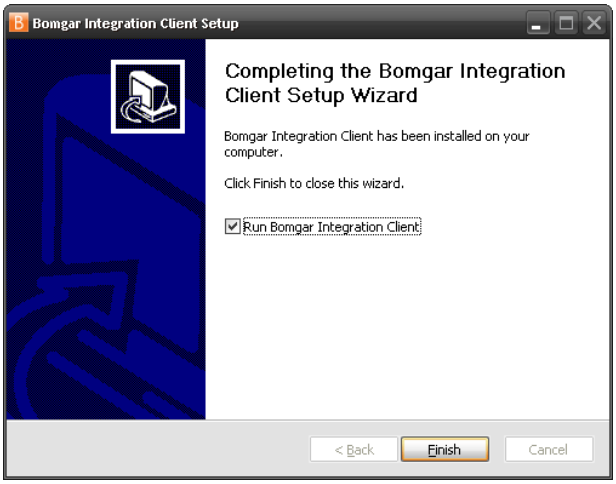
6. Read and accept the license agreement. If you do not accept the license agreement, you will not be able to proceed with the installation.



7. Choose where you would like the integration client to install. The default location is **C:\Program Files\Bomgar\Integration.**

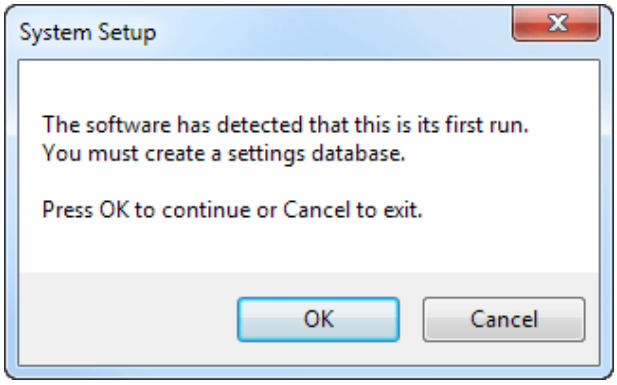


8. After installing the integration client, choose **Run BeyondTrust Integration Client** and then click **Finish.** Once the tool is installed, it must be configured before it can begin extracting session data from a BeyondTrust site.

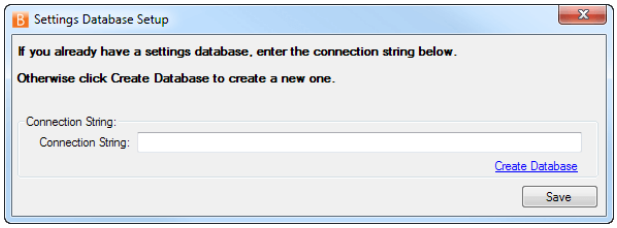


Create the Settings Database

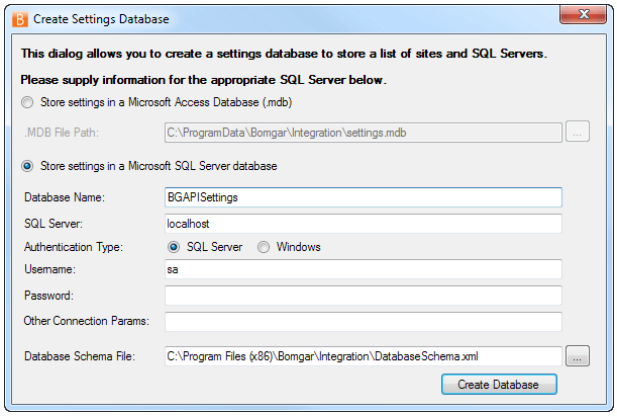
1. The first time you run the integration client, you will be prompted to create a settings database. Click **OK** to continue.



2. This database stores the BeyondTrust site information, schedule settings, and other configuration information that the tool will use to transfer information. To create a settings database, click the **Create Database** link.



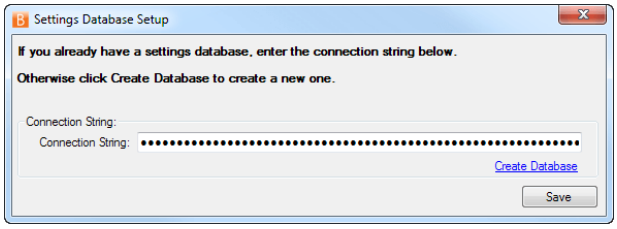
3. In the configuration dialog, enter the settings for your new database. These settings are defined in the table below. Once you have entered the appropriate settings, click the **Create Database** button.



Field	Explanation
Store Settings In	Choose to store the Integration Client settings in a Microsoft Access Database (.mdb) or a Microsoft SQL Server Database.
.MDB File Path	The path to the Microsoft Access Database that will store the settings database.
Database Name	The name to give this SQL Server settings database.
SQL Server	The name of the SQL Server instance that will store the settings database.
Authentication Type	Use **SQL Server** if a specific username and password are required. Use **Windows** if the logged-in Windows user account is required. See your SQL Server documentation for more details.
Username	The username used to access the SQL Server database.
Password	The password used to access the SQL Server database.

Field	Explanation
Other Connection Params	Optional field. Use this to specify additional connection string parameters which may be necessary for your specific database environment. See your SQL Server documentation for more details.
Database Schema File	Leave this at its default unless otherwise instructed by a BeyondTrust technician.

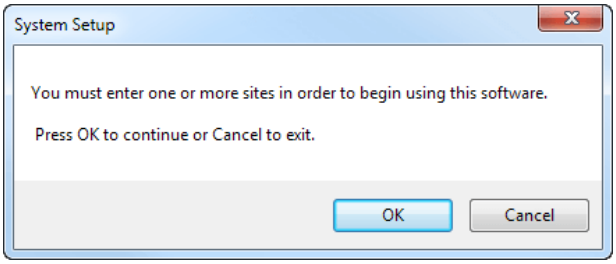
4. Once you have created the database, the new string for the settings database will automatically populate the **Connection String** field. Click **Save** to complete the settings database setup.



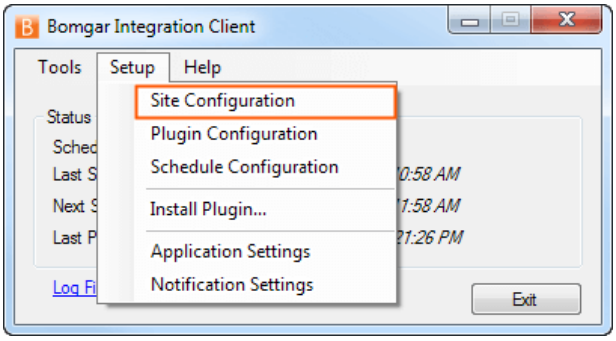
Note: *The settings database is distinct and must be kept separate from all storage databases.*

Configure the BeyondTrust Site

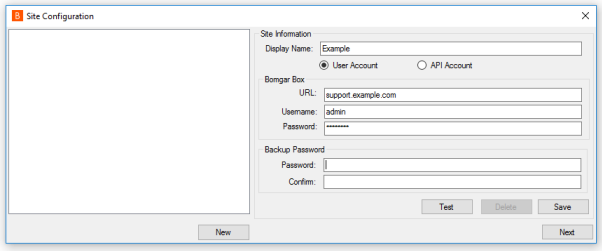
1. After you have created the settings database, you will be prompted to enter information for one or more BeyondTrust sites from which the integration client will extract session data. Click **OK** to continue.



2. If the integration client is already installed and you wish to update or add a site, select **Site Configuration** from the integration client **Setup** dropdown.



3. When the **Site Configuration** dialog appears, click the **New** button to input your BeyondTrust site information.
4. Enter a name for this site configuration and the URL of the site (note that **https://** should NOT be included)
5. For BeyondTrust sites on version 16.1 and above, you must provide the **Client ID** and **Client Secret** for an API account with permissions to view reports and recordings. If you plan to pull site backups, backup API permissions must also be enabled for the API account. Click **Edit** on the API user account to identify the OAuth Client ID, and click **Generate New Client Secret** and record the secret.



> ***Note:*** *For BeyondTrust sites running version 16.1 and above, if the account's password is reset, the integration client stops pulling data until the site configuration is updated. To prevent this break, it is recommended that you create a special account for the integration client with only permissions needed to retrieve the desired data and with a password set to never expire.*

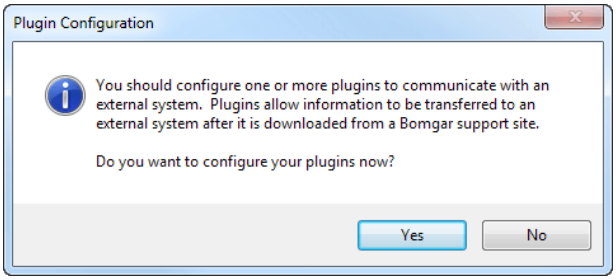
6. Optionally, you may apply a password to any backups created. If you do choose to set a password, you must provide this password in order to revert to the backup.
7. Test the supplied credentials and then click **Save**.
8. Note that the integration client supports more than one site. If session data from additional sites needs to be extracted, click the **New** button again and repeat the configuration process. The **host_name** in the **session** table distinguishes the data.
9. When you have finished entering your BeyondTrust site information, click **Next**.

Configure the File System or SQL Server Plugins

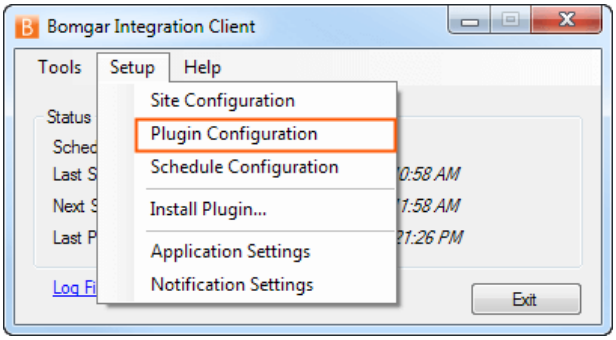
Plugins are used to send the downloaded data to external systems. You can add plugins during your initial setup, or you can also add plugins at any time once the integration client is installed. There are two standard plugins that are installed when you install the integration client: **File System** and **SQL Server**.

> ***Note:*** *Session data is stored on Windows file systems as XML files. Reading these files may prove difficult without a third-party XML parser. SQL Server databases make parsing and management of session data significantly more manageable. However, SQL Server databases cannot be used to store session recordings or site backups.*

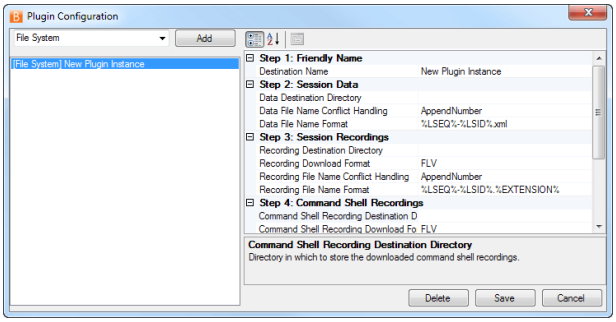
1. After the initial installation, click **Yes** to install your plugin.



2. If the integration client is already installed and you wish to update or add a plugin, select **Plugin Configuration** from the integration client **Setup** dropdown.



3. From the dropdown at the top of the plugin configuration dialog, select the type of plugin you would like to configure, and then click **Add.**
4. Specific directions for the standard plugins' configurations are detailed in the tables below. Configure the settings and click **Save**.



File System Plugin Settings

The table below details the fields on the right of the **Plugin Configuration** screen for the File System plugin.

Field Name	Description
Destination Name	The name to give this plugin instance. This name is used by the integration client for display and logging purposes.
Destination Directory	The directory in which to store the appropriate XML data, recordings, or site backups. Do not enter a mapped drive in the directory. Unless the directory is local you must enter a UNC path.
Download Format	Leave this at its default unless otherwise instructed by a BeyondTrust technician.
File Name Conflict Handling	The action to take when the file name already exists. See below for information about the handling options.
File Name Format	The format in which to create the file name for the appropriate XML data, recordings, or site backups. See below for special replacement values.
Retention Count	The number of prior backups to keep. Leave this field empty to keep all backups.

The table below describes the options available for **File Name Conflict Handling**.

Option	Description
AppendNumber	If the destination file name exists, then a new file will be created with a number inserted just before the last period in the file name. The value of the number is the smallest possible integer capable of guaranteeing that the file name is unique. For example, if the recording for session **LSID 1234** is downloaded and the file **support.example.com-support-1234.flv** already exists, then the file **support.example.com-support-1234.1.flv** will be created. Likewise, if **support.example.com-support-1234.1.flv** already exists, then **support.example.com-support-1234.2.flv** will be created.
Overwrite	If the destination file name already exists, then the contents of the existing file will be overwritten with the most recently downloaded data.
Skip	If the destination file name exists, then the existing file will NOT be overwritten, and no new file will be created.

Special Replacement Values

These are **File Name Format** replacement value fields. All special replacement values start and end with a percent sign (%). Only UPPER CASE characters and underscores (_) are valid characters between the percent signs. Replacement values are case sensitive.

Replacement Name	Description
%LSEQ%	An incrementing number that can be used if your application needs to represent support sessions in a non-string format. ***Note:*** *The LSEQ element is not guaranteed to be unique or strictly sequential.*
%LSID%	The session's unique string ID.
%EXTERNAL_KEY%	The session's external key.

Replacement Name	Description
%SESSION_TYPE%	The session's type name.
%SUPPORT_SITE_HOST%	The hostname of the support site from which the session data or the site backup was downloaded.
%INSTANCE%	The instance number of the command shell recording or Show My Screen recording. Because multiple shells can be run during a session, one session may have multiple command shell recordings. Likewise, multiple Show My Screen sessions may be run, resulting in multiple Show My Screen recordings.
%EXTENSION%	The file extension name.

Service Properties for Remote Locations

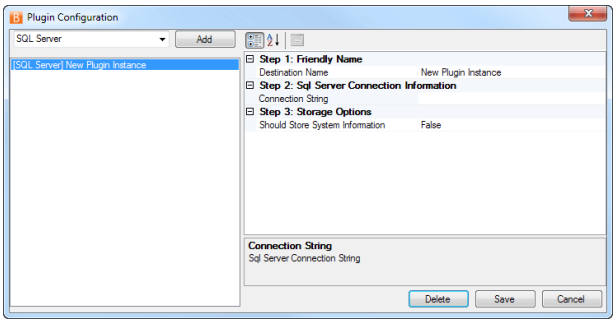
1. If you configured a plugin to save to a network drive or IP address, you will need to modify the BeyondTrust Integration Client service. Open your services management console by typing **services.msc** using your Windows **Run** dialog.

> ***Note:*** *Do not enter a mapped drive in the Plugin Configuration screen's* ***Destination Directory*** *field. Unless this directory is local you must use a UNC path.*

2. Right-click the BeyondTrust Integration Client scheduler service and select **Properties**.
3. Select the **Log On** tab and change the **Log on as** setting to **This account**, using an account with rights to the network location. This will most likely be a domain account.
4. Apply the changes and close the dialog.
5. Restart the BeyondTrust Integration Client scheduler service for the change to take effect.

SQL Server Plugin

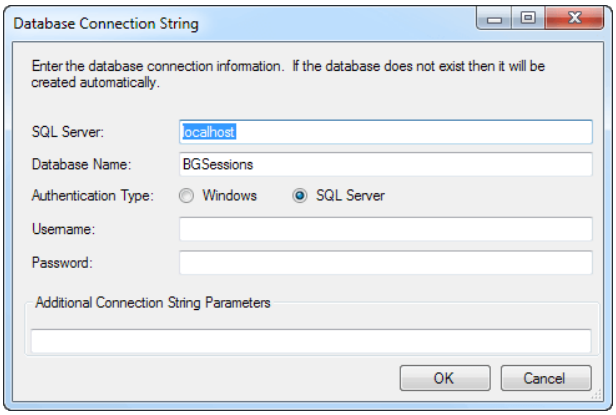
When you select to add a SQL Server plugin, the following screen is displayed.



The table below details the fields for the SQL Server plugin.

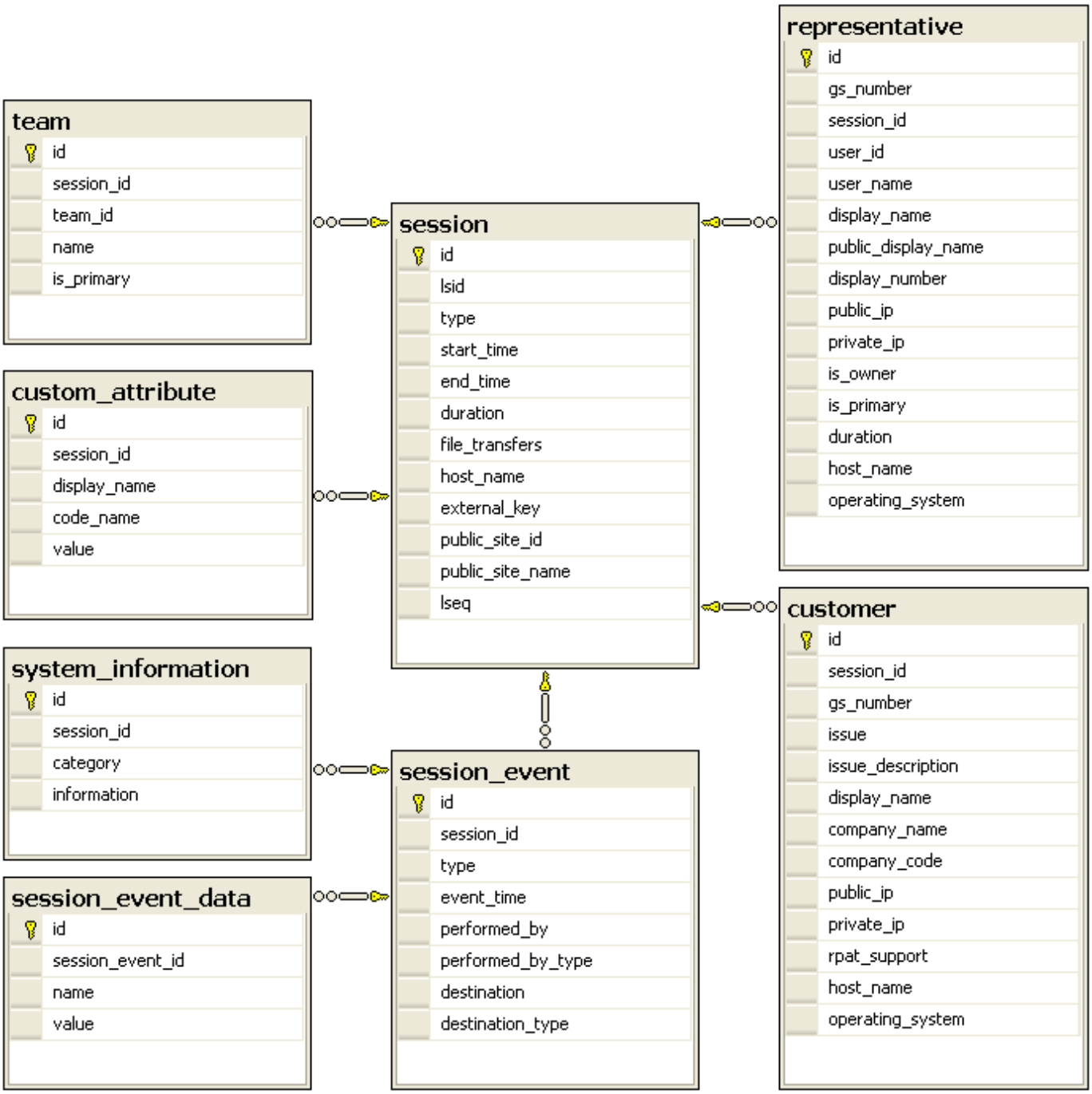
Name	Description
Destination Name	The name to give this plugin instance. This name is used by the integration client for display and logging purposes.
Connection String	The connection string used to connect to the database where session data will be stored.
Should Store System Information	**True** or **False**. Designates if the system information will be stored. This is the system information collected during a remote session.

Clicking the **[...]** button next to the **Connection String** setting will bring up another dialog. Enter your database connection information as required.



SQL Server Storage Database

The following diagram shows the tables that will be created in the SQL Server database with the standard SQL Server plugin. An explanation of each of the tables is detailed in the following pages.



Session Table

This table is the root of all information inserted into the database. Each row represents a BeyondTrust session.

id	An auto-incrementing number that uniquely identifies this field in the database.
lsid	An alphanumeric identification which uniquely identifies this session.
type	The type of session. Currently, **support** is the only value supported.
start_time	The time at which the session began either by the customer's running the customer client or by the representative's initiating a Jump session. Date is returned in UTC format.
end_time	The time at which the session ended by the representative's closing the session. Date is returned in UTC format. This field will be empty for sessions which are still in progress when the data was extracted or which closed abnormally.
duration	Session length in HH:MM:SS format.
file_transfers	The number of file transfers which occurred during the session.
host_name	The hostname of the BeyondTrust support site through which the session occurred.
external_key	An arbitrary string that can link this session to an identifier on an external system, such as a help desk ticket ID. This can be input from within the representative console or defined programmatically.
public_site_id	The identification of the site. This defaults to **1**.
public_site_name	The name of the BeyondTrust site. Unless set, this contains the value **Default**.
lseq	An incrementing number used to represent support sessions in a non-string format.  ***Note:*** *The LSEQ element is not guaranteed to be unique or strictly sequential.*

Session_Event Table

This table links to the **session** table via the **session_id** field. Each row in this table represents a session event that took place during this session. Sessions can have multiple session events.

id	An auto-incrementing number that uniquely identifies this field in the database.
session_id	The ID of the session in which this event occurred. This field links a **session_event** row to a **session** row.

type	The type of event that occurred. Event types include: Callback Button Deployed / Registry Exported Callback Button Removed / Registry Imported Chat Message / Registry Key Added Command Shell Session Started / Registry Key Deleted Conference Member Added / Registry Key Renamed Conference Member Departed / Registry Value Added Conference Member State Changed / Registry Value Deleted Conference Owner Changed / Registry Value Modified Customer Exit Survey / Registry Value Renamed Directory Created / Remote Shell Session Started External Key / Representative Exit Survey File Deleted / Service Access Allowed File Download / Session Assigned File Download Failed / Session Assignment Response File Moved / Session End File Upload / Session Note Added File Upload Failed / Session Start Files Shared / Show My Screen Recording Foreground Window Changed / System Information Retrieved Legal Agreement Response
event_time	The time at which the event occurred. The time is returned in UTC format.
performed_by	The name of the entity that performed the action.
performed_by_type	The type of entity that performed the action, indicating whether this action was performed by the **System**, a **Customer**, or a **Representative**.
destination	The name of the entity to whom this action was directed.
destination_type	The entity to which the event was directed, indicating whether this action was directed to the **System**, a **Customer**, or a **Representative**.

Session_Event_Data Table

This table links to the **session_event** table via the **session_event_id** field. Each row in this table represents a single key-value pair associated with a particular session event. Session events can have multiple key-value pairs.

id	An auto-incrementing number that uniquely identifies this field in the database.
session_event_id	The ID of the session event to which this key-value pair belongs. This field links a **session_event_data** row to a **session_event** row.

name	The key of this field.
value	The value of this field.

System_Information Table

This table links to the **session_event** table via the **session_event_id** field. Each row in this table represents multiple categories of system information collected per a **session_event**.

> ***Note:*** *System information is logged only when pulled automatically at the beginning of the session and not when specifically requested by the representative. This is to prevent overload with the large amount of dynamic data that can be retrieved from the remote system.*

id	An auto-incrementing number that uniquely identifies this field in the database.
session_event_id	The ID of the session event to which this system information belongs.
category	The type of system information. Types include categories, such as the following: **General, Memory, Drives, Processes, Events, Network,** and **Programs**. There are additional categories available. The category is based on the remote operating system.
information	Contains multiple **<field>** elements, each of which contains a descriptor for the specific data field. For example, the **Drives** category would have **<field>** elements **Drive, Type, Percent Used**, etc.

Representative Table

This table links to the **session** table via the **session_id** field. Each row in this table represents a representative who participated in this session. Sessions can have multiple representative fields.

id	An auto-incrementing number that uniquely identifies this field in the database.
gs_number	Uniquely identifies the representative regarding their current connection to the Secure Remote Access Appliance. A gsnumber is assigned on a per-connection basis, so if a representative leaves a session and then rejoins without logging out of the Secure Remote Access Appliance, their gsnumber will remain the same. However, if the representative's connection is terminated for any reason, when that representative logs back into the Secure Remote Access Appliance, they will be assigned a new gsnumber. A gsnumber may be recycled, so while two people connected at the same time will never have the same gsnumber, one person may have a gsnumber that was assigned to another person in the past.
session_id	The ID of the session in which this representative participated. This field links a **representative** row to a **session** row.
user_id	The unique ID assigned to the representative.
user_name	The username assigned to the representative.
display_name	The private display name assigned to the representative. Note that this field contains the private display name's value at the time of the session, which may not match the current value if the private display name has subsequently been changed.

public_display_name	The public display name assigned to the representative. Note that this field contains the public display name's value at the time of the session, which may not match the current value if the public display name has subsequently been changed.
display_number	The display number assigned to the representative. This is the display number at the time of the session and may not match the current value.
public_ip	The representative's public IP address.
private_ip	The representative's private IP address.
is_owner	Integer value (**1** or **0**) indicating whether the representative was an actual owner of the session or was merely a conference member.
is_primary	Integer value (**1** or **0**) indicating if the representative was the final representative to own the session.
duration	Integer value indicating the number of seconds the representative was involved in this support session.
host_name	The hostname of the representative's computer.
operating_system	The operating system of the representative's computer.

Customer Table

This table links to the **session** table via the **session_id** field. Each row in this table represents a customer who participated in this session. In the current BeyondTrust version, there will always be one customer field per session.

id	An auto-incrementing number that uniquely identifies this field in the database.
session_id	The ID of the session in which this customer participated. This field links a **customer** row to a **session** row.
gs_number	Uniquely identifies the customer regarding their current connection to the Secure Remote Access Appliance. A gsnumber may be recycled, so while two people connected at the same time will never have the same gsnumber, one person may have a gsnumber that was assigned to another person in the past. Can be used to correlate a **<customer>** element with a **<primary_cust>** or with an event's **<performed_by>** or **<destination>** element.
issue	The numeric ID of the issue or the representative which the customer selected from the dropdown of the issue submission form or which was designated programmatically.
issue_description	The description of the problem as entered by the customer in the **Describe Your Issue** text area field of the issue submission form or as programmatically assigned.
display_name	The display name is the name the customer provided in the issue submission form. If no name was provided, then this is populated with the name associated with the user logged into the customer's computer.
company_name	The company name which the customer entered in the **Company** field on the issue submission form or which was programmatically assigned.

company_code	The code which the customer entered in the **Company Code** field on the issue submission form or which was programmatically assigned.
public_ip	The customer's public IP address.
private_ip	The customer's private IP address.
rpat_support	Integer value (**1** or **0**) indicating whether the customer session is provided via RPAT.
host_name	The hostname of the customer's computer.
operating_system	The operating system of the customer's computer.

Team Table

This table links to the **session** table via the **session_id** field. Each row in this table represents a support team queue to which this session was assigned. Sessions can have multiple team fields.

id	An auto-incrementing number that uniquely identifies this field in the database.
session_id	The ID of the session of which this team was an owner. This field links a **team** row to a **session** row.
team_id	The unique ID assigned to this support team.
name	The display name of the support team. Note that this field contains the team name as it currently appears, which may not match the value at the time of the session if the team name has been subsequently changed.
is_primary	Integer value (**1** or **0**) indicating if this support team was the last team to which the session was transferred.

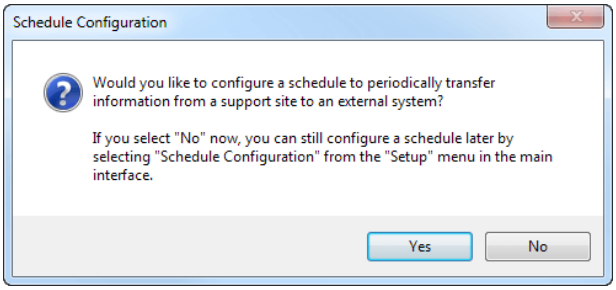
Custom_Attribute Table

This table links to the **session** table via the **session_id** field. Each row in this table represents a custom attribute which was assigned to this session. Sessions can have multiple custom attribute fields.

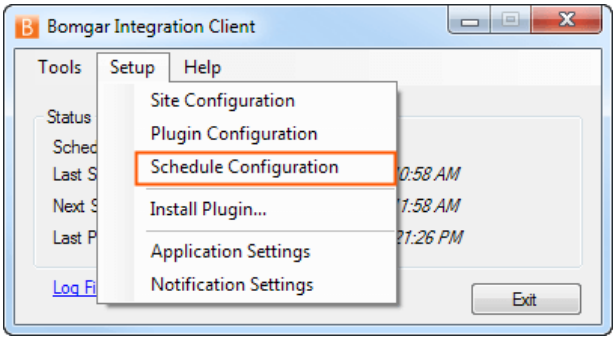
id	An auto-incrementing number that uniquely identifies this field in the database.
session_id	The ID of the session to which this custom attribute was assigned. This field links a **custom_attribute** row to a **session** row.
display_name	The display name of the custom attribute.
code_name	The code name of the custom attribute.
value	The value of the custom attribute. This may have been provided by the customer or assigned programmatically.

Set the Integration Client Schedule

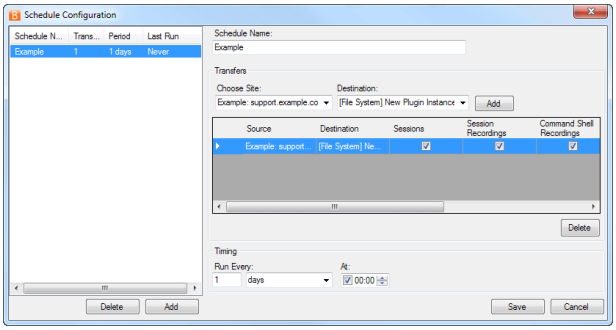
1. It is generally a best practice to set a schedule for the integration client to run periodically. Upon initial installation, you are also prompted to set up a schedule. When prompted, click **Yes.**



2. If the integration client is already installed and you wish to update or add a schedule, select **Schedule Configuration** from the **Setup** dropdown.



3. From the schedule configuration dialog, schedule a data transfer by clicking the **Add** button.
4. Enter a descriptive name in the **Schedule Name** field.
5. From the **Choose Site** dropdown, select a configured BeyondTrust site to act as a source from which to pull data.
6. From the **Destination** dropdown, select a plugin instance to act as a destination for the transfer.
7. Click **Add** to create this data transfer instance. A new schedule transfer row will be added to the table beneath. You can add multiple transfer instances to one schedule if you wish the transfers to occur simultaneously.



8. For each transfer instance, select the types of data that you wish to transfer. Recordings and site backups can only be saved to a file and not to a SQL Server database. It is recommended to create separate schedules for backups and data extracts.

> ***Note:*** *Among the five types of data the BeyondTrust integration client can download (session data, session recordings, presentations, presentation recordings, and site backups). Session data, session recordings, and site backups can all be stored on a Windows file system. Session and presentation data is stored as .xml files, session and presentation recordings are stored as .flv files, and site backups are stored as proprietary .nsb files. The first two types of files can be parsed with XML or Flash video tools as appropriate. The .nsb files can only be read by Secure Remote Access Appliances via upload from the* ***/login > Management > Software Management*** *page in the* ***Software :: Backup Settings*** *section.*

9. The **Run Every** setting determines how often these transfers should occur. Transfers can be scheduled from every minute to every seven days. Transfers that are scheduled in increments of days can be set for a particular time. This could allow the

transfer to run when more server resources and bandwidth are expected to be available, such as during non-production hours.

10. Click **Save** to finish configuring this schedule. You can add multiple schedules to run multiple transfers.
11. After configuring a schedule, it will appear in the left pane along with a summary of information:
    - **Schedule Name:** The descriptive name given to this schedule.
    - **Transfers:** The number of transfer instances this schedule initiates. This is equal to the number of rows in the **Transfers** section. One transfer instance may transfer multiple types of data.
    - **Period:** The length of time scheduled between each transfer.
    - **Last Run:** When the scheduled transfer last was run.

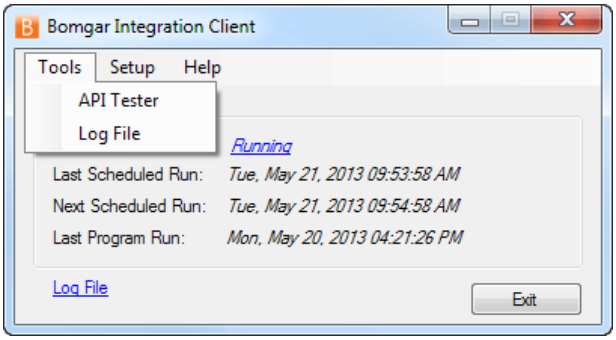
> ***Note:*** *If the integration client cannot connect to the Secure Remote Access Appliance or to its transfer destination, it will still mark the schedule complete and update the last run date. If the integration client itself experiences an error, it will neither mark the schedule complete nor update the last run date.*

IMPORTANT!

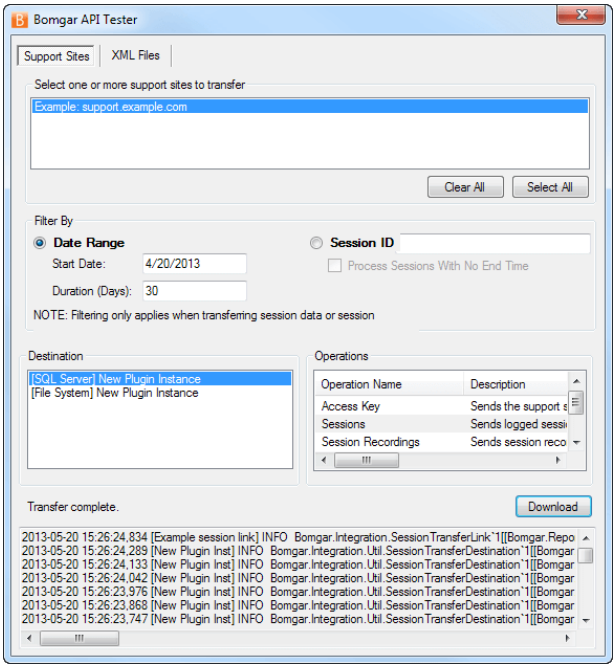
Ensure that the clock on the server hosting the integration client is not ahead of the clock on the Secure Remote Access Appliance from which data is being extracted. If the server clock is ahead, some data may not be downloaded. The best way to ensure that the integration client's host server and the Secure Remote Access Appliance have synchronized clocks is to use the same network time protocol (NTP) server for both.

Test the Integration Client

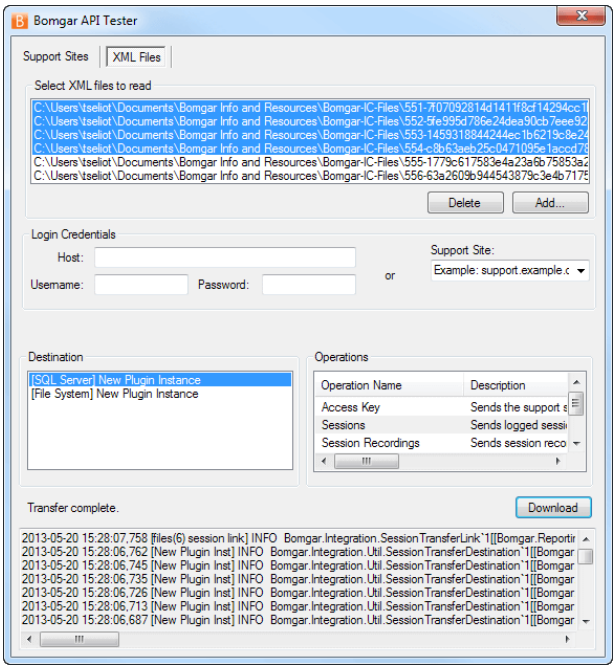
1. Once you have at least one plugin and the support site configured, it is recommended that you verify the configuration. Open the integration client from its directory location (**C:\Program Files\Bomgar\Integration** by default) and then run the **API Tester** from the **Tools** menu.



2. To test the database configuration, select one or more sites to verify from the list of configured sites.
3. Choose either a start date and duration for which to pull data or enter a specific session ID number.
4. Select the destinations you would like to test.
5. Finally, from the list of plugin operations, select one or more types of data to transfer.
6. Once you click the **Download** button, the API tester will begin transferring data based on the parameters you defined. Once the transfer is finished, verify that the appropriate information was successfully transferred to the selected destinations.



7. Alternatively, test your plugin settings by parsing data from previously downloaded XML files. From the **XML Files** tab, click the **Add** button to browse to one or more BeyondTrust XML files and then select the files you would like to parse.
8. If you choose to transfer any type of data other than session details (session recordings, Show My Screen recordings, command shell recordings, access keys, or site backups), the API tester will need to connect back to the BeyondTrust site. Enter either the appropriate hostname and credentials for the site from which the data was extracted or select the appropriate site from the dropdown of configured sites.
9. Select the destinations you would like to test.
10. Finally, select one or more types of data to transfer from the list of operations.
11. Once you click the **Download** button, the API tester will begin parsing the XML files via the methods you specified. Once the transfer is finished, verify that the appropriate information was successfully transferred to the selected destinations.

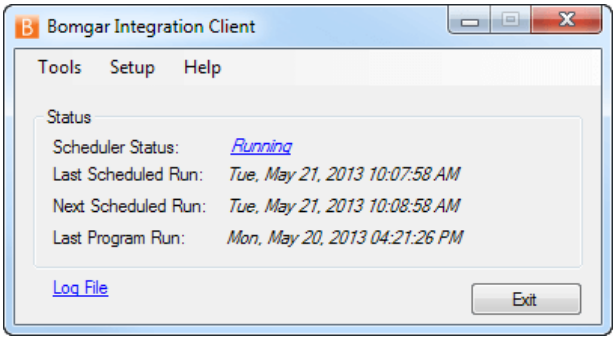


Integration Client Tools

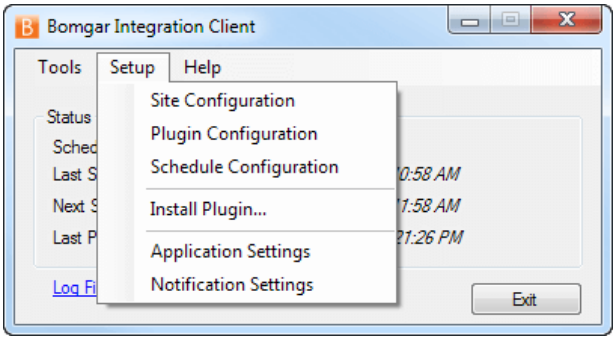
Once you have finished setting up your integration client, you can start it from **Start > Programs > BeyondTrust > Integration**.

> ***Note:*** *The integration client should always be run as administrator if the configuration needs to be changed. Otherwise, file permission access errors are likely to occur.*

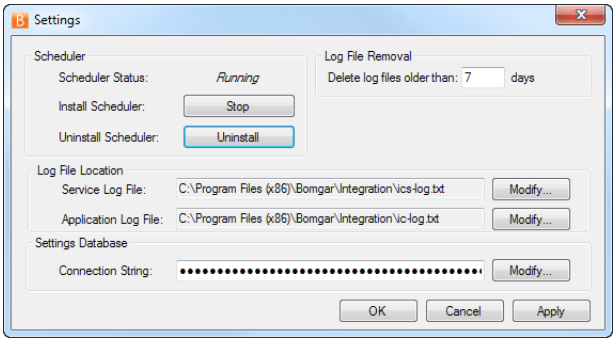
The integration client shows whether your scheduler is running, stopped, or is not installed. Every minute, the scheduler will check to see if it has any transfers to perform. **Last Program Run** displays the last time that the integration client itself was invoked.



From the **Setup** menu you can modify your site, plugin, or schedule configurations. You can also install additional plugins, such as in-house/proprietary applications or third-party applications. Finally, you can set integration client application and notification settings.

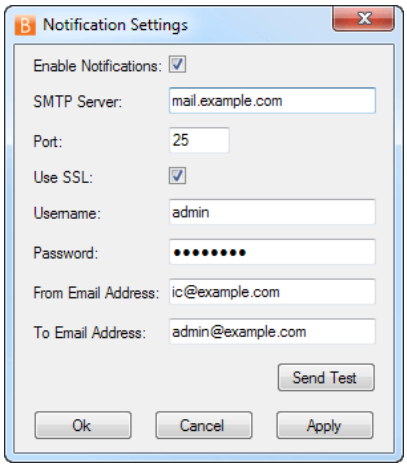


From **Application Settings** you can start and stop the scheduler, uninstall the scheduler, and change log file directory locations. You can also set the **Log File Removal** field to delete the integration client's activity logs after a certain number of days to save disk space and make review of activity logs more manageable.



From the Notification Settings window, you can set notification parameters. Notifications are sent any time the integration client logs an error.

If you wish to verify that your SMTP settings are accurately configured, click **Send Test**.



To view a log of the integration client's activity, click **Log File** at the bottom of the integration client window or select it from the **Tools** menu. The **IC Log** tab shows activity within the integration client tool itself, while the **Service Log** tab shows activity within the scheduler service.