

Distributed Services Architecture (DSA)

DSA brings together entities like devices, services, and applications into one data model that updates in real time. By representing these entities as one system, DSA makes possible or simplifies various tasks, such as analytics, inter-device communication, distributed computing, and application development.

You can read more about DSA on the [DSA home page](#).

Structure of DSA

DSA represents each of the entities in the system as one of the following types:

- **Broker**—A broker does a wide variety of management tasks. For example, a broker manages security, links, subscriptions, and node permissions. Other entities in the system can operate only as permitted by a broker. A broker also saves configuration data to disk. A broker also routes data—that is, it moves data from a source to a destination.
 - **DSLlink**—A DSLink connects to a broker. A DSLink creates, publishes, and interacts with data. A DSLink can also subscribe to data in the DSA system—that is, it can receive data whenever the data changes. You can find DSLinks in [IOT-DSA on Github](#).
 - **Node**—Some nodes are brokers, DSLinks, metrics, actions, or attributes. Other nodes, such as folders, do not fit into any of these categories.
 - **Metric**—A metric can exist on any broker, DSLink, or other organizational node. A metric is a key/value pair, in which the value can be any of the data types listed in Supported Data Types, including an arbitrary value map.
 - **Data Node**—Data nodes allow data to be stored on the broker's host server.
 - **Action**—An action can exist on any broker, DSLink, other node, or metric. An action is an invocable command that can affect an entity. For example, an action might create a node or set a metric value.
 - **Attribute**—An attribute can exist on any broker, DSLink, other node, or metric. An attribute is metadata for the selected entity, represented as a key/value pair.
-

Upstream and Downstream Connections

Upstream and *downstream* connections are important concepts for permissions configuration when you work with multiple servers or with multiple brokers on one server. A downstream entity requests permission, and an upstream entity either grants or refuses that permission. A broker is always upstream from its DSLinks. A broker can be either upstream or downstream from another broker.

Connecting Two DSA Installations (Brokers)

When you connect two DSA installations, one is an upstream broker and the other is a downstream broker.

For an example, see [this video](#).

To connect two DSA installations:

1. From the downstream broker, right-click **sys > upstream**, and choose **Add Upstream Connection**.
2. For **Name**, specify the node name that you want to appear in the downstream tree to represent the upstream broker.
3. For **Url**, specify the port, followed by /conn, for example <http://localhost:8081/conn>.
4. For **Broker Name**, specify the name that you want to appear in the upstream tree to represent the downstream broker.
5. For **Group**, specify the permission group for the upstream broker.
6. Log in to the upstream broker.
7. To authorize the downstream broker to remove it from quarantine, from the upstream broker, choose **sys > quarantine > Deauthorize**, and specify the downstream broker.

Supported Data Types

DSA supports these data types:

- **String**—A sequence of characters or an empty string.
- **Number**—A number or a null value.
- **Bool**—A true or false value.
- **Array**—An array object or a null value. The values in the array are of the dynamic data type. An example array of number

values is `[2,3,5,7,11]`. An example array of map values is `[{"hello":"world","number":1}, {"hello":"world"}]`.

- **Map**—A map object containing key/value pairs, or a null value. The key is always a string, and the value is the dynamic data type. An example value is `{"hello":"world","primes":[2,3,5,7,11]}`.
- **Binary**—A byte array expressed as a string, or a null value. The string begins with `\u001Bytes:` and ends with a byte array encoded in base 64.
- **Dynamic**—A value that can be any of the above types.

Installing DSA

To install DSA on Mac OS X, Microsoft Windows, Linux, or other platforms, follow the steps in the relevant video on [this page](#), depending on your platform.

If you are installing DSA on a Raspberry Pi, BeagleBone Black, or DGBox, you can also follow the steps in [this blog post and video](#).

If you install DSA with DGLux5, and do not have a DGLux5 license, follow the steps to request a license when you are prompted to do so.

Orientation to the DGLux5 Data and Metrics Panels with DSA

The following image shows the [Data panel](#) and [Metrics panel](#) in DGLux5 when DSA is used.

The screenshot shows a web application interface with a tree view on the left and a metrics section at the bottom. The tree view is under the 'Project Data' tab and includes nodes like 'data', 'downstream', 'System', 'Weather', 'sys', 'config', 'dglux', 'links', 'dataflow', 'JDBC', 'System', 'Weather', 'quarantine', 'tokens', 'upstream', 'users', and 'upstream'. The metrics section is titled 'Metrics' and contains a search bar and a list of system metrics: Architecture : x86_64, Battery Level : 88 %, CPU Usage : 28.56 %, Diagnostics Mode : disabled, and Disk Usage : 45.1 %. Five numbered callouts (1-5) are placed over the interface to indicate specific nodes or elements.

1	data node	2	downstream node	3	sys node
4	sys > links node	5	metrics at the currently selected node (downstream > System)		

Storing Data on the DSA Server

To store data on the DSA Server:

1. In the Data panel, right-click the data node.
2. Choose **Add > Node** or **Add > Value**.

The **Value** option lets you specify a data type.

3. Without moving the mouse pointer away from the pop-up menu, enter a node name, and optionally choose a data type.

A top-level data node is created.

4. In the Metrics panel, right-click the new data node that you created.
5. Choose **Add > Node** or **Add > Value** to create a child of this node.
6. Continue adding nodes until all nodes are represented.

To set a value:

1. In the Data panel, select the data node.
2. In the Metrics panel, right-click the metric whose value you want to set.
3. Choose **@set**, enter the value, and click **Invoke**.

To invoke an action on a metric:

1. In the Data panel, select the data node.
2. In the Metrics panel, right-click the metric on which you want to invoke an action.

A pop-up menu of available data actions is displayed.

3. Choose the action, enter any required information, and click **Invoke**.

Installing and Updating a DSLink

To install a DSLink:

1. In the Data panel, right-click the **sys > links** node.
2. Choose **Install Link**, and choose an installation method:
 - If you have a link to a ZIP file, choose **from Url**
 - If you have a ZIP file to upload, choose **from Zip**.
 - If the link can be found in [IOT-DSA on Github](#), choose **from Repository**.
3. Specify a name for the new DSLink.

4. Without moving the mouse pointer away from the pop-up window, click **Invoke**.

When the link has been successfully installed, a "Success" message appears.

5. To start the link, right-click the link under **sys > links > [link name]**, and choose **Start Link**.

When the link has started successfully, it appears under **downstream > links > [link name]**.

To update a DSLink:

1. In the Data panel, right-click the link under **sys > links**.
2. Choose **Update** and choose an update type, and then click **Invoke**.

Remote Dataflow

DSA installations feature a dataflow DSLink, also referred to as *remote dataflow*. This tool is similar to DGLux5 dataflow, described [here](#), also known as *local dataflow*.

Differences between the two are:

- Remote dataflow is stored on the DSA server with other DSLinks, not in the project files. Therefore, remote dataflow can run even when the project is not open in a browser.
- Certain blocks, such as Browser API blocks, are not available in remote dataflow.

To create a remote dataflow model:

1. In the Data panel, right-click **downstream > dataflow**.
2. Choose **Create Dataflow**.
3. Specify a name for the new dataflow model, and click **Invoke**.
4. Choose the new dataflow model at **downstream > dataflow > [dataflow model name]**.

Block property values in a remote dataflow model appear as metrics in the Metrics panel. They can be used like any other metric. The following image demonstrates an example of remote dataflow metrics.

The screenshot displays a dataflow editor interface. On the left, a project tree under 'Data' shows a folder 'example_dataflow' containing an 'add' block, which is highlighted with a red box. Below the tree, a search bar and a list of items are visible, with '0 : 2', '1 : 2', and 'output : 4' highlighted by a red box. A 'Metrics' section below shows various system parameters like '\$base', '\$dgid', '\$dgType', '\$is', '\$writable', '@length', '@ps', '@x', and '@y'. In the center, a 'Blocks' palette is open, showing categories like 'Variables', 'Data Services', and 'Browser API'. On the right, a 'Dataflow' canvas shows a diagram with two 'number' blocks (value: 2) connected to an 'add' block (input 0: 2, input 1: 2, output: 4), which is highlighted with a red box. A 'Text' area above the canvas contains the text 'Text' and a red circle.

Security and Permissions

See also: [Tokens](#).

Quarantined Brokers and Links

When *quarantine* is enabled on a broker, any downstream broker or link without an authorized token is held in quarantine. The system can read, subscribe to, and command nodes that are in quarantine, but a node that is in quarantine cannot access other nodes in the system.

Whether quarantine is enabled is specified in `dsa/dglux-server/server.json`.

To remove a node from quarantine, you can *authorize* or *deauthorize* the node. An authorized node is granted access as the permission group that you specify. A deauthorized node is refused access and removed from the system. You can also deauthorize any downstream broker or node that has previously been authorized.

When quarantine is disabled on a broker, any link can connect to the broker without approval.

To authorize a broker or link:

1. Right-click **sys > quarantine > Authorize**.

A list of quarantined nodes appears.

2. For **Dsid**, choose the node that you want to authorize.
3. For **Group**, choose the permission group that you want to assign.
4. Specify a name, and click **Invoke**.

To deauthorize a broker or link:

1. Right-click **sys > quarantine > Deauthorize**.

A list of quarantined nodes appears.

2. For **Name**, choose the node that you want to authorize.
3. Click **Invoke**.

Permission Groups

In DSA, you assign permission groups to users. Permission groups are defined within `dsa/dglux-server/server.json`. Information about permission groups can be found [here](#). Any user with "superuser" enabled has the maximum permission level and does not need a permission group defined.

More information about DSA permissions can be found [here](#).

Users

To create a new user:

1. Right-click **sys > users**, and choose **Create User**.
2. Without moving the mouse pointer away from the pop-up window, enter a user name and password, and specify whether the user has superuser access.
3. Click **Invoke**.

To edit a username:

1. Right-click **sys > users > [username]**, and choose **Rename User**.

To edit a user password:

1. Right-click **sys > users > [username]**, and choose **Change Password**.
2. Enter the new password, and click **Invoke**.

To edit a user's permission group, landing page, or other user properties:

1. Right-click **sys > users > [username]**, and choose **Edit User**.
2. Edit the properties, and click **Invoke**.

To delete a user:

- Right-click **sys > users > [username]**, and choose **Remove User**.

From:

<https://wiki.dglogik.com/> - DGLogik

Permanent link:

https://wiki.dglogik.com/dsa_wiki:home?rev=1475623018

Last update: **2021/09/20 14:19**

