Remote Support
HEAT Software Integration
BeyondTrust Remote Support Integration with HEAT Software

IMPORTANT!

You must purchase this integration separately from both your Remote Support software and your HEAT Software solution. For more information, contact BeyondTrust sales.

Support and IT organizations using HEAT software can integrate with BeyondTrust Remote Support to improve service levels, centralize support processes, and strengthen compliance. This document describes the installation and configuration of the BeyondTrust Remote Support integration with HEAT.

The integration between HEAT and BeyondTrust Remote Support enables service desk technicians to launch a secure remote support session directly from within HEAT.

The HEAT integration with BeyondTrust Remote Support provides the following functionality:

- A self-service user can start a BeyondTrust click-to-chat session directly from an incident within HEAT.
- A representative can generate a session key directly from an incident, change, or service request.
- Details of all BeyondTrust sessions initiated from an incident, change, or service request context are linked to the associated item and are viewable as a list as well as in a detailed view on each item type.
- While in a BeyondTrust session linked to a HEAT item, a custom link in the BeyondTrust representative console allows a representative to launch a browser window directly to the HEAT item in order to view additional information.
- A BeyondTrust Jump session can be initiated to access a configuration item from an incident, with the Jump session details being linked to the incident.
- If a session originates outside the normal HEAT workflow, with a ticket for the incident already existing in HEAT, the representative can manually link the session to the HEAT ticket to ensure all details are captured in the HEAT system.
- With assistance from BeyondTrust Professional Services, the integration can automatically generate incidents in HEAT when a session is received, thus saving a number of steps for the support representative.
Prerequisites for the BeyondTrust Remote Support Integration with HEAT Software

To complete this integration, please ensure that you have the necessary software installed and configured as indicated in this guide, accounting for any network considerations.

Applicable Versions

- BeyondTrust Remote Support: 14.x and newer
- HEAT Software: 2015.2 and newer (includes cloud and on-premises releases)

Network Considerations

The following network communication channels must be open for the integration to work properly.

<table>
<thead>
<tr>
<th>Outbound From</th>
<th>Inbound To</th>
<th>TCP Port #</th>
<th>Purpose</th>
</tr>
</thead>
<tbody>
<tr>
<td>BeyondTrust Middleware Engine Server</td>
<td>HEAT Software</td>
<td>443</td>
<td>API calls from the BeyondTrust Middleware Engine server.</td>
</tr>
<tr>
<td>BeyondTrust Middleware Engine Server</td>
<td>Secure Remote Access Appliance</td>
<td>443</td>
<td>API calls from the BeyondTrust Middleware Engine server.</td>
</tr>
<tr>
<td>Secure Remote Access Appliance</td>
<td>BeyondTrust Middleware Engine Server</td>
<td>8180 (default) 443 (optional)</td>
<td>The BeyondTrust Middleware Engine server receives outbound events from the appliance. However, if polling is used instead of outbound events, then this port does not have to be open.</td>
</tr>
</tbody>
</table>

Prerequisite Installation and Configuration

The HEAT Software integration is a BeyondTrust Middleware Engine plugin.

For more information on installing and working with the BeyondTrust Middleware Engine, please see the BeyondTrust Remote Support Middleware Engine Installation and Configuration document at [www.beyondtrust.com/docs/remote-support/how-to/integrations/middleware-engine](www.beyondtrust.com/docs/remote-support/how-to/integrations/middleware-engine).
Configure HEAT Software for Integration with BeyondTrust Remote Support

**Note:** Before importing the package, use a text editor to ensure all URLs to the Secure Remote Access Appliance have been updated. To do so, search for support.example.com and replace any instances with the hostname of the appliance.

Two development packages are provided as part of the integration:

- BeyondTrust Standard Integration - Core - 1.x.x.MetadataPatch
- BeyondTrust Standard Integration - UI Elements - 1.x.x.MetadataPatch

**IMPORTANT!**

Before applying any development package, we recommend that you first back up the database in case changes need to be rolled back.

Always import the **Core** package first, as this contains all of the necessary business objects, quick actions, etc. needed by the integration.

Once the contents of the **Core** development package have been verified:

1. Make sure you have taken the steps described in the note above to find and replace the example Secure Remote Access Appliance hostname in the package file.
2. Log into the HEAT tenant as an administrator and access the admin UI.
3. Under **Build > HEAT Development Package > Package**, select **Import Package** and browse to the file.

When the development package is successfully imported, a new **BeyondTrust API Role** should be listed under Configure > Users and Permissions > Roles.

1. Select the role and add a new employee to be used for API calls.
2. Ensure that **Internal Auth** is enabled.
3. Ensure that an **Internal Auth Password** is set.
4. Ensure that **Password Expiration** is disabled.

Next, determine whether the **UI Elements** package can be used or whether the changes must be made manually. Because the package makes changes to various forms and layouts, it can cause issues when importing the package in an environment in which you have made changes to any of the following forms or layouts:

1. **Business Objects > Change# > Layouts > Change > formView**
2. **Business Objects > Change# > Layouts > Change.SDA > formView**
3. **Business Objects > Change# > Layouts > Change.SDM > formView**
4. **Business Objects > ServiceReq# > Layouts > ServiceReqLayout.New > formView**
5. **Business Objects > Incident# > Layouts > IncidentLayout.ServiceDesk > formView**
6. **Business Objects > Incident# > Forms > IncidentDetails.ServiceDesk**
7. **Business Objects > Incident# > Forms > Incident.WebSelfService.Edit**
Only in the case that all of the above items are still in their out-of-the-box state, repeat the earlier steps used to import the Core package, but this time, import the UI Elements package.

If any of the items have been modified, use these steps to manually create the UI elements that would have been created by the package:

1. Under Build > Business Objects, select Change > Layouts, and then click the Change layout.
2. In the Views in this layout grid, select formView to edit the view.
3. Click the Edit link to the right of the Toolbar section.
4. In the Toolbar Editor, find and select the BeyondTrust - Generate Session Key action from the list at the bottom and drag it into position as the first item in the right half of the toolbar menu.
5. Configure the new toolbar button as follows:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>BeyondTrust - Generate Session Key</td>
</tr>
<tr>
<td>Display</td>
<td>Image and Text</td>
</tr>
<tr>
<td>Image</td>
<td>icon_Bomgar.png</td>
</tr>
<tr>
<td>Text (Custom)</td>
<td>BeyondTrust Session Key</td>
</tr>
<tr>
<td>Tooltip</td>
<td>Click to generate a BeyondTrust session key</td>
</tr>
</tbody>
</table>

6. Click Save on the Button Editor and Save again on the Toolbar Editor.

7. Scroll to the bottom of the Child Panels grid and click the link to Add Child Panel.
8. Configure the new panel as follows:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Name</td>
<td>BeyondTrust Sessions</td>
</tr>
<tr>
<td>Object</td>
<td>BomgarSession()</td>
</tr>
<tr>
<td>List</td>
<td>BomgarSession().isGrid</td>
</tr>
<tr>
<td>Forms(s)</td>
<td>BomgarSession().ViewOnlyForm</td>
</tr>
<tr>
<td>Show Count</td>
<td>[Checked]</td>
</tr>
</tbody>
</table>

9. Click the Edit link in the Toolbar column and drag all of the toolbar buttons for the panel to the trash so that no buttons show on the BeyondTrust Sessions panel.

10. Click Save at the top to save all changes to the layout.

11. Repeat steps a-i for the Change.SDA and Change.SDM layouts and/or any custom layouts you may be using instead of these default layouts.

12. Under Build > Business Objects > Service Request, select the Service Request business object, and then repeat steps a-i for the ServiceReqLayout.New layout and/or any custom layouts you may be using instead of these default layouts.

13. Under Build > Business Objects > Incident, select the Incident business object, and then repeat steps a-i for the IncidentLayout.ServiceDesk layout and/or any custom layouts you may be using instead of these default layouts.

14. With the Incident business object still selected, swap from Layouts to Forms, and then select the IncidentDetails.ServiceDesk form.

15. Check the box to Show layout cells, and select the cell containing the Assets control.

16. Click the link above the form to Insert Row Above.

17. Expand the toolbar to the left of the form to show Incident > Fields, and then drag the CILink field into the newly created row.
18. Click the new control to select it, and then configure it as follows:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Display Field</td>
<td>[CI]Name</td>
</tr>
<tr>
<td>Editable</td>
<td>true</td>
</tr>
<tr>
<td>Label Pos</td>
<td>On the top</td>
</tr>
</tbody>
</table>

19. Collapse the Incident tree in the toolbar, expand it to show Other items, and then drag a URL Button to the right of the CILink field.
20. Select the new button and configure it as follows:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Image</td>
<td>icon_Bomgar.png</td>
</tr>
</tbody>
</table>

**Note:** Use the Image Manager tool in the Expression Editor to select the image.

21. Click **Save** at the top to save changes to the form.

22. Select the **Incident.WebSelfService.Edit** form.

23. Expand the toolbar to show **Other** items, and then drag a **Command Button** to the right of the **Close Incident** button on the form.
24. Select the new button and configure it as follows:

<table>
<thead>
<tr>
<th>Field Name</th>
<th>Value</th>
</tr>
</thead>
<tbody>
<tr>
<td>Action</td>
<td>BeyondTrust - Get Help Now</td>
</tr>
<tr>
<td>CausesValidation</td>
<td>false</td>
</tr>
<tr>
<td>Image</td>
<td>icon_Bomgar.png</td>
</tr>
</tbody>
</table>

**Note:** Use the Image Manager tool in the Expression Editor to select the image.

25. Click **Save** at the top to save changes to the form.
Configure BeyondTrust Remote Support for the HEAT Software Integration

Several configuration changes are necessary on the Secure Remote Access Appliance to integrate with HEAT Software. You must make these changes on each appliance for which you intend to create a plugin configuration, described in "Configure the HEAT Plugin for Integration with BeyondTrust Remote Support" on page 14.

All of the steps in this section take place in the BeyondTrust /login administrative interface. Access your Remote Support interface by going to the hostname of your Secure Remote Access Appliance followed by /login (e.g., https://support.example.com/login).

Verify the API Is Enabled

This integration requires the BeyondTrust XML API to be enabled. This feature is used by the BeyondTrust Middleware Engine to communicate with the BeyondTrust APIs.

Go to /login > Management > API Configuration and verify that Enable XML API is checked.

Create an API Service Account - BeyondTrust 16.1 and Earlier

The API user account is used from within the integration to make BeyondTrust Command API calls to BeyondTrust.

2. Click Create New User and name it Integration or something similar.
3. Leave Must Reset Password at Next Login unchecked.
5. Set Allowed to View Support Session Reports to View All Sessions.
6. Check Allowed to view support session recordings.
7. Set Allowed to View Presentation Session Reports to View All Sessions.
8. Check Allowed to Use Reporting API and Allowed to Use Command API.
9. Scroll to the bottom and save the account.
Create an API Service Account - BeyondTrust 16.2 and Later

1. Go to Management > API Configuration and click Add to create a new API account.
2. Provide a name for the API account.
3. Under Permissions, check Full Access to the Command API.
4. For the Reporting API, check Allow Access to Support Session Reports and Recordings and Allow Access to Presentation Session Reports and Recordings.
5. Be sure to copy the values for both the OAuth Client ID and OAuth Client Secret for use in a later step.
6. Click Save to create the account.

Add an Outbound Event URL

1. Go to /login > Management > Outbound Events.
2. In the HTTP Recipients section, click Add and name it Integration or something similar.
3. Enter the URL to use:
   - If using an appliance ID of "default":
     http://<middleware-host>:<port>/ERSPost. The default port is 8180.
   - If using an appliance ID other than "default":
     http://<middleware-host>:<port>/ERSPost?appliance=<appliance-id> where
     <middleware-host> is the hostname where the BeyondTrust Middleware Engine is installed. The default port is 8180. The <appliance-id> is an arbitrary name, but note the value used, as it is entered later in the plugin configuration. This name accepts only alphanumeric values, periods, and underscores.
4. Scroll to Events to Send and check the following events:
   - Support Session End
   - Customer Exit Survey is Completed
   - Representative Survey is Completed
   - Someone Joins a Support Session (Optional)
5. Click Save.
6. Now, the list of outbound events should contain the event just added. The Status column displays a value of OK if
communication is working. If communication is not working, the **Status** column displays an error which you can use to repair communication.

**Add Custom Fields**

Under **Configuration > Custom Fields**, add two new custom fields. Enter the following values:

<table>
<thead>
<tr>
<th>Display Name</th>
<th>Code Name</th>
<th>Show in Rep Console</th>
</tr>
</thead>
<tbody>
<tr>
<td>Item ID</td>
<td>item_id</td>
<td>Checked</td>
</tr>
<tr>
<td>Item Type</td>
<td>item_type</td>
<td>Checked</td>
</tr>
</tbody>
</table>

**Add a Custom Link**

BeyondTrust custom links can be configured to allow representatives to quickly access the HEAT item that is associated with the session.

**BeyondTrust 15.1 and later**

1. Browse to **Rep Console > Custom Links**.
2. Click **Add**.
3. Enter a name for the link, and then set the URL to
   
   [https://support.example.com/files/bomgarCustomLink.html?view=1&externalKey=%SESSION.CUSTOM.EXTERNAL_KEY%](https://support.example.com/files/bomgarCustomLink.html?view=1&externalKey=%SESSION.CUSTOM.EXTERNAL_KEY%) where `support.example.com` is your BeyondTrust site hostname.

   **Note:** The page referenced in the custom link URL does not yet exist but is created at the end of the section “Configure the HEAT Plugin for Integration with BeyondTrust Remote Support” on page 14.

4. Click **Save** to save the new link.
Configure the HEAT Plugin for Integration with BeyondTrust Remote Support

Now that you have configured HEAT Software and the Secure Remote Access Appliance, deploy and configure the HEAT plugin.

For more information on installing and working with the BeyondTrust Middleware Engine, please see the BeyondTrust Remote Support Middleware Engine Installation and Configuration document at www.beyondtrust.com/docs/remote-support/how-to/integrations/middleware-engine.

1. Copy the provided plugin zip file to the server hosting the BeyondTrust Middleware Engine.
2. Extract the plugin zip file to the Plugins folder in the directory where the BeyondTrust Middleware Engine is installed.
3. Restart the BeyondTrust Middleware Engine Windows service.
4. From the server, launch the middleware administration tool. The default URL is http://127.0.0.1:53231.
5. The HEAT Plugin shows in the list of plugins. Click the clipboard icon to add a new configuration.

Secure Remote Access Appliance

The first portion of the plugin configuration provides the necessary settings for communication between the plugin and the Secure Remote Access Appliance. The configuration sections include:

1. **Plugin Configuration Name**: Any desired value. Because multiple configurations can be created for a single plugin, allowing different environments to be targeted, provide a descriptive name to indicate how this plugin is to be used.
2. **Appliance ID**: This can be left as **Default** or can be given a custom name. This value must match the value configured on the outbound event URL in the Secure Remote Access Appliance. If outbound events are not being used, this value is still required, but any value may be used.
3. **Secure Remote Access Appliance Host Name**: The hostname of the Secure Remote Access Appliance. Do not include https:// or other URL elements.
4. **BeyondTrust Integration API OAuth Client ID**: When using API accounts in BeyondTrust Remote Support 16.2.1 or newer, this field should contain the Client ID of the OAuth account.
5. **BeyondTrust Integration API OAuth Client Secret**: When using API Accounts available in BeyondTrust Remote Support 16.2.1 or newer, this field should contain the client Secret of the OAuth account.
6. **BeyondTrust Integration API User Name**: If using a BeyondTrust Remote Support version prior to 16.2.1, this field should contain the username of the API service account created on the Secure Remote Access Appliance.
7. **BeyondTrust Integration API Password**: If using a BeyondTrust Remote Support version prior to 16.2.1, this field should contain the password of the above user.
8. **Locale Used for BeyondTrust API Calls:** This value directs the Secure Remote Access Appliance to return session data in the specified language.

9. **Disabled:** Enable or disable this plugin configuration.

10. **Allow Invalid Certificates:** Leave unchecked unless there is a specific need to allow. If enabled, invalid SSL certificates are allowed in calls performed by the plugin. This would allow, for example, self-signed certificates. This is not recommended in production environments.

11. **Use Non-TLS Connections:** Leave unchecked unless it is the specific goal to use non-secure connections to the Secure Remote Access Appliance. If checked, TLS communication is disabled altogether. If non-TLS connections are allowed, HTTP access must be enabled on the BeyondTrust /login > Management > API Configuration page. Using non-secure connections is discouraged.

   **Note:** When using OAuth authentication, TLS cannot be disabled.

12. **Outbound Events Types:** Specify which events the plugin processes when received by the middleware engine. Keep in mind that any event types selected here must also be configured to be sent in BeyondTrust. The middleware engine receives any events configured to be sent in BeyondTrust but passes them off to the plugin only if the corresponding event type is selected in this section.

   a. **Support Session End**
   b. **Customer Exit Survey is Completed**
   c. **Representative Survey is Completed**

13. **Polling Event Types:** If network constraints limit connectivity between the Secure Remote Access Appliance and the middleware engine such that outbound events cannot be used, an alternative is to use polling. The middleware engine regularly polls the Secure Remote Access Appliance for any sessions that have ended since the last session was processed. At this time, only the **Support Session End** event type is supported.

   **Note:** One caveat to polling behavior versus the use of outbound events is that if a session has ended but the customer exit survey has not yet been submitted within the same polling interval, the customer exit survey is not processed. This does not apply to representative surveys since the session is not considered to be complete if a representative survey is still pending.

14. **Polling Interval:** Enter only if polling is used. This determines how often the middleware engine polls the Secure Remote Access Appliance for sessions that have ended.

15. **Retry Attempt Limit:** Enter the number of retries that can be attempted if the plugin fails to process an event.

16. **Retry Outbound Event Types:** Specify which outbound events the plugin retries if it fails to process the event.

17. **Retry Polling Event Types:** Specify which polling events the plugin retries if it fails to process the event.

**HEAT Software Instance**

The remainder of the plugin configuration provides the necessary settings for communication between the plugin and the HEAT Software instance. The configuration settings include:
1. **HEAT SDK Web Services URL**: The SDK Web Services URL for the HEAT instance (e.g., https://heat.example.com/ServiceAPI/FRSHEATIntegration.asmx)

2. **HEAT Username**: The username of the API account.

3. **HEAT Password**: The password of the above user.

4. **HEAT Tenant ID**: The ID of the tenant instance targeted by the integration. If unknown, you can find this by logging into the ConfigDB web interface and viewing the Tenants list. The Tenant ID is listed as the Login URL in the list and detail views.

5. **Enable Automatic Incident Creation on Session Start**: If checked, the plugin processes `support_conference_member_added` events and the external key to determine whether to create a ticket within HEAT or not. The plugin attempts to create the ticket only if this setting is enabled, if the conference member joining the conference is a representative, and if the external key is a JSON string.

6. **Ticket Owner Info Source**: By default, the integration uses the BeyondTrust representative's username to identify them as the owner of the newly created ticket. Optionally, it can also use the representative's public display name in various formats or ignore the representative's information altogether, instead always using the default value supplied in the JSON string configured for **Ticket Default Data**.

7. **Ticket Default Data**: A JSON string containing values that can be used to prepopulate certain fields on the newly created ticket.

Save the configuration.

**Test Settings and Generate HTML Content**

Once the proper configuration has been entered and saved, use the tool's **Test this Plugin Configuration** function to verify the settings. In addition to verifying settings for both BeyondTrust and HEAT, the page outputs a snippet of HTML that is used in the creation of an intermediate page used for custom links. Once the settings are successful and the markup is generated, create the custom link as described below.
Create the Custom Link Page

1. Create an empty text file using Notepad or another text editor and copy and paste the HTML content from generated content on the plugin test page into the file.
2. Save the file as BeyondTrustCustomLink.html.
3. Log into the BeyondTrust/login interface and upload the file to the file store under Public Portals > File Store.

Report Templates

On the BeyondTrust Middleware Engine server, in the `<install dir>/Plugins/<integration>/Templates` folder, there are multiple files ending with `.hbs`. These files are used by the plugin to format the textual session report and exit surveys that are added to the corresponding ticket each time a BeyondTrust session ends or each time a survey is submitted. The templates can be edited if desired.

Note: If changes need to be made to a template, it is a good idea to first back up the original in case the changes ever need to be reverted.

For more information on Handlebars templates, please see handlebarsjs.com.
Use Cases for the HEAT Software Integration with BeyondTrust Remote Support

Generate Session Key

Support staff can generate a session key that can be given to the end user over the phone or via email to initiate a support session that is automatically associated with the selected ticket.

Import BeyondTrust Session Data into Ticket
Once the session ends, the ticket is automatically updated with information gathered during the session, including:

- **Chat Transcript** (including files transferred, special actions, and other events)
- **System Information** (the General section plus other select details such as disk, memory, and network)
- **Session Notes**
- **Surveys** (customer and representative)

**Jump to Configuration Item**

Support staff can leverage BeyondTrust Jump Technology to access a configuration item associated with a ticket directly from the HEAT ticket. By default, the button attempts to Jump to a pinned Jump Client, but the URL can be modified to perform one of the other Jump types as needed.
Click-to-Chat for Self Service Users

Self Service users can open their submitted tickets and start a chat support session directly from the HEAT ticket. This allows the user the quickest path to resolution while also providing the representative with the necessary context to assist the user. Sessions can be elevated to full support sessions if enabled and when necessary.

Access Ticket from Representative Console

Using BeyondTrust's custom links ability, a representative can access the associated ticket directly from within the representative console. This saves time searching for the ticket in HEAT and provides the representative with any available issue details, history, or other context to help quickly resolve the issue.

Manually Associate Sessions with Tickets

Whether a representative has just created a ticket for the current session or has found that one already exists, even sessions originating outside the scope of a HEAT item can be manually associated with the appropriate item, allowing session details to be automatically added to the ticket when the session ends.

To make this association, the representative enters the numeric ID of the Change, Incident, or Service Request into the External Key field while in session. Because these numeric IDs are not unique across business objects in HEAT, the representative can indicate the appropriate item type with a -c (Change), -i (Incident), or -s (Service Request) appended to the end of the number. For example, 15302-c would be associated with Change #15302.

Automatic Ticket Creation

Simplify workflows and reduce the number of clicks necessary for a support representative by automatically creating a ticket from a session.

Note: This feature requires a separate services engagement to implement.