Business Value Dashboard

Software Version: 10.63

Integration Guide

Document Release Date: December 2017
Software Release Date: November 2017

Hewlett Packard
Enterprise

Legal Notices

Warranty

The only warranties for Seattle SpinCo, Inc and its subsidiaries (“Seattle”) products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. Seattle shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Except as specifically indicated, valid license from Seattle required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notice

Trademark Notices

Adobe® and Acrobat® are trademarks of Adobe Systems Incorporated.

AMD, the AMD Arrow symbol and ATI are trademarks of Advanced Micro Devices, Inc.

Citrix® and XenDesktop® are registered trademarks of Citrix Systems, Inc. and/or one more of its subsidiaries, and may be registered in the United States Patent and Trademark Office and in other countries.

Google™ and Google Maps™ are trademarks of Google Inc.

Intel®, Itanium®, Pentium®, and Intel® Xeon® are trademarks of Intel Corporation in the U.S. and other countries.

iPad® and iPhone® are trademarks of Apple Inc.

Java is a registered trademark of Oracle and/or its affiliates.

Linux® is the registered trademark of Linus Torvalds in the U.S. and other countries.

Microsoft®, Windows®, Lync®, Windows NT®, Windows® XP, Windows Vista® and Windows Server® are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

NVIDIA® is a trademark and/or registered trademark of NVIDIA Corporation in the U.S. and other countries.

Oracle is a registered trademark of Oracle Corporation and/or its affiliates.

Red Hat® is a registered trademark of Red Hat, Inc. in the United States and other countries.

SAP® is the trademark or registered trademark of SAP SE in Germany and in several other countries.

UNIX® is a registered trademark of The Open Group.

Contents

Integration Guide

BVD provides out-of-the-box integrations with HPE Software products such as Operations Manager i (OMi) and Operations Connector (OpsCx).

However, you can create your own integrations for any data source by writing an adapter for BVD. The adapter must convert the source data to JSON and send the JSON-enabled data to the BVD data receiver. Predefined dashboards, which are designed and configured to display the received data, then complement your integration.

This guide includes the following topics:

- "Integrate with OMi" on page 5
- "Integrate with OpsCx" on page 9
- "Create custom integrations" on page 14

Integrate with OMi

You can configure OMi 10.11 or later versions to send Event Status, KPI Status and Performance Dashboard data to BVD. Use the BVD Data Forwarding rules manager on the OMi server to specify which data you want to forward.

Support matrix

For a list of supported OMi versions including required hotfixes, see the support matrix:

Support Matrices for Operations Center products

Download and extract the support matrix files, open the document SUMA.htm and select **Operations Manager i Business Value Dashboard** from the product list.

Data forwarding

The following OMi data can be forwarded to BVD:

Event Status data: The event status data to be forwarded is collected from an OMi monitoring dashboard that you specify.

KPI Status data: The KPI status is collected from all CIs that are associated with a view and the KPI set that you specify. If you do not specify a KPI set, all KPIs of the chosen view are forwarded.

Performance Dashboard data: The performance dashboard data is collected from your public favorites in OMi. To forward Performance Dashboard data, save your performance dashboard charts as favorites with the **Share as Public** option enabled before including this data in a rule. For details, see "Enable Performance Dashboard Data Forwarding on the OMi Server" on the next page.

For information on how to forward this data to BVD, see "Forward OMi Data to BVD" on the next page.

This section includes:

- "BVD Data Channels for OMi Data" below
- "Enable Performance Dashboard Data Forwarding on the OMi Server" on the next page
- "Forward OMi Data to BVD" on the next page

BVD Data Channels for OMi Data

When you send BVD data to OMi by using forwarding rules, BVD creates data channels that uniquely identify the data you are sending.

Each data channel consists of tags and dimensions (dims). Tags are static labels and dimensions are names that are associated with a specific value. For more information, see "Create custom integrations" on page 14.

The data channels are structured differently depending on the data you choose to forward:

Event Status

```
<tags connected server><tags forwarding rule><dim monitoring dashboard><dim
widget label><dim widget type>
```

KPI Status

```
<tags connected server><tags forwarding rule><dim view name><dim CI name><dim
KPI name>
```

Performance Dashboard

```
<tags connected server><tags forwarding
rule><metricName><instanceName><dSName><systemName><className>
```

`<tags connected server>` are all tags that are specified when adding a BVD Connected Server.
`<tags forwarding rule>` are all tags that are specified when creating a BVD Data Forwarding Rule.

Enable Performance Dashboard Data Forwarding on the OMi Server

To enable performance dashboard data forwarding to BVD, follow these steps:

1. In the Performance Perspective, select the charts that you want to forward and save them as favorites:

   a. In OMi, access **Workspaces > Operations Console > Performance Perspective**.

   b. In the **View Explorer**, select a view and then the CI for which you want enable data forwarding.

   c. In the **Performance** pane, choose a performance dashboard from the drop-down list.

   d. Click the title of the chart you want to save as favorite and click **Add to Favorite**.

   e. Choose to add the favorite to the default page, user-defined favorite page, or create a new user-defined favorite page. Click **Save**.

   f. Open the favorite in the **Performance** pane, access the menu and click **Save**. Check the **Share as Public** option and click **Save**.

Forward OMi Data to BVD

> **Important:** Before you create BVD data forwarding rules in OMi, access

> **Administration > Setup and Maintenance > Connected Servers** to add your BVD instance as a Connected Server. For details on which information to enter, see the *OMi Help*.

To forward OMi data to BVD, access **Administration > Setup and Maintenance > BVD Data Forwarding**.

1. In the right pane, click **Create**. Alternatively, click ✳ **New**.
2. In the **General** section, complete the following information:
   a. Enter a display name and (optional) a description for the forwarding rule.
   b. *Optional*. Enter a comma-separated list of tags.

      Data channels can be tagged to separate data from incoming streams and to create more specific data channels. For example, if you have separate OMi servers for different regions and you want separate dashboards for each region, you can add a tag that identifies the region in which this OMi server is located.

      The tags you enter will be added to the data channel after the tags specified for the BVD Connected Server. For details, see "BVD Data Channels for OMi Data" on page 5.
3. In the **Target BVD Server** section, choose one or multiple Connected Servers from the drop-down list. This list shows all Connected Servers of the type BVD and Alias. If your BVD connection is not listed in the drop-down, access **Administration > Setup and Maintenance > Connected Servers** to ensure that the server connection was set up correctly.
4. *Optional*. In the **Event Status** section, choose one or multiple monitoring dashboards from which you want to forward data to BVD.

   > **Caution:** If the monitoring dashboard name is changed, the data channel is **not** updated. Instead, a new data channel with the changed monitoring dashboard name is created. Widgets that use the old data channel in BVD will not receive data from OMi anymore and need to be updated to the new data channel.
5. *Optional*. In the **KPI Status** section, choose the one or multiple views from which you want to forward KPI status data. Click 🔧 next to the view name to choose specific KPIs. If no individual KPIs are selected, KPI status data will be forwarded for all CIs that are associated with the chosen view.
6. *Optional*. In the **Performance Dashboard Data** section, choose one or multiple public favorites of your performance dashboards for which you want to forward data to BVD. For information on how to add performance dashboards as favorites, see "Enable Performance Dashboard Data Forwarding on the OMi Server" on the previous page.

7. *Optional*. Clear the check box **Activate after save** if you want the status of the rule to be inactive after clicking **Save**. You can activate the rule at a later point in time.
8. Click **Save** to save the BVD data forwarding rule.

Integrate with OpsCx

To forward performance data from Operations Connector (OpsCx) to BVD, create a Data Forwarding policy in Operations Connector. The policy must specify your BVD instance as a target and contains rules that specify which data is sent to BVD.

Learn More

Support matrix

For a list of supported Operations Connector versions, see the support matrix:

Support Matrices for Operations Center products

Download and extract the support matrix files, open the document SUMA.htm and select **Operations Manager i Business Value Dashboard** from the product list.

Tasks

This section includes:

- "Configuring data forwarding in Operations Connector" below
- "Forwarding data using an HTTP proxy" on page 12
- "Forwarding data using HTTPS" on page 12

Configuring data forwarding in Operations Connector

1. In BVD, open 🔧 **Administration > Settings**, and copy the **API Key**.

   This key identifies your BVD instance and must be included in the data submitted by the data senders.

   > **Note:** The Settings page is only available to users with administrator privileges.

2. In Operations Connector, click **Create** in the toolbar. Then click **Forwarding > Data Forwarding**.
3. In the **Properties** page, define information that is related to the policy itself (for example, the name and description of the policy).
4. In the **Targets** page, set up your BVD instance as a target to which the data is forwarded:
   a. In the target list, click **New Item** above the **Name** column to add a new target.
   b. Enter the name of the BVD target and a description.
   c. Enter the URL of your BVD data receiver. The URL must include your API key and dimensions or tags (or both dimensions and tags) to create unique data channels for your metrics data. BVD expects to receive your data as HTTP post requests in JavaScript Object Notation (JSON) format.

      The URL should look something like this:

      `https://`*`<external_access_host>`*`:`*`<port>`*`/bvd-receiver/api/submit/`*`<API_key>`*`/dims/`*`<dims>`*

      `<external_access_host>`

      The fully qualified domain name of the host which you specified as `EXTERNAL_ACCESS_HOST` in the `install.properties` file during the Container Deployment Foundation installation. Usually, this is the master node's FQDN.

      `<port>`

      The port must be specified as part of the URL, even for the standard HTTP or HTTPS port. Default: 80 (HTTP) or 443 (HTTPS)

      > **Example**: If your target is `https://bvd1.example.net/bvd-receiver/api/submit/...`, add the standard HTTPS port to the URL: `https://bvd1.example.net:443/bvd-receiver/api/submit/...`

      `<API_key>`

      Identifies your BVD instance. You can find the API key in **Administration > Settings**.

      `<dims>`

      The names in your JSON name-value pairs. Select and combine dims that uniquely identify your data.

For more information on the URL of the BVD data receiver, see "Sending dimensions and tags in the receiver URL" on page 14.

> **Example**
>
> You can add the following metrics attributes as dimensions to the URL:
>
> ```
> dims/dataDomain,relatedCi,name,metricClass,node
> ```
>
> This results in the following data channel:
>
> ```
> HPBsmIntOEM oem.example.com/oracle_database/oemdb
> Average Active Sessions Throughput oem.example.com
> ```

For information on the metrics attributes, see the Operations Connector documentation.

d. Select **JSON** as **Wire format**.

e. Make sure **Use Guaranteed Delivery** is cleared. If you select this check box, Operations Connector attempts to resend data that could not be delivered earlier and BVD may receive outdated data.

5. Configure data forwarding rules:

   - To forward metrics data after policy rules are applied, select the **Metric** page and configure metric data forwarding rules.

     Metric forwarding rules define what a Data Forwarding policy should do in response to specific metric data. Each rule consists of a condition and of settings for the data generated by the policy. The settings enable you to configure what data Operations Connector forwards to which target.

   - To forward structured input data, that is, data gathered by a policy before the policy rules are applied, select the **Structured Input** page and configure structured input data forwarding rules.

     Structured input data rules define what a Data Forwarding policy should do in response to specific structured input data. Each rule consists of a condition and of settings for the data generated by the policy. The settings enable you to configure what data Operations Connector forwards to which target.

   Discarding rules have a higher precedence than forwarding rules.

6. Click **Save and Close** to save the policy and close the editor.

7. Download the root CA certificates from the BVD CA and import them to the Operations Connector trusted root certificate store. For details, see "Forwarding data using HTTPS" on the next page.

8. Activate the Data Forwarding policy in Operations Connector. Operations Connector starts sending data to BVD.

   In BVD, connect your widgets to the data channels opened up by Operations Connector.

Forwarding data using an HTTP proxy

If you must redirect the connection from Operations Connector to BVD through an HTTP proxy, configure the Operations Connector system as follows:

1. Edit the XPL configuration file:

   ```
   ovconfchg -edit
   ```

2. Add the following lines:

   ```
   [bbc.http]
   PROXY=<proxy_hostname>:<proxy_port>+(<included hosts>)-(<excluded hosts>)
   ```

   Replace *<included_hosts>* with a comma-separated list of hostnames or IP addresses to which the proxy enables communication. Replace *<excluded hosts>* with a comma-separated list of hostnames or IP addresses to which the proxy cannot connect. Asterisks (*) are wild cards in hostnames and IP addresses. Both *<included_hosts>* and *<excluded hosts>* are optional.

   To specify multiple proxies, separate each proxy with a semicolon (;). The first suitable proxy in the list takes precedence.

   > **Example**
   >
   > ```
   > [bbc.http]
   > PROXY=myproxy.example.com:8888+(mybvd.example.com)
   > ```

3. Restart the Operations Connector process that forwards the data:

   ```
   ovc -restart opcgeni
   ```

Forwarding data using HTTPS

By default, Operations Connector only trusts the OMi server as certificate authority (CA). For Operations Connector to trust the CA that issued the BVD certificate, you must download the root CA certificates from the BVD CA and import them to Operations Connector trusted root certificate store.

1. *Only for SaaS deployments*. Download the following certificates from https://certs.godaddy.com/repository/:

   - GoDaddy Class 2 Certification Authority Root Certificate
   - GoDaddy Class 2 Certification Authority Root Certificate - G2
   - GoDaddy Secure Server Certificate (Intermediate Certificate) - G2

2. *Only for container or onPrem deployments*. Obtain the root CA certificates from the root and any intermediate authorities that issued the BVD server certificate.

3. On the Operations Connector system, import the certificates to the trusted root certificate store:

   ```
   ovcert –importtrusted –file <certificate_file>
   ```

4. *Optional.* On the Operations Connector system, run the following command to verify that the trust has been established:

   ```
   bbcutil -ping https://<BVD_server>:<port>
   ```

   > **Example**
   >
   > ```
   > bbcutil -ping https://bvdserver.example.com:12225
   > ```

   The following output indicates that the HTTPS connection is successful.

   ```
   https://<BVD_server>:<port>:
   (bbc-288) status=eServiceError coreID= bbcV= appN= appV= conn=0 time=471 ms
   ```

   `eServiceError` is normal and indicates that there is no BBC service on the BVD system. If the output includes eSSLError (or similar), the HTTPS connection was not successful, possibly because the trust has not been established correctly.

Create custom integrations

BVD expects to receive your data as HTTP post requests in JavaScript Object Notation (JSON) format.

We recommend that your JSON input contains flat data, consisting of name-value pairs. If you must send nested data, BVD automatically flattens the data (see also "Nested JSON data" on page 17). You can also send JSON data in arrays. This enables you to send multiple data objects in a single web service call (see also "JSON data arrays" on page 17).

See also "Example: Sending JSON Data to BVD" on page 18.

Learn More

This section includes:

- "Sending dimensions and tags in the receiver URL" below
- "Sending dims and tags as HTTP parameters" on page 16
- "Sending dims and tags in the receiver URL and as HTTP parameters" on page 16
- "JSON data arrays" on page 17
- "Nested JSON data" on page 17
- "Data storage" on page 17

Sending dimensions and tags in the receiver URL

The BVD receiver URL should look something like this:

- URL with dimensions only:

  ```
  https://<external_access_host>/bvd-receiver/api/submit/<API key>/dims/<dims>
  [,<dims=value>]
  ```

- URL with tags only:

  ```
  https://<external_access_host>/bvd-receiver/api/submit/<API key>/tags/<tags>
  ```

- URL with both dimensions and tags:

  ```
  https://<external_access_host>/bvd-receiver/api/submit/<API key>/dims/<dims>
  [,<dims=value>]/tags/<tags>
  ```

  The tags can also precede the dims:

  ```
  https://<external_access_host>/bvd-receiver/api/submit/<API
  key>/tags/<tags>/dims/<dims>[,<dims=value>]
  ```

If the application sending the data is also installed as a suite container, define the receiver URL as follows:

```
http://bvd-receiver.<namespace>.svc.cluster.local:4000/bvd-
receiver/api/submit/<API_key>
```

`<external_access_host>`

> The fully qualified domain name of the host which you specified as `EXTERNAL_ACCESS_HOST` in the `install.properties` file during the Container Deployment Foundation installation. Usually, this is the master node's FQDN.

`<namespace>`

> The namespace assigned to your suite deployment. You can check the namespace by accessing **SUITE > Management** in the Management Portal.

`<API_key>`

> Identifies your BVD instance. You can find the API key in 🔧 **Administration > Settings**.

`<tags>`

> Static labels that you can attach to your data to create more specific data channels.

`<dims>`

> The names in your JSON name-value pairs. Select and combine dimensions (dims) that uniquely identify your data.

`<dims=value>`

> The names and values in your JSON name-value pairs. Directly assign values with names to improve data identification. Use this option, for example, if you have separate servers of the same data source for different locations and you want separate dashboards for each location. These name-value pairs do not have to be part of the JSON input. If they are, the values in the JSON input will be overwritten by the values in the URL.

Sending dims and tags as HTTP parameters

You can also submit the dims and tags as HTTP parameters of the URL.

> **Example**
>
> ```
> https://bvd.example.com/bvd-receiver/api/submit/47a648e9065d465012e541288b5a345e?dims=viewName,ciName,kpiName,location=nyc&tags=omi,kpi
> ```

Sending dims and tags in the receiver URL and as HTTP parameters

You can combine the receiver URL and HTTP parameters to send dims and tags. Define the dims and tags as part of the URL path first, then add additional dims and tags as HTTP parameters.

> **Example**
>
> ```
> https://bvd.example.com/bvd-receiver/api/submit/47a648e9065d465012e541288b5a345e/dims/viewName,ciName=abc/tags/omi,kpi?dims=kpiName,location=nyc&tags=bvd
> ```

However, if the same dimension or tag is specified more than once, the value of the last query parameter overwrites the values of the previous parameters. The value of the last query parameter appears multiple times as data channel.

> **Example**
>
> ```
> https://bvd.example.com/bvd-receiver/api/submit/47a648e9065d465012e541288b5a345e/dims/location=boston?dims=location=nyc&tags=east&dims=location=atlanta
> ```
>
> In this example, the dim `location` will have the value `atlanta`. Because dimensions are accumulated, the value `atlanta` appears three times as data channel.

JSON data arrays

You can submit multiple JSON objects in a single web service call by adding them to an array.

Array:

```
[
   {
      a: 1,
      b: 2
   },
   {
      c: 3,
      d: 4
   }
]
```

Nested JSON data

If the input contains nested data, BVD automatically flattens the data by renaming nested name-value pairs to include the names of the parent elements, separated by slashes (/), for example:

Nested JSON data:

```
{
   a: 1,
   b: 2,
   c: {
      x: 6,
      y: 7
   }
}
```

Flattened JSON data:

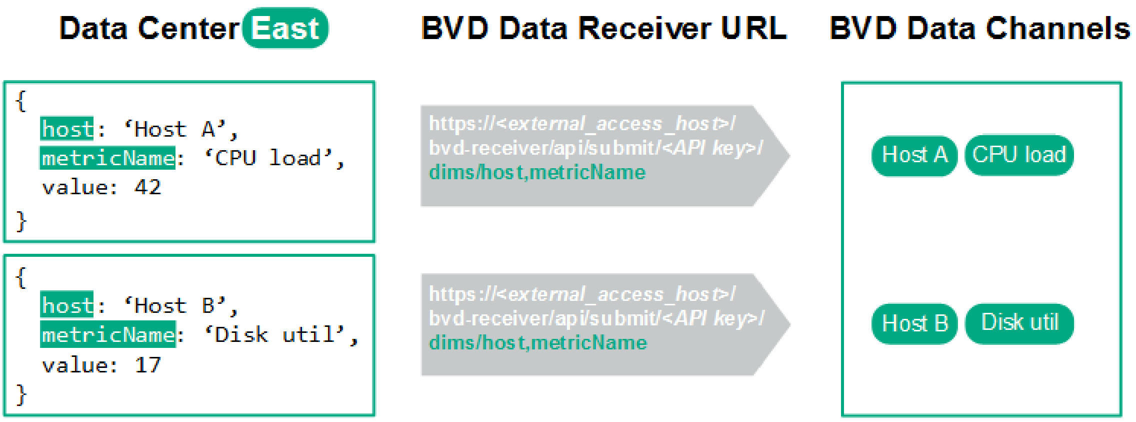
```
{
   a: 1,
   b: 2,
   c/x: 6,
   c/y: 7
}
```

Data storage

BVD stores only a specific number of data records per channel. The records are only kept if they are related to a widget. The BVD data aging process scans the database every 30 minutes to identify and automatically delete records that exceed the configured maximum age.

Example: Sending JSON Data to BVD

Sending Data from Data Center East



```
{
  host: ‘Host A’,
  metricName: ‘CPU load’,
  value: 42
}
```

```
{
  host: ‘Host B’,
  metricName: ‘Disk util’,
  value: 17
}
```

https://<external_access_host>/bvd-receiver/api/submit/<API key>/dims/host,metricName

https://<external_access_host>/bvd-receiver/api/submit/<API key>/dims/host,metricName

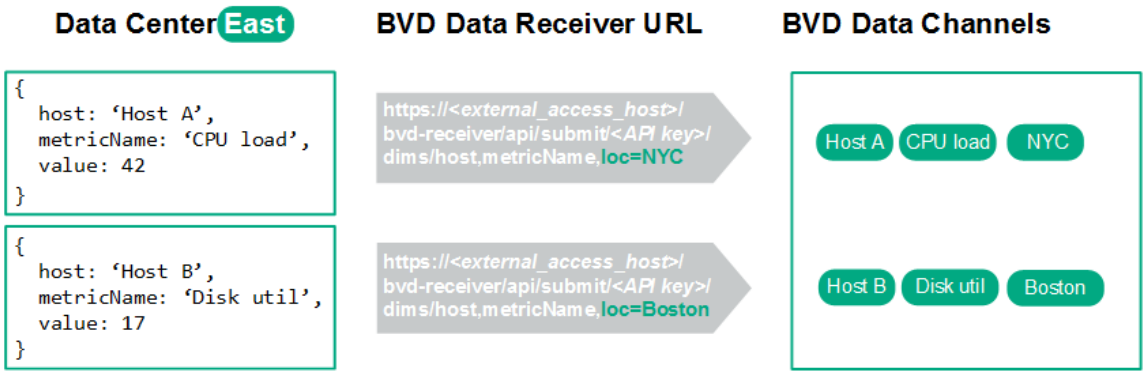
In this example, Data Center East sends two sets of JSON data to the BVD receiver. In both sets, the data fields `host` and `metricName` uniquely identify the value. The fields are therefore selected as dimensions (dims) and included in the URL. Once received by the BVD server, the JSON data creates two data channels:

`Host A` `CPU load` and `Host B` `Disk util`.

Lessons learned: Pick the fields in your data that uniquely identify the values you want to send to BVD and include the fields as dimensions in the HTTP post request.

> **Note:** If you send data to BVD from an application that is **not** part of the suite container deployment (for example a classically installed OMi), define the receiver URL as follows:
>
> `https://<external_access_host>/bvd-receiver/api/submit/<API key>`
>
> If you send data to BVD from an application that is also installed as a suite container, define the receiver URL as follows:
>
> `http://bvd-receiver.<namespace>.svc.cluster.local:4000/bvd-receiver/api/submit/<API_key>`
>
> *`<namespace>`* is the namespace assigned to your suite deployment. You can check the namespace by accessing **SUITE > Management** in the Management Portal.

Sending Additional Data From Data Center East



```
{
  host: 'Host A',
  metricName: 'CPU load',
  value: 42
}
```

```
{
  host: 'Host B',
  metricName: 'Disk util',
  value: 17
}
```

https://*<external_access_host>*/bvd-receiver/api/submit/*<API key>*/dims/host,metricName,loc=NYC

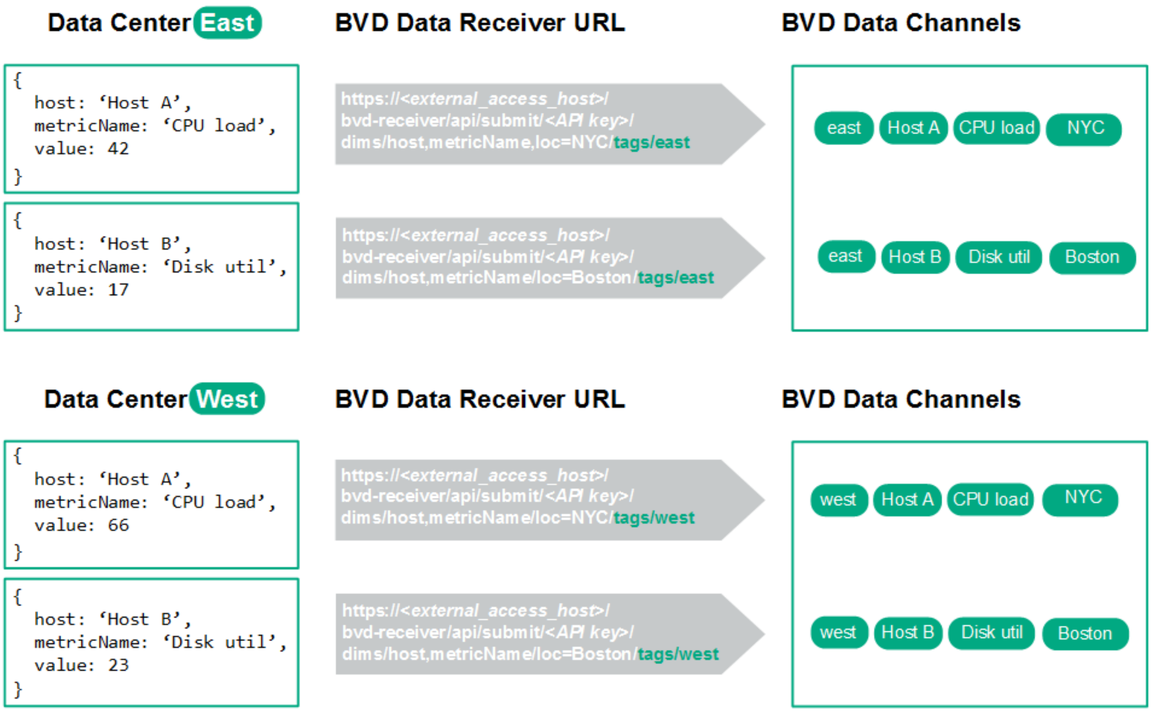
https://*<external_access_host>*/bvd-receiver/api/submit/*<API key>*/dims/host,metricName,loc=Boston

The primary location of Data Center East is in New York City, with backup servers located in Boston. Both locations send the same set of JSON data. To differentiate the data from the two locations without modifying the JSON data, you can add an additional dimension `loc` with the corresponding value to the URL. The modified URL updates the data channels to `Host A` `CPU load` `NYC` and `Host B` `Disk util` `Boston`.

In this example, we added the dimension `loc` to the URL.

Lessons learned: Directly assign values to your dimensions by adding dim=value pairs to the HTTP post request.

Sending Data From Data Center West



Data Center East
BVD Data Receiver URL
BVD Data Channels
https://<external_access_host>/ bvd-receiver/api/submit/<API key>/ dims/host,metricName,loc=NYC/tags/east
https://<external_access_host>/ bvd-receiver/api/submit/<API key>/ dims/host,metricName/loc=Boston/tags/east
east
Host A
CPU load
NYC
east
Host B
Disk util
Boston
Data Center West
BVD Data Receiver URL
BVD Data Channels
https://<external_access_host>/ bvd-receiver/api/submit/<API key>/ dims/host,metricName/loc=NYC/tags/west
https://<external_access_host>/ bvd-receiver/api/submit/<API key>/ dims/host,metricName/loc=Boston/tags/west
west
Host A
CPU load
NYC
west
Host B
Disk util
Boston

```
{
  host: 'Host A',
  metricName: 'CPU load',
  value: 42
}
```

```
{
  host: 'Host B',
  metricName: 'Disk util',
  value: 17
}
```

```
{
  host: 'Host A',
  metricName: 'CPU load',
  value: 66
}
```

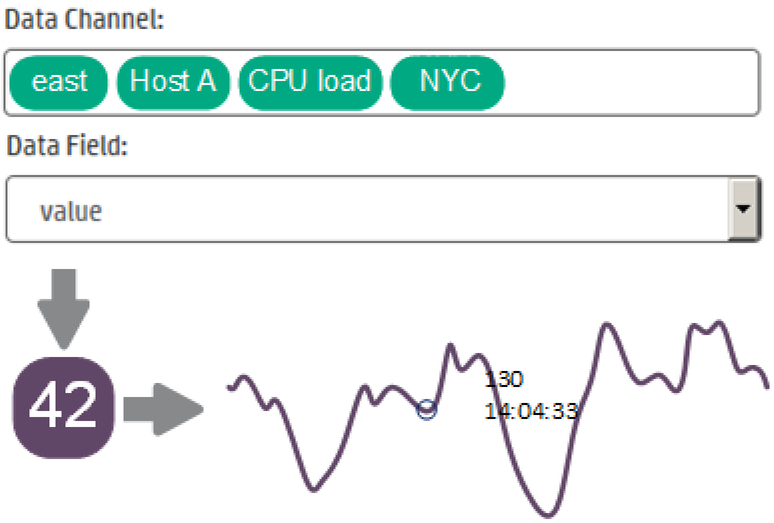
```
{
  host: 'Host B',
  metricName: 'Disk util',
  value: 23
}
```

A second data center, Data Center West, starts sending data similar to the JSON data sent by Data Center East. The data from Data Center West uses the same data channels as the data from East. To distinguish the data from the two centers, you must add the origin to the data. You can do this by adding tags to the URL. Tags are static labels that you can attach to your data to create more specific data channels.

In this example, we added the tags `east` and `west` to the URL. The tags precede the dims in the data channels.

Lessons learned: Attach tags to your data to create specific data channels.

Associating Data Channels with Widgets



Data Channel:

east Host A CPU load NYC

Data Field:

value

Once BVD has received the data, it creates the corresponding data channels. You can then associate a data channel with your widget in the widget's properties. In this example, the data channel `east` `Host A` `CPU load` NYC has been selected for the sparkline widget.

By default, the widget consumes data from the `value` data field. In this example, the current value is 42. If the field that holds the values you are interested in has a different name (for example, metricVal), select that name in the Data Field property of the widget.

Lessons learned: Connect your data to a widget by selecting the corresponding data channel in the widget's properties.

Send documentation feedback

If you have comments about this document, you can contact the documentation team by email. If an email client is configured on this system, click the link above and an email window opens with the following information in the subject line:

Feedback on Integration Guide (Business Value Dashboard 10.63)

Just add your feedback to the email and click send.

If no email client is available, copy the information above to a new message in a web mail client, and send your feedback to ovdoc-asm@hpe.com.

We appreciate your feedback!