
°C
Desigo

Desigo

- Desigo CC is an integrated building management platform for managing high-performing buildings. With its open design, it has been developed to create comfortable, safe, and efficient facilities. It is easily scalable from simple single-discipline systems to fully integrated buildings.
- It uses a RESTful API with a bearer token to provide security

Easier Imports

- The naming conventions used vary greatly depending geographic region, contractor, or even day of the week!
- To help with this, we have our own import solution
- Our Desigo Import Scripts that use incredibly powerful regular expressions come bundled with some of our Desigo packages!

Licensing

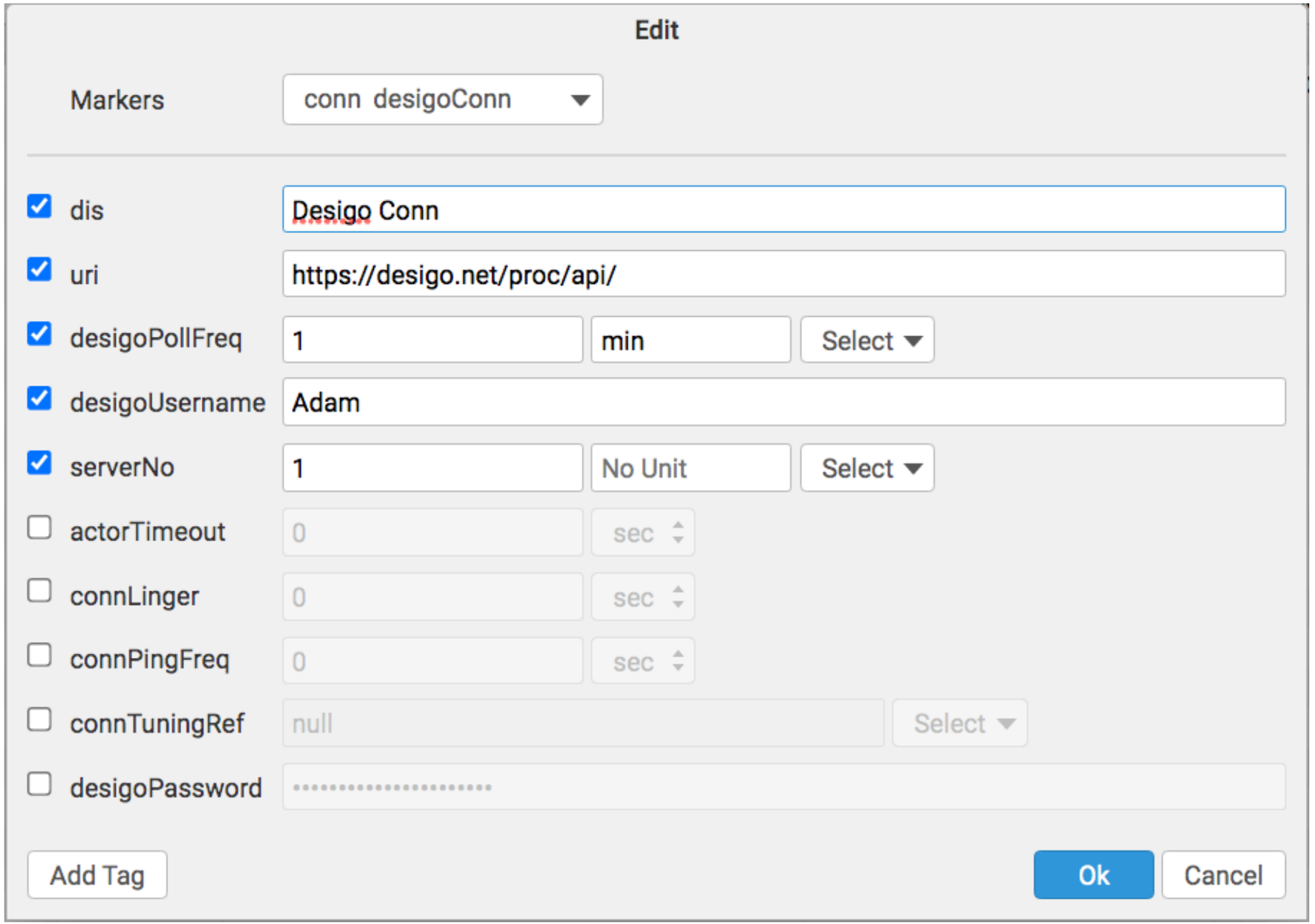
- Each Desigo connector extension is licensed to a single SkySpark Node Id
- Each Desigo connector extension is licensed to a set maximum number of points
- If either of these states goes into fault, the issue must be corrected and SkySpark must be restarted

# of Points on Single SkySpark Node	SkySpark List Price	Desigo Connector Price	Unlimited Desigo Points on One SkySpark Node	Desigo Import Scripts	Support
10	$ 60.00	Not Sold	N/A	N/A	N/A
100	$ 400.00	$ 125.00	No	No	$175/hr for help
1,000	$ 3,050.00	$ 1,000.00	No	No	$175/hr for help
5,000	$ 11,500.00	$ 3,800.00	No	Yes	2 hours of complementary support and then $175/hr for help
10,000	$ 15,500.00	$ 5,100.00	No	Yes	3 hours of complementary support and then $175/hr for help
50,000	$ 34,000.00	$ 11,300.00	Yes - Best Value!	Yes	6 hours of complementary support and then $175/hr for help

Desigo Connectors in SkySpark

- Creating a Desigo connector:
  - dis: Any Readable Name
  - uri: `http://host/xyz/api/` ie `https://desigo.net/proc/api/`
  - desigoUsername: This will be provided by your Desigo Team
  - desigoPassword: This will be provided by your Desigo Team
  - serverNo: Some buildings have multiple Desigo servers running. If you are not sure, start with: 1
  - The desigoPollFreq tag determines the minimum amount of time before the cache can be refreshed and how often the curVal values will be updated (which is from where both polling with intervals and COV come). The default value of 1min is good for detecting COV changes. The curVals do not matter for trend log synching.
  - Of course, these records will have the desigoConn and conn tags
- These can be created in the connector app

Desigo Connector

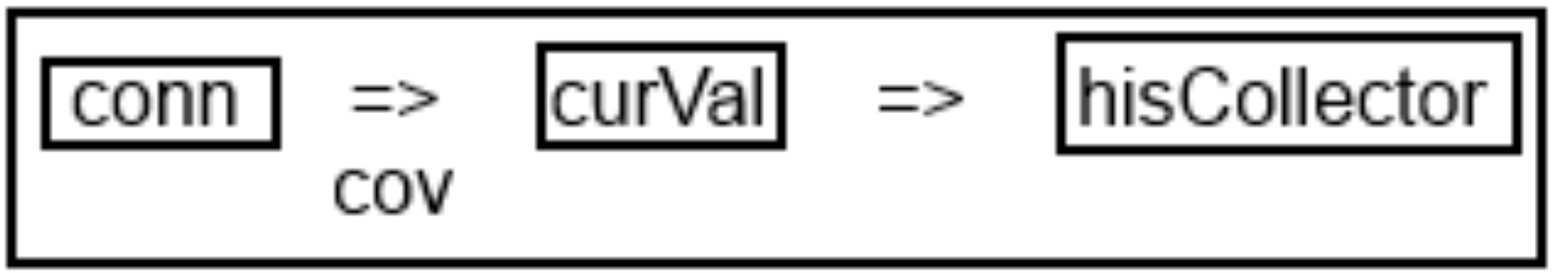


Desigo Connector Set Up

- The connector should authenticate by itself and stay authenticated automatically, but if you need to give it jumpstart or wake it up, you may run: read(desigoConn).desigoToken

Polling of curVal

- desigoPollFreq (duration)
  - How often you want SkySpark to poll the connector for COV
  - The default is 1min
  - It is a tuning parameter
  - Note that this determines how often the curVal in SkySpark changes. You would still need to set how often the data is collected with the hisCollect tags.



Desigo Functions

- desigoCmd – send Desigo a custom get request
  - read(desigoConn).desigoCmd
- desigoPost – send Desigo a custom post request
  - read(desigoConn).desigoPost
- desigoLearn – return a grid of all available cur and his points (walkable in Builder)
  - read(desigoConn).desigoLearn
- desigoRawLearn – return a grid of all available cur and his points
  - read(desigoConn).desigoRawLearn
- desigoSyncHis – sync history from a trend log
  - readAll(desigoConn).desigoSyncCur
- desigoSyncCur – force one or more points to sync
  - readAll(desigoConn).desigoSyncCur
- desigoPing – send ping to connector to check connection
  - read(desigoConn).desigoPing

Getting Data for Desigo Points

Manually or Automatically

- Collecting current values (from curVal):
  - Each point must have these tags:
    - desigoConnRef (ref) which is a ref that points to the Desigo Connector
    - desigoCur (str) which points to the Desigo point object
    - his (marker)
    - cur (marker)
    - hisCollectCov (marker or number) for cov collection OR
    - hisCollectInterval (duration) for polling
- Getting history data from Trend Logs:
  - Each point must have these tags:
    - desigoConnRef (ref) which is a ref that points to the Desigo Connector
    - desigoHis (str) which points to the Desigo Trend Log object
    - his (marker)
    - Run the desigoSyncHis(timeRange) function
      - generally as a task
      - readAll(desigoHis).desigoSyncHis(null)

Note: we recommend also using a hisCollectInterval of 24hr when using a hisCollectCov to catch faulty sensors.

convertTo Feature

- In rare cases, you may want to convert the data to something other than a point's datatype. This allows you to set up enums.
- You can do this today for desigoHis and desigoCur points
- The connector will do this before anything else
- The tag options are:
  - convertTo: "Str"
  - convertTo: "Number"
  - convertTo: "Bool"

Basic Troubleshooting

- Note that it is normal for a connection to close when it has no points in a watch for a given period of time
- For larger systems, you want to set the poll frequency to a higher value to avoid network congestion
- Make sure time zones, kinds, and units are the same in SkySpark and Desigo

Getting Points via Code

```
() => do
  dConn: read(desigoConn)
  dConn.desigoRawLearn().map r => do
    r = r.merge(
    {
      equipRef: read(equip)->id,
      siteRef: read(site)->id,
      hisCollectInterval: if (r->kind == "Number") 15min,
      hisCollectCov: if (r->kind != "Number") marker(),
    })
  end.each r => commit(diff(null, r, {add}))
```

Run as a task in large systems to avoid a timeout error

Getting Points with GUI

- We recommend using the Builder App's drag and drop capability to select which points are of interest as there are often many superfluous points