

Horizon DaaS Platform 15.3.2 REST API

This document provides an overview of the DaaS platform REST API.

July 2015

vmware[®]

Revision History

Date	Version	Description
7/20/2015	1.0	Initial release

© 2015 VMware, Inc. All rights reserved. This product is protected by U.S. and international copyright and intellectual property laws. VMware products are covered by one or more patents listed at <http://www.vmware.com/go/patents>.

VMware is a registered trademark or trademark of VMware, Inc. in the United States and/or other jurisdictions. All other marks and names mentioned herein may be trademarks of their respective companies.

VMware, Inc.
3401 Hillview Ave.
Palo Alto, CA 94304
www.vmware.com

Contents

1 About This Document	5
1.1 Intended Audience	5
1.2 Organization of this Document	5
1.3 Questions	5
2 Understanding the API	6
2.1 Resources	6
2.2 Scope	6
2.3 Properties and Links	6
2.4 Link Attributes	6
2.5 HTTP Requests	7
3 Getting Started	9
3.1 Retrieve the Available DaaS Versions	9
3.2 Login and Save the HTTP Response Header Authorization	9
3.3 Get the DtSecurityManager	10
3.4 Retrieve All Users for a Domain	10
3.5 Get the DtInfrastructureManager	11
3.6 Retrieve the Mapped Pools or Pattern for a Specific User	12
3.7 Example Error	14
4 Resources	15
4.1 DtVersion	15
4.2 DtInfrastructureManager	16
4.2.1 DtDataCenter	19
4.2.2 DtDesktopManagerNetwork	20
4.2.3 DtDesktopModel	20
4.2.3.1 DtLicenseEntitlement	21
4.2.4 DtHypervisorManager	21
4.2.5 DtComputePool	22
4.2.5.1 DtComputePoolProperty	24
4.2.6 DtNetwork	24
4.2.7 DtSessionQuota	25
4.2.8 DtStorageConfig	25
4.2.9 DtStorageSystem	26
4.2.10 DtTenantDesktopManager	26
4.3 DtInstallManager	27
4.3.1 DtAppliance	27
4.4 DtNotificationManager	28
4.4.1 DtMaintenanceNotice	28
4.4.2 DtNotification	29
4.4.3 DtNotificationFilter	29
4.4.4 DtVirtualMachineNotification	29
4.5 DtPoolManager	30
4.5.1 DtApplication	32
4.5.2 DtApplicationIcon	33
4.5.3 DtApplicationSessionPool	33
4.5.4 DtDesktopPool	35
4.5.4.1 DtPoolPolicy	36
4.5.5 DtDynamicDesktopPattern	37
4.5.6 DtGoldPattern	38
4.5.7 DtMapping	39
4.5.8 DtPatternFilter	39

4.5.9 DtPoolTask	40
4.5.10 DtRemoteApplication	40
4.5.11 DtSessionPool	41
4.5.12 DtStaticDesktopPattern	42
4.5.13 DtVirtualMachine	43
4.5.14 DtVirtualMachineFilter	44
4.6 DtQuotaManager	45
4.6.1 DtDesktopModelQuota	45
4.6.2 DtQuotaFilter	46
4.6.3 DtRemoteProtocolQuota	46
4.6.4 DtTemplateQuota	47
4.7 DtReportingManager	47
4.7.1 DtBillingReportFilter	48
4.7.2 DtConcurrentUsersFilter	48
4.7.3 DtConcurrentUsersReport	49
4.7.4 DtQuotaBillingReport	49
4.7.5 DtReportFilter	50
4.7.6 DtSuperTenantBillingReport	50
4.7.7 DtTenantReport	51
4.7.8 DtTenantReportKey	51
4.7.9 DtUserEventReport	51
4.7.10 DtUserEventReportFilter	52
4.8 DtSecurityManager	53
4.8.1 DtDomain	53
4.8.1.1 DtDomainAccount	55
4.8.1.2 DtGroup	55
4.8.1.3 DtRole	56
4.8.1.4 DtUser	56
4.8.2 DtOrganization	57
4.8.2.1 DtContact	58
4.9 DtSessionManager	58
4.9.1 DtActiveSession	59
4.9.2 DtSessionCountFilter	60
4.9.3 DtSessionFilter	60
4.10 DtSettingsManager	60
4.10.1 DtFeatureSetting	61
4.11 DtSystemManager	61
4.11.1 DtPlatform	62
4.11.2 DtSystemVersion	62
4.11.3 DtTermsOfService	63
4.12 DtTaskManager	63
4.12.1 DtTaskFilter	64
5 Enumerated Data Types	65
5.1 DtComputePoolType	65
5.2 DtDisplayProtocol	65
5.3 DtDomainAccountType	65
5.4 DtEndpointPlatformType	66
5.5 DtNetworkType	66
5.6 DtPatternType	66
5.7 DtPoolSizeType	67
5.8 DtQuotaBillingType	67
5.9 DtTaskStatus	67
5.10 DtUserEvent	67
5.11 DtUserEventError	68
5.12 DtVMLifeState	68
5.13 DtVMPowerState	69

1 About This Document

This document provides an overview of the DaaS platform REST API.

1.1 Intended Audience

The audience for this document is those who want to programmatically use DaaS platform APIs. This document assumes that you are familiar with:

- REST
- Software as a service

1.2 Organization of this Document

[About This Document](#) describes the content and intent of subsequent sections and introduces some terminology used in the remainder of the document.

[Understanding the API](#) introduces the architecture of the DaaS platform API and the concepts you need to understand to use the API, such as the links through which you traverse the object model to retrieve resources.

[Getting Started](#) presents examples that explain the basics of using the API. The examples show you how to login, save the HTTP Response Header authorization for use in subsequent HTTP requests, and perform several actions by traversing the object model to obtain the URIs for those actions.

[Resources](#) lists the resources, their links, and their attributes.

[Enumerated Data Types](#) lists the enumerated types that are the legal values for certain resource attributes.

1.3 Questions

If you need assistance or have questions about any of the information in this document, contact VMware Customer Support.

2 Understanding the API

The DaaS API provides access to DaaS platform features and functionality (resources) via REST-based web services. Exchanges between the caller and the platform take place using HTTP and HTTPS requests and the platform returns XML to the caller in the HTTP response.

2.1 Resources

Figure 1 shows the top-level entry points for the DaaS REST API.

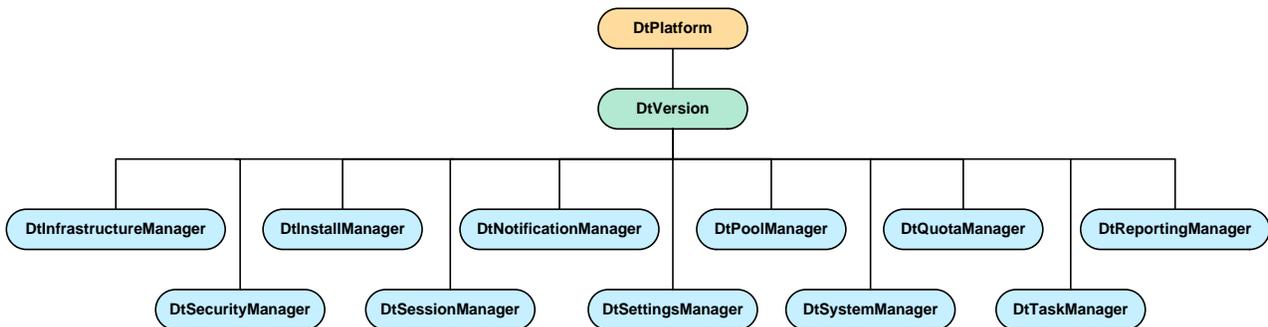


Figure 1: Top-Level Resource Tree

2.2 Scope

The DaaS platform REST APIs have a common object model across its management appliances – service provider or tenant. However, there are some resources that are available only to service providers and others that are only available to the tenants. Among the resources that are common to both service provider and tenants, there is a scope associated with individual links in the resource instances that restricts the availability of some of the links to service providers or tenants. So the same resource will contain different links based on the context of its retrieval.

2.3 Properties and Links

Resources contain properties and links:

- Properties store values. For example, the DtUser resource has a loginName property that indicates the name used by a user to login.
- A link specifies a URI of a related resource or action. For example, the DtUser resource has a desktopPatterns link that retrieves the desktop patterns assigned to a user.

2.4 Link Attributes

A link has the following attributes:

- **name** – The unique name of the link. The name typically describes the purpose or target of the attribute.

- **href** – The URI relative to the server, webapp and version, for example `/infrastructure/pool/1000`
- **method** – The http method used with the href in the REST API invocation: GET, POST, PUT, or DELETE.
- **rel** – Describes the relationship between the link and the resource that contains the link. The rel attribute is for informational purposes and is not required when coding to the API. The table below defines the link relationships.

Table 2–1 Link Relationships

Relationship	Description
ACTION	Indicates that the URI performs an action with the resource
ASSOCIATION	Indicates that the URI retrieves an associated resource
MAP	Indicates that the URI retrieves a MAP between resources (one-one, one-many, many-many)
SELF	Indicates that the URI is the URI of the container resource. A REST response always contains a link to itself. A GET on this URI returns the same URI
TOP	Indicates that the URI is a top-level root resource

2.5 HTTP Requests

The HTTP request you use to retrieve a DaaS resource is constructed as follows:

`https://<host>/dt-rest/<version>/<URI>`

- *host* is the IP address of the Enterprise Center or Service Center.
- *version* is the version of the REST API (See Section 3.1 on page 5 for information on retrieving available versions).
- *URI* is the path to the resource, constructed by traversing the object model and extracting path information from the href attribute of a link.

For example:

`/infrastructure/manager/mapping/users/default?sep={sep}&users={users}&`

Note the following features of URI syntax:

- The first component of the path is always the functional area of the DaaS platform: system, infrastructure, or security.
- The `/manager` component of the path is used only for top-level resources (infrastructure, system, and security) and links served by that resource.
- Curly braces are used to represent request parameter values which you must supply in the actual HTTP request. The href attribute uses the following syntax to represent request parameters:

`?p1={p1}&p2={p2}`

So for the preceding example, the actual values for the two request parameters (sep and users) might look as follows (where sep specifies the character you are using to delimit the user ids in the request):

`?sep=, &users=1234, 5678&`

- URL encoding is used to replace characters outside the ASCII set with a percent sign character (%) followed by two hexadecimal digits. For example, the question mark character (?) is replaced with %3F and the ampersand character (&) is replaced with %26.

Here is the complete HTTP request:

```
https://Host/dt-  
rest/v100/infrastructure/manager/mapping/users/default?sep=,&users=1234,5678&
```

3 Getting Started

This chapter gives you the basics you need to get started using the API, showing you how to:

- Retrieve the available DaaS versions via the published URL `/system/platform`
- Login and save the HTTP Response Header authorization.
- Get the Security Manager.
- Retrieve a domain and all users for that domain.
- Get the Infrastructure Manager.
- Retrieve the mapped pools or pattern for a specific user.

Before using the samples in this chapter to begin coding to the API, you will need the Tenant or Service Provider IP address and version 6.0.0 or greater of the DaaS platform installed.

3.1 Retrieve the Available DaaS Versions

To retrieve the available DaaS versions, submit the following HTTP request, where Host is the IP address of the Enterprise Center or Service Center.

```
https://Host/dt-rest/system/platform
```

The response includes a single `DtPlatform` XML element which contains one `DtVersion` element for each available DaaS version. The latest version is the `DtVersion` element for which the attribute `latest="true"`. For example:

```
<DtVersion id="v100" latest="true">
  <DtLink href="/system" method="GET"
    name="DtSystemManager" rel="top"></DtLink>
  <DtLink href="/infrastructure/manager" method="GET"
    name="DtInfrastructureManager" rel="top"></DtLink>
  <DtLink href="/security/manager" method="GET"
    name="DtSecurityManager" rel="top"></DtLink>
  <loginURI>/v100/system/login</loginURI>
</DtVersion>
```

The `DtVersion` element also defines the three top-level links you use to traverse the DaaS REST API: `DtSystemManager`, `DtInfrastructureManager`, and `DtSecurityManager`.

3.2 Login and Save the HTTP Response Header Authorization

The API provides secure access via domain authentication. The `loginURI` property of the `DtVersion` element specifies URI you need to append to the base URL to submit an HTTP POST to login:

```
<loginURI>/v100/system/login</loginURI>
```

For example, if the login URI is `/v100/system/login`, then you could submit the following HTTP POST.

```
https://Host/dt-rest/v100/system/login
```

where the POST parameters are the user, pw, and domain.

The HTTP response header you receive when you login contains key-value pairs, one of which is named Authorization. The table below lists some of values you receive in the HTTP response header.

Table 3–1 Sample Hardware Requirements for DaaS Management Host

Key	Value
Server	Apache-Coyote/1.1
Authorization	Basic REVWXGVudGFkbWluOkRlc2t0b25lMQ==
Transfer-Encoding	chunked
Content-Encoding	gzip
Vary	Accept-Encoding
Date	Wed, 07 Dec 2011 14:53:05 GMT

You need to save the value for Authorization when you receive the HTTP response to your login request. Then, in every subsequent HTTP request you submit after you login, include this Authorization key-value pair in the HTTP request header.

3.3 Get the DtSecurityManager

To retrieve the DtSecurityManager, use the path information specified by the DtSecurityManager link of the DtVersion element:

```
<DtLink href="/security/manager" method="GET" name="DtSecurityManager" rel="top">
```

The href attribute indicates the URI to append to the base URI. Using this information, you could construct the following HTTP request:

```
https://Host/dt-rest/v100/security/manager
```

The response contains the following XML:

```
<List type="DtLink">
  <DtLink href="/security/manager" method="GET" name="DtSecurityManager"
rel="self"></DtLink>
  <DtLink href="/security/manager/domains" method="GET"
name="Domains" rel="association"> </DtLink>
  <DtLink href="/security/manager/organizations" method="GET"
name="Organizations" rel="association"> </DtLink>
</List>
```

This XML specifies three links you can traverse:

- The first link is a link to the DtSecurityManager itself.
- The second link indicates the URI to append to the base URL to retrieve all the domains.
- The third link indicates the URI to append to the base URL to retrieve all the organizations.

3.4 Retrieve All Users for a Domain

To retrieve all domains, use the path information specified by the Domains link of the DtSecurityManager element:

```
<DtLink href="/security/manager/domains" method="GET" name="Domains"
rel="association"></DtLink>
```

The href attribute indicates the URI to append to the base URI. Using this information, you could construct the following HTTP request:

```
https://Host/dt-rest/v100/security/manager/domains
```

The response contains one DtDomain element for each available domain, and each domain element contains four links:

- The first link is a link to the DtDomain itself.
- The second link indicates the URI to append to the save a domain.
- The third link indicates the URI to append to the base URL to retrieve all the groups in the domain.
- The fourth link indicates the URI to append to the base URL to retrieve all the users in the domain.

For example:

```
<DtLink href="/security/domain/DEV" method="GET" name="DtDomain"
rel="self"></DtLink>
<DtLink href="/security/domain/DEV/save" method="POST" name="Save"
rel="action"></DtLink>
<DtLink href="/security/domain/DEV/groups" method="GET" name="Groups"
rel="association"></DtLink>
<DtLink href="/security/domain/DEV/users" method="GET" name="Users"
rel="association"></DtLink>
```

To retrieve all users for the DEV domain, use the path information specified by the Users link of the DtDomain element:

```
<DtLink href="/security/domain/DEV/users" method="GET" name="Users"
rel="association"></DtLink>
```

The href attribute indicates the URI to append to the base URI. Using this information, you could construct the following HTTP request:

```
https://Host/dt-rest/v100/security/domain/DEV/users
```

The response contains one XML DtUser element for each user. Each DtUser element specifies the links you can traverse to obtain properties for the User. The following XML fragment shows two of the returned links:

```
<DtLink href="/security/domain/DEV/user/563d81c8d85a6f46a6d7599ad7b6d1c5/pools"
method="GET" name="DesktopPools" rel="association"></DtLink>
<DtLink href="/security/domain/DEV/user/563d81c8d85a6f46a6d7599ad7b6d1c5/patterns"
method="GET" name="DesktopPatterns" rel="association"></DtLink>
```

The user ID in this example is 563d81c8d85a6f46a6d7599ad7b6d1c5. To obtain information for this user, save the href attribute for use in subsequent calls (you can also save the self DtLink).

3.5 Get the DtInfrastructureManager

To retrieve the DtInfrastructureManager, use the path information specified by the DtSecurityManager link of the DtVersion element:

```
<DtLink href="/infrastructure/manager" method="GET"
name="DtInfrastructureManager" rel="top"></DtLink>
```

The href attribute indicates the URI to append to the base URI. Using this information, you could construct the following HTTP request:

```
https://Host/dt-rest/v100/infrastructure/manager
```

The response specifies the links you can traverse to obtain properties such as patterns, pools, desktop models, and quotas. The following XML fragment shows two of the returned links:

```
<DtLink href="/infrastructure/manager/dcs" method="GET"
  name="DataCenters" rel="association"></DtLink>
<DtLink href="/infrastructure/manager/models" method="GET"
  name="DesktopModels" rel="association"></DtLink>
```

3.6 Retrieve the Mapped Pools or Pattern for a Specific User

To retrieve the mapped pools or pattern for a list of users, use the path information specified by the DefaultMappingOfUsers link of the DtInfrastructureManager:

```
<DtLink href="/infrastructure/manager/mapping/users/default?sep={sep}&users={users}&"
  method="GET" name="DefaultMappingOfUsers" rel="map"></DtLink>
```

The href attribute indicates the URI to append to the base URI. Using this information, you could construct the following HTTP request:

```
https://Host/dt-
rest/v100/infrastructure/manager/mapping/users/default?sep=,&users=563d81c8d85a6f46a6d7599ad7b6d1c5,
599ad7b6d1c5,bcce77ea74cdd148b9c2c08d84dfff2a&
```

The response contains one XML Entry element for each user. Each Entry element specifies the default pool or pattern for the user. For example:

```
<Map>
  <Entry>
    <key xmlns:xsi=http://www.w3.org/2001/XMLSchema-instance
  xmlns:xs=http://www.w3.org/2001/XMLSchema
  xsi:type="xs:string">563d81c8d85a6f46a6d7599ad7b6d1c5</key>
    <DtDesktopPool id="1078">
      <DtLink href="/infrastructure/pool/1078" method="GET"
        name="DtDesktopPool" rel="self"></DtLink>
      <DtLink href="/infrastructure/pool/1078/delete" method="DELETE"
        name="Delete" rel="action"></DtLink>
      <DtLink href="/infrastructure/pool/1078/update" method="PUT"
        name="Update" rel="action"></DtLink>
      <DtLink href="/infrastructure/pool/1078/patterns" method="GET"
        name="Patterns" rel="association"></DtLink>
      <DtLink href="/infrastructure/pool/1078/vms" method="GET"
        name="VirtualMachines" rel="association"></DtLink>
      <DtLink
        href="/infrastructure/pool/1078/assign/domain/%7BdId%7D/group/%7BgId%7D"
        method="PUT" name="AssignGroup" rel="action"></DtLink>
      <actualSize>0</actualSize>
      <dateCreated>2011-09-02T17:23:57.473Z</dateCreated>
      <DtDesktopModel id="28428506-fce0-43c3-a4be-13043c79b3f5">
        <DtLink href="/infrastructure/model/28428506-fce0-43c3-a4be-13043c79b3f5"
          method="GET" name="DtDesktopModel" rel="self"></DtLink>
        <defaultProtocol>RDP</defaultProtocol>
        <desktopMemoryInMBs>2048</desktopMemoryInMBs>
        <diskSpaceInGBs>0</diskSpaceInGBs>
        <enabled>true</enabled>
        <name>DebStaticDesktopModel</name>
        <numberOfCpus>2</numberOfCpus>
        <supportedProtocols>ICA RDP VNC</supportedProtocols>
        <type>static</type>
      </DtDesktopModel>
      <domainName>DEV</domainName>
      <DtGoldPattern id="G.1001.62">
        <DtLink href="/infrastructure/pattern/gold/G.1001.62"
          method="GET" name="DtGoldPattern" rel="self"></DtLink>
        <DtLink href="/infrastructure/pattern/gold/G.1001.62/update"
          method="PUT" name="Update" rel="action"></DtLink>
```

```

<DtLink href="/infrastructure/pattern/gold/G.1001.62/convert/desktop"
  method="POST" name="ConvertToDesktop" rel="action"></DtLink>
<DtLink href="/infrastructure/pattern/gold/G.1001.62/pools"
  method="GET" name="DesktopPools" rel="association"></DtLink>
<DtLink href="/infrastructure/pattern/gold/G.1001.62/rename?name=%7N%7D&"
  method="PUT" name="Rename" rel="action"></DtLink>
<DtLink href="/infrastructure/pattern/gold/G.1001.62/vm"
  method="GET" name="VirtualMachine" rel="association"></DtLink>
<DtLink href="/infrastructure/pattern/gold/G.1001.62/dcs"
  method="GET" name="DataCenters" rel="association"></DtLink>
<deleted>>false</deleted>
<name>Deb_Gold_Pattern</name>
<companyName>Deskton</companyName>
<enabled>>true</enabled>
<key></key>
<modified>>false</modified>
<notes></notes>
<osType>Windows7_64</osType>
<timeZoneId>EST</timeZoneId>
<username>Administrator</username>
<vmUuid>4222f9e2-0552-ade8-623b-6f77200599ab</vmUuid>
</DtGoldPattern>
<highlyAvailable>>false</highlyAvailable>
<lastUpdated>2011-11-14T20:25:39.937Z</lastUpdated>
<name>DebUserPool</name>
<DtOrganization id="1001">
  <DtLink href="/security/organization/1001" method="GET"
    name="DtOrganization" rel="self"></DtLink>
  <DtLink href="/security/organization/1001/quotas" method="GET"
    name="DesktopModelQuotas" rel="association"></DtLink>
  <dateCreated>2011-09-01T20:20:50.216Z</dateCreated>
  <dateUpdated>2011-09-01T20:20:50.216Z</dateUpdated>
  <disabled>>false</disabled>
  <name>Tenant1001</name>
</DtOrganization>
<patternType>S</patternType>
<poolModeMessage></poolModeMessage>
<poolOnline>>true</poolOnline>
<DtPoolPolicy id="2140429207">
  <allocatorSessionTimeout>360000</allocatorSessionTimeout>
  <allowFullDesktop>>false</allowFullDesktop>
  <allowedToJoinDomain>>true</allowedToJoinDomain>
  <assignedGroupNames></assignedGroupNames>
  <clipboardRedirect>>true</clipboardRedirect>
  <comPortRedirect>>false</comPortRedirect>
  <desktopDeallocAction></desktopDeallocAction>
  <drivesRedirect>>true</drivesRedirect>
  <maxPoweredOnVMs>50</maxPoweredOnVMs>
  <minPoweredOnVMs>10</minPoweredOnVMs>
  <nonPersistent>>false</nonPersistent>
  <organizationalUnit></organizationalUnit>
  <printerRedirect>>true</printerRedirect>
  <smartCardRedirect>>false</smartCardRedirect>
</DtPoolPolicy>
<poolSizeType>Fixed</poolSizeType>
<requestedSize>3</requestedSize>
<supportedPoolProtocols>RDP</supportedPoolProtocols>
<vmRootName>DebUserPool</vmRootName>
</DtDesktopPool>
</Entry>
</Map>

```

3.7 Example Error

The table below shows the key-value pairs returned in the HTTP response to an incorrectly formulated HTTP request to retrieve non-existent pool information.

Table 3–2 Sample Error Code in HTTP Response

Key	Value
x-dt-error-num	1
x-dt-error-code	DT_RESOURCE_NOT_FOUND
X-dt-error-msg	Cannot find desktop pool with id nonexistent

When you receive an HTTP error, the XML returned contains a DtErrorInfo element. For example:

```
<DtErrorInfo>
  <errorCode>DT_RESOURCE_NOT_FOUND</errorCode>
  <errorMessage>Cannot find desktop model with id nonexistent</errorMessage>
  <errorTime>2011-09-29T14:23:07.031-04:00</errorTime>
</DtErrorInfo>
```

4 Resources

4.1 DtVersion

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DtInfrastructureManager	A top level entry point to traverse the DaaS resources	GET	Top	Both
DtInstallManager	A top level entry point to traverse resources related to the installation of DaaS appliances	GET	Top	SP
DtPoolManager	A top level entry point to traverse desktop or session pool related resources	GET	Top	Tenant
DtReportingManager	A top level entry point to retrieve reports	GET	Top	Both
DtSecurityManager	A top level entry point to traverse the DaaS security object model	GET	Top	Both
DtSystemManager	Represents the DaaS Platform and provides various miscellaneous functionality and information about the current installation	GET	Top	Both

Properties

Name	Description	Data Type
domainRegistrationURI	URI used by clients to register a domain to the local appliance. This URI can only be used to register the first domain for the organization. Additional domains can be registered using DtSecurityManager's RegisterDomain link	String
loginURI	The URI used by clients to login to the DaaS Platform with this version	String

4.2 DtlInfrastructureManager

The top level entry point to traverse the DaaS resources.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
ApplianceTemplate	Assigns a virtual machine as an appliance template Return value - no return value (void)	POST	Action	SP
AssignHostManagerToElement	Assign the specified hypervisor manager to a tenant desktop manager (Service Provider only)	POST	Action	SP
AssignNetworkToDesktopManager	Assign the specified network to the desktop manager	PUT	Action	SP
AssignStorageConfigToElement	Create a storage mount point (datastore) based on the given storage config on all hosts used by a given tenant desktop manager (Service Provider only)	POST	Action	SP
CreateDesktopModel	Creates a new desktop model that you will be able to modify later	POST	Action	SP
CreateOrUpdateDesktopManagerDMQuota	Sets or updates a collection of quotas for a specified desktop model to the requested amount	POST	Action	SP
CreateOrUpdateTemplateQuota	Creates a new template quota or updates an existing template quota (Service Provider only)	POST	Action	SP
CreateOrUpdateRemoteProtocolQuota	Creates a new remote protocol quota or updates an existing remote protocol quota (Service Provider only)	POST	Action	SP
CreateOrUpdateDesktopModelQuota	Creates new or updates a collection of quotas for desktop models for a tenant organization and datacenter to the requested amounts (Service Provider only)	POST	Action	SP
CreateOrUpdateSessionQuota	Creates a new session quota or updates an existing session quota for	POST	Action	SP

	a given tenant organization and datacenter (Service Provider only)				
CreateStorageConfig	Creates a new storage mount point on a discovered storage system (Service Provider only)	POST	Action	SP	
DataCenters	Retrieves a collection of all data centers in the platform	GET	Association	Both	
DesktopModels	Retrieves a collection of all available desktop models	GET	Association	Both	
DesktopModelQuotas	Retrieves a collection of all desktop model quotas	GET	Association	Both	
DiscoverHypervisorManager	Discover a particular hypervisor manager (Service Provider only)	POST	Action	SP	
DiscoverHypervisorManagerByAddress	Discover a hypervisor manager for the given resource manager by its DNS address Return Value: DtHypervisorManager	POST	Action	SP	
DiscoverStorageSystem	Discover a storage system (Service Provider only)	POST	Action	SP	
HypervisorManagers	Retrieves all the hypervisor managers currently discovered in this environment. (Service Provider only)	GET	Association	SP	
RemoveStorageConfig	Removes a storage mount point configuration in the environment (Service Provider only)	POST	Action	SP	
RemoveStorageSystem	Removes a storage system from the environment (Service Provider only)	POST	Action	SP	
RetrieveTemplateQuotaForTenant	Retrieves the template quotas for all datacenters in the tenant organization (Service Provider only)	GET	Action	SP	
RetrieveAllRemoteProtocolQuotas	Retrieves the remote protocol quotas for all datacenters in the tenant organization (Service Provider only)	GET	Action	SP	
RetrieveAllStorageConfigs	Retrieves all storage mount point	GET	Action	SP	

	configurations (Service Provider only)				
RetrieveAllStorageSystems	Retrieves all discovered storage systems (Service Provider only)	GET	Action	SP	
RetrieveSessionQuotaForTenant	Retrieves the session quota for all datacenters in the tenant organization (Service Provider only)	GET	Action	SP	
RetrieveStorageConfigsForStorageSystem	Retrieve all storage configs belonging to the specified storage system (Service Provider only)	GET	Action	SP	
TenantDesktopManagers	Retrieves the collection of tenant desktop managers in the current organization	GET	Association	Both	
TenantDesktopManagersByDataCenter	Retrieves the collection of DtTenantDesktopManager objects filtered for a specified data center ID.	GET	Association	Tenant	
TenantNetworks	Retrieves all networks assigned to the given tenant organization. Return Value - Collection of DtNetwork resources	GET	Association	SP	
DtInfrastructureManager	Link to this DtInfrastructureManager	GET	Self	Both	
UnassignHostManagerFromElement	Remove the assignment of the specified hypervisor manager from the given tenant desktop manager (Service Provider only)	POST	Action	SP	
UnassignNetworkFromDesktopManager	Unassign the specified network to the desktop manager	PUT	Action	SP	
UnassignStorageConfigFromElement	Remove the specified datastore config from each host of the given tenant desktop manager (Service Provider only)	POST	Action	SP	

Properties

NA

4.2.1 DtDataCenter

A logical grouping of infrastructure elements, containing information about the server, network, and storage components that are logically grouped to aid administration and provisioning of services.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DtDataCenter	Link to this resource	GET	Self	Both

Properties

Name	Description	Data Type
assignedOrganizationIds	The organizations assigned to this data center	Long
backBoneNetworkId	The backbone network ID that is used for internal communications and management of the data center appliances	String
backBoneNetworkType	The network type of the backbone network (VLAN ID or network name)	String
creationDate	The date of creation of the data center	String
description	Any description associated with this data center or its purpose	String
dnsServer	The IP address of the DNS server of this data center	String
friendlyName	The human readable friendly name	String
frontNetworkId	The front-end network ID of this data center	String
frontNetworkType	The network type of the front-end network (VLAN ID or network name)	String
gateway	The gateway IP address of this datacenter	String
ipAddressBlock	The IP address block served by this data center	String
lastUpdate	The last update time on this data center	String
mcastAddress	The multicast IP address for this data center	String
mcastClusterName	The multicast cluster name for this data center	Integer
mcastPort	The multicast port of this data center	String
Name	The name of the data center	String
ntpServers	The list of NTP (Network Time Protocol) servers assigned to this data center	String
subnetMask	The subnet mask of this data center	String
vmgrId	The virtualization manager for this data center	String
vmgrUIDs	The list of virtualization manager UUIDs of this datacenter	String

4.2.2 DtDesktopManagerNetwork

A Network associated with a Desktop Manager.

Scope: Tenant

Properties

Name	Description	Data Type
DesktopManagerId	Desktop Manager associated with this Network Mapping	String
NetworkId	Network ID of mapping associated with Desktop Manager	String
NetworkName	Network Name of mapping associated with Desktop Manager	String

4.2.3 DtDesktopModel

Provides information about a virtual desktop's hardware and software configuration. A desktop model is assigned to a desktop pool and is used by the pool when it creates virtual desktops.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DtDesktopModel	Link to this resource	GET	Self	Both
Save	Persist the properties of this desktop model	POST	Action	Both

Properties

Name	Description	Data Type
currentDesktopCount	The current count of desktops using this desktop model	Long
dateCreated	The date the organization was created	String
defaultProtocol	The default remote access protocol used by desktops using this desktop model, for example RDP	Integer
desktopMemoryInMBs	The RAM size for the desktops using this desktop model	Integer
desktopQuota	The desktop quota of this desktop model	Long
diskSpaceInGBs	The disk space for desktops using this desktop model	Long
enabled	Specifies if this desktop model is enabled or disabled	Boolean
lastUpdated	The date the organization was last updated	String
licenseEntitlements	The license entitlement of this desktop model	Collection of LicenseEntitlement
name	The name of the desktop model, for example Normal	String
numberOfCpus	The number of CPUs in this desktop model	Integer
refId	The reference ID for the given desktop model	String
sessionBased	Assigns the session based attribute of the desktop model.	Boolean

supportedProtocols	The list of protocols supported by this desktop model: ICA, RDP, RGX, VNC, NX	Collection of DisplayProtocol
type	The type of the desktop model, for example selectable	String

4.2.3.1 DtLicenseEntitlement

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
dateCreated	The date and time the license entitlement was created	String
deleted	Specifies if this license entitlement is deleted	Boolean
desktopModelId	The unique ID of the desktop model to which this license entitlement is applied	String
entitlementContentType	The content-type of the entitlement file	String
entitlementFileName	The file name of the license entitlement	String
fileLocation	The entitlement file location	String
lastUpdated	The date the desktop model quota was last updated	String
organizationId	The ID of the organization to which the entitlement is applied	String
tenantUpdate	Specifies if this is a tenant update	Boolean
version	The version of the license entitlement	Float

4.2.4 DtHypervisorManager

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
AssignToElement	Assign this hypervisor manager to a tenant desktop manager	POST	Action	SP
AssignToTenantDesktopManager	Assigns a tenant desktop manager to this hypervisor manager so that compute pools from the hypervisor manager can be assigned to the specified desktop manager. Return Value: DtTenantDesktopManager	POST	Action	SP
Delete	A link to delete this hypervisor manager	POST	Action	SP
DtHypervisorManager	A link to this resource	GET	Self	SP

Elements	Retrieves all tenant desktop managers using this hypervisor manager	GET	Association	SP
ComputePools	Retrieves all compute pools managed by this hypervisor manager	GET	Action	SP
UnassignFromElement	Removes assignment of this hypervisor manager to a given tenant desktop manager	POST	Action	SP

Properties

Name	Description	Data Type
address	The DNS name or IP address of this hypervisor manager	String
state	The current assignment state of this hypervisor manager (Discovered or Assigned)	String
status	Current status of the hypervisor (for example, Online)	String
type	The virtualization technology (for example, vCenter)	String
url	The URL for API invocations	String
vendorId	The unique identifier of the vendor	String

4.2.5 DtComputePool

This interface represents a pool of compute resources that serves virtual machines as appliances or desktops.

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
DtComputePool	A link to this DtComputePool	GET	Self	SP
HypervisorManager	Retrieves the hypervisor manager that manages this compute pool	GET	Association	SP
TenantDesktopManager	Retrieves the tenant desktop manager (element) assigned to use this compute pool, NULL for SP	GET	Association	SP
Update	Persists updates to assign or unassign tenant desktop manager or organization to a compute pool. Return Value: no return value (void)	PUT	Action	SP
UpdateOverutilizationRatios	Updates the memory and CPU overallocation ratios using the parameters <code>memoryOverallocationRate</code> and <code>cpuRatioVirtualToPhysical</code> in the link's URL	POST	Action	SP

Properties

Name	Description	Data Type
computeRefId	Unique ID that is used by the IaaS layer to uniquely reference the logical/physical instance this compute pool uses. It could be the UUID of an ESX hypervisor host, the unique cluster name or the org VDC name in vCloud	String
cpu	Total number of CPUs offered by the underlying resource of this compute pool	Integer
cpuOverallocation	The ratio of the virtual CPUs (that will be served by this hypervisor in its VMs) to its actual number of physical CPUs	Double
cpuPartition	The number of CPUs offered by this compute pool. If the underlying resource of the compute pool is shared amongst multiple organizations then this would be a subset of the total CPUs of the underlying resource	Integer
dcId	The datacenter in which this compute pool will serve appliances or desktops	String
displayName	The public display name for this compute pool	String
memoryOverallocation	The ratio of the virtual RAM (that will be served by this hypervisor in its VMs) to its actual physical memory	Double
memoryPartitionInMBs	The amount of memory in MBs offered by this compute pool. If the underlying resource of the compute pool is shared amongst multiple organizations then this would be a subset of the total CPUs of the underlying resource	Long
organizationId	Unique identifier of the organization of this compute pool	Long
partitioned	Specifies whether this hypervisor host is partitioned or not	Boolean
properties	A collection of DtComputePoolProperties that represent the variable properties of this compute pool based on the type	DtComputePoolProperty
tenantDesktopManagerId	The unique ID of the tenant desktop manager that is using this compute pool, NONE if unassigned	String
type	The type of the hypervisor host, for example esx	String

4.2.5.1 DtComputePoolProperty

Scope: Service Provider

Links

NA

Properties

Name	Description	Data Type
computePoolId	Unique ID that is used by the IaaS layer to uniquely reference the logical/physical instance this compute pool uses. It could be the UUID of an ESX hypervisor host, the unique cluster name or the org VDC name in vCloud.	String
name	Name of the compute pool property e.g. status	String
role	The role that this host is in (active, standby, reserved)	String
status	The high level status as reported from the AP	String
swDescription	The long description of the software running on this host	String
swVersion	The version of the software running on this host	String
totalCpuCount	The total number of physical CPUs on this host	Integer
value	Property value for this name	String

4.2.6 DtNetwork

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
DtNetwork	Specifies if this is a link-local or a backbone network	GET	Self	SP

Properties

Name	Description	Data Type
datacenterId	Datacenter where this network operates	String
dateCreated	Date network was created	Date
dateUpdated	Date network was last updated	Date
defaultVlan	Specifies if this network is the default network of a tenant organization	Boolean
dnsServer	Address of the DNS server in this network	String
gateway	Address of the gateway for this network	String
networkId	Unique ID of this network (VLAN for a network type; VLAN and name for a SDN)	String
networkLabel	User-defined network label	String
networkType	DtNetworkType of this network	DtNetworkType

orgId	Organization of this network	Long
privateBackbonePort	Private backbone port if applicable	Long
subnetMask	Subnet mask of this network	String

4.2.7 DtSessionQuota

Session quota assigned to a tenant cluster (organization + datacenter).

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
DCId	Unique ID of the datacenter of this session quota	String
dateCreated	Creation date this session quota	Date
dateUpdated	Last update of this session quota	Date
orgId	Unique ID of the organization of this session quota	Long
quota	The maximum number of sessions allowed for this tenant cluster	Long

4.2.8 DtStorageConfig

A storage mount point configuration upon a discovered storage system.

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
DtStorageConfig	A link to this resource	GET	Self	SP

Properties

Name	Description	Data Type
localMountPoint	Path to local mount point of the storage configuration on the host	String
name	Name of this mount point on the host	String
ownerId	The unique ID of the owner of this storage configuration (e.g. organization ID)	Long
remoteMountPoint	Path to remote mount point of the storage configuration on the storage system	String
storageSystemId	The unique ID of the storage system where this storage configuration resides	String
tenantDesktopManagerIds	Unique ids of all tenant desktop managers using this storage configuration to create end-user desktops	Collection of String

4.2.9 DtStorageSystem

A storage system discovered and assigned to resource managers. Storage mount points on this storage system persist virtual machines (management appliances or desktops).

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
DtStorageSystem	A link to this storage system instance	GET	Self	SP
StorageConfigs	Retrieves all storage mount points configured for this storage system instance for the given resource manager	GET	Association	SP

Properties

Name	Description	Data Type
address	IP or DNS address of the storage system	String
type	Type of the storage system (Local, NFS, Desktone, Netapp, Nexenta, Isilon, WSS)	String
vendorId	Vendor ID that uniquely identifies the vendor of this storage system	String

4.2.10 DtTenantDesktopManager

An element in the infrastructure that manages tenant desktops.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
GetNetworks	Retrieves the networks assigned to this desktop manager.	GET	Action	Both

Properties

Name	Description	Data Type
dataCenterId	The datacenter that contains the tenant desktop manager	String
dateCreated	The date of creation of the tenant desktop manager	String
getVmgrId	The resource manager ID that this desktop manager is associated to	String
lastMonitorTime	The last time this tenant desktop manager was contacted by the monitoring host	String
lastUpdated	The last update time on this tenant desktop manager	String
monitoringHost	The unique ID of the host currently monitoring this tenant desktop manager	String
name	The friendly name of this tenant desktop manager	String
organizationId	The organization ID of this tenant desktop manager	Long

state	The current operational state of this tenant desktop manager	String
-------	--	--------

4.3 DtInstallManager

A top level entry point to manage all management appliances and associated resources and actions.

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
AllAppliances	Retrieves a collection of all appliances managed by the service provider organization	GET	Association	SP
DtInstallManager	A link to this resource	GET	Self	SP

Properties

NA

4.3.1 DtAppliance

A management appliance.

Scope: Service Provider

Links

Name	Description	Method	Relationship	Scope
DtAppliance	A link to this resource	GET	Self	SP
Restore	Restores the management appliance by deleting it and recreating and reinstalling it. (CAUTION - all snapshots of the appliance are lost if restore is performed)	POST	Action	SP
ResourceManagerId	Retrieves the unique ID of the resource manager that this appliance uses to interface with computing resources.	GET	Action	SP

Properties

Name	Description	Data Type
capabilities	Bit-wise value representing all capabilities of this management appliance (possible values are SP_FABRIC = 0, TRANSIT_SERVER=1, VMGR=2, APPLIANCE_TEMPLATE=3, Tenant_FABRIC=4, EMGR=5, PRIMARY_NODE=6, PRIMARY_NODE_ACROSS_DCS=7)	Long
computePoolId	Unique Id of the compute pool that hosts this management appliance	String
datacenterId	Unique Id of the datacenter where this management appliance resides	String

displayName	Management appliance's human readable display name	String
lastMonitorTime	Time when the management appliance was last monitored	Date
lifeState	Life state of this management appliance (Online, Offline, Unknown)	String
monitoringHost	Unique ID of the monitoring host of this management appliance	String
name	Management appliance's name	String
numMissedHeartbeats	Number of missed heartbeats from this management appliance	Integer
organizationId	Unique Id of the organization that this management appliance serves	String
state	State of this appliance (Reserved, Created, Installed, Disabled, Deleted)	String
version	Version of this management appliance	String
virtualMachineId	Unique ID of the virtual machine in the hypervisor host that operates this management appliance	String

4.4 DtNotificationManager

A top level entry point to handle notifications.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
CreateNotificationFilter	Creates a notification filter	GET	Association	Tenant
DtNotificationManager	Link to this resource	GET	Self	Tenant
Notifications	Retrieves all notifications that match the specified filter	GET	Association	Tenant

Properties

NA

4.4.1 DtMaintenanceNotice

Representation of a maintenance notice.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
type	Type of notification	String

dateCreated	Date the maintenance notice was created	Date
dateUpdated	Date the maintenance notice was last updated	Date
expirationDate	Date the notice expires	Date
message	Message of the maintenance notice	String
startDate	Date the notice becomes active	Date

4.4.2 DtNotification

Generic representation of a notification.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
type	Type of notification	DtNotificationType

4.4.3 DtNotificationFilter

Representation of a notification filter.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
active	Indicates whether to retrieve only active notification	Boolean
type	Type of notifications to retrieve	DtNotificationType

4.4.4 DtVirtualMachineNotification

Representation of a virtual machine notification.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
VMs	Retrieves the VMs affected by the notification	GET	Association	Tenant

Properties

Name	Description	Data Type
count	Number of VMs affected by the notification	Int

4.5 DtPoolManager

A top level entry point to manage desktop and session pools of tenants and associated resources and actions.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
AssignUserToStaticDesktopPattern	<p>Assigns a static desktop to the given user from the given static pool. Repeated invocations for the same input will return the user's existing static desktop assignment. If invoked on a pool where there are no more static desktops available, will return DT_RESOURCE_ASSIGNMENT_FAILED. The pool must be a static desktop pool, otherwise will return DT_INVALID_PATTERN_TYPE.</p> <p>Note this method may be invoked by a non-admin user, provided that the logged in user is the same as the user loginName specified. In other words, it can only be invoked by a non-admin user to perform the assignment to the same user.</p> <p>The user must be already be entitled to the specified pool.</p> <p>Parameters: poolId - the ID of the static pool to assign the desktop pattern from loginName - the loginName of the user to assign the desktop to</p> <p>Returns: DtStaticDesktopPattern assigned to the user</p>	PUT	Action	Tenant
ConvertToGoldPattern	Converts a static desktop pattern to a gold pattern.	POST	Action	Tenant
CreateDesktopPool	Creates a new pool of desktops or sessions (Tenant only)	POST	Action	Tenant
CreateApplicationSessionPool	Persists the given instance of an application session based pool as a new object into the database.	POST	Action	Tenant
CreatePatternFilter	Creates an empty pattern filter.	GET	Action	Tenant

CreateSessionPool	Persists the given instance of a session based pool as a new object into the database.	POST	Action	Tenant
CreateVirtualMachineFilter	Creates an empty virtual machine filter.	GET	Action	Tenant
DefaultMappingOfUsers	Retrieves the default mappings of a list of users	GET	map	Tenant
DesktopPoolOfUsers	Retrieves the desktop pools assigned to this user	GET	map	Tenant
DesktopPools	Retrieves a collection of all available desktop pools	GET	Association	Tenant
DesktopPoolsOfGroups	Retrieves the pool mappings of a list of groups	GET	map	Tenant
DtPoolManager	Link to this resource	GET	Self	Tenant
FindApplications	Retrieves all the applications.	GET	Association	Tenant
FindDesktopPatterns	Finds desktop patterns which matches filter criteria.	POST	Association	Tenant
FindVirtualMachines	Finds virtual machines which matches filter criteria.	POST	Association	Tenant
Patterns	Retrieves all patterns that belong to the current organization of the specified type. Specify one of the following types of patterns: <ul style="list-style-type: none"> • G - A pattern based on a template of the hypervisor. • D - A pattern that describes a dynamic virtual desktop machine. • S - A pattern that describes a static virtual desktop machine • U User - A pattern that describes a non-VM desktop machine. • A - A pattern that describes a desktop or application served by an app server that describes a non-VM desktop machine. 	GET	Association	Tenant
PatternsOfUsers	Retrieves the pattern mappings of a list of users.	GET	map	Tenant
Pool	Retrieves the pool specified by the poolId.	GET	Association	Tenant
Pools	Retrieves a collection of all available desktop pools.	GET	Association	Tenant
PurgeRecyclePool	Purges the recycle pool. Static patterns are moved to the recycle pool when a user does not want the virtual desktop any more. The purge recycle pool action goes ahead and deletes virtual desktops in the recycle pool from the hypervisor.	POST	Action	Tenant
RDP	Gets an RDP connection to the VM associated with the pattern.	GET	Action	Tenant

RefreshDynamicPool	Starts a Dynamic Pool Refresh task for the given gold pattern and dynamic pool	POST	Action	Tenant
ReserveDesktopPattern	Reserves a static pattern so it can then become eligible for gold pattern conversion. (Tenant only)	POST	Action	Tenant
UnreservePattern	Migrates the specified static pattern back to the pool it came from and creates a new static pattern	POST	Action	Tenant
VirtualMachine	Retrieves a virtual machine by virtual machine ID and pattern ID	GET	Association	Tenant
VirtualMachines	Retrieves a collection of all available virtual machines	GET	Association	Tenant

Properties

NA

4.5.1 DtApplication

Application information to provide applications sessions to users.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DtApplication	Link to this resource	GET	Self	Tenant
update	Persists the properties of this application	PUT	Action	Tenant

Properties

Name	Description	Data Type
applicationFilePath	Application file path of the application	String
applicationName	Application name of the application	String
commandline	Command line arguments of the application	String
dateCreated	Date the application was created	Date
dateUpdated	Date the application was last updated	Date
displayName	Display name of the application	String
hidden	Whether the application is hidden or not	Boolean
icons	Icons for the application	Collection
numberOfPoolsUsing	Number of pools using the application	Int
pattern	Gold pattern of the application	DtGoldPattern
patternVersionId	Pattern version ID for the application	Long
publisher	Publisher of the application	String
version	Version of the application	String

4.5.2 DtApplicationIcon

Data object representing an application icon.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DtApplicationIcon	Link to this resource	GET	Self	Tenant
IconData	Retrieves the icon data for the application icon.	GET	Action	Tenant

Properties

Name	Description	Data Type
height	Height of the application icon	Int
md5Sum	MD5 sum for the application icon	String
width	Width of the application icon	Int

4.5.3 DtApplicationSessionPool

A pool of RDS servers which only provide applications to users - not desktops.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
Applications	Retrieves all remote applications configuration for this pool	GET	Association	Tenant
AssignGroup	Assigns the specified user group to this desktop pool	PUT	Action	Tenant
AssignUserToPool	Assigns a specific user to a virtual desktop pool	PUT	Action	Tenant
Delete	Deletes this desktop pool and any relationships this pool had with other resources	DELETE	Action	Tenant
DtApplicationSessionPool	Link to this resource	GET	Self	Tenant
GroupsOfPool	Retrieves the groups currently assigned to the pool	GET	Association	Tenant
RemoteApplications	[Deprecated] DtRemoteApplication will no longer be the application listing utility. getApplications should be used instead.	GET	Association	Tenant
RemoveGroup	Removes the specified user group from this pool	PUT	Action	Tenant
RemoveUserFromPool	Removes a specific user from a pool	PUT	Action	Tenant

Update	Updates this pool and any new relationships this pool has with other resources	PUT	Action	Tenant
UpdateServerDensity	Updates the existing Session pool with a new Server Density	POST	Action	TENANT
UpdateWithApplications	Updates this pool and any new application relationships this pool	PUT	Action	TENANT
UsersAssignedByAdmin	Retrieves the users administrator assigned to pool regardless of whether they have a desktop mappings or not. Users that are not directly assigned to this pool, via a group mapping, will not be returned.	GET	Association	TENANT
UsersOfPool	Retrieves the users currently assigned to the pool. Users that are not directly assigned to this pool, via a group or only a desktop mapping, will not be returned.	GET	Association	TENANT
VirtualMachines	Retrieves the virtual machines in this desktop pool	GET	Association	TENANT

Properties

Name	Description	Data Type
actualSize	Actual size of the pool (i.e. the actual number of provisioned virtual desktops that are available)	Long
applicationIds	IDs of the applications associated with this pool	String
dateCreated	Date the pool was created	Date
defaultPoolProtocol	Default remote display protocol for this pool	DtDisplayProtocol
domainName	Security domain name for the desktops in this pool	String
highlyAvailable	Specifies if the pool has high availability	Boolean
lastUpdated	Date the pool was last updated	Date
name	Pool name	String
patternType	Pattern type of this pool (static or dynamic)	DtPatternType
poolModeMessage	Reason that the pool is online or offline	String
poolOnline	Specifies if the pool is online or offline	Boolean
poolSessionType	Pool session type (e.g. desktop, application, both)	DtPoolSessionType
poolSizeType	Pool size type (e.g. Fixed, Elastic)	DtPoolSizeType
serverCount	Number of servers to be provisioned for this pool	Long
serverDensity	Number of users that can be supported by RDS Server in the pool	Long
sessionBased	Indicates whether the pool is session-based	Boolean

supportedPoolProtocols	Remote access protocols that can be used to access virtual desktops from this pool. This list must be a subset (or the same) as the supported protocols in the pool's DtDesktopModel.	List<DtDisplayProtocol>
usedByDataCenterIds	IDs of data centers where this pool can have desktop instances	Collection
vmRootName	Base name for all virtual machines provisioned from this pool	String

4.5.4 DtDesktopPool

A pool of virtual desktops or sessions. Every pool has a gold pattern that serves as the base image to create desktops in the pool. A pool can be static or dynamic - a static pool has a static pattern for each virtual desktop that typically resides on a persistent disk so that user activity is preserved over multiple sessions. A dynamic pool has a single dynamic pattern that is applied to every virtual desktop in the pool typically in a non-persistent disk so every user session resets the virtual machine to its original image.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DtDesktopPool	A link to this resource	GET	Self	Tenant
Patterns	Retrieves the patterns in this desktop pools	GET	Association	Tenant
RemoteApplications	Retrieves all remote applications configured for this pool	GET	Association	Tenant
RemoveGroupAndAutoAssignedPatternMappings	For static desktop pools only, removes the specified user group from the desktop pool, and also removes any static desktop pattern mappings for users who were automatically assigned to desktops via initial connection to the pool	PUT	Action	Tenant
Update	Updates this desktop pool and any new relationships this pool has with other resources	PUT	Action	Tenant
UserDesktopMappings	Retrieves User Desktop Mappings for Static Desktop Pools	GET	Association	Tenant

Properties

Name	Description	Data Type
requestedSize	The requested size of the pool	Long
sessionBased	Specifies if the pool is a desktop or a session pool. True or false	Boolean
usedByDataCenterIds	A list of data center IDs where this pool can have desktop instances	String
virtualMachines	The Virtual Machines in this desktop pool	Collection of DtVirtualMachine

4.5.4.1 DtPoolPolicy

Configuration and initial setup options for desktop pools. Each desktop pool has one pool policy and is used for provisioning and pool management operations.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
allocatorSessionTimeout	The session timeout for allocators in the pool using this policy	Long
allowedToJoinDomain	Specifies the desktop pool is allowed to join a security domain	Boolean
allowFullDesktop	Connecting to the full desktop is allowed from clients (in addition to remote applications) for virtual desktops in the pool using this policy	Boolean
assignedGroups	The domain groups assigned to the pool with this policy	Collection
assignedGroupNames	The domain group names assigned to the pool with this policy	Collection of String
assignedNetworks	Tenant networks (network ids) assigned to this pool. All desktops in the pool will be provisioned to these networks.	Collection of String
comPortRedirect	Specifies remote access settings have enabled mapping client's local com ports to the virtual desktop.	Boolean
drivesRedirect	Specifies remote access settings have enabled mapping client's local drives to the virtual desktop.	Boolean
licenseKey	The license key for the desktop pool that uses this policy.	String
maxPoweredOnVMs	The maximum number of virtual machines that can be PoweredOn state in the pool that has this policy.	Integer
minPoweredOnVMs	The minimum number of virtual machines that must be PoweredOn in the pool that has this policy.	Integer
nonPersistent	Specifies the virtual desktops in the pool using this policy use non-persistent disks.	Boolean

organizationalUnit	The AD organization unit of the pool using this policy.	String
printerRedirect	Specifies remote access settings have enabled mapping client's printers to the virtual desktop.	Boolean
registeredTo	Used to populate the registered owner and registered organization properties of a windows desktop in that pool.	String
smartCardRedirect	Specifies remote access settings have enabled mapping client's smart cards to the virtual desktop.	Boolean
timezone	The time zone of the pool that uses this policy.	String
video3DEnabled	Indicates whether a pool is 3D video enabled or not.	Boolean

4.5.5 DtDynamicDesktopPattern

A virtual desktop that does not have specific user assignments. Any authorized user can use the virtual desktops from a pool of desktops.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DataCenters	Retrieves the data centers of this pattern	GET	Association	Tenant
DtDynamicDesktopPattern	A link to this resource	GET	Self	Tenant
Rename	Assigns a name to the pattern as would be known in the hypervisor (e.g. the VM name or the template name)	PUT	Action	Tenant
VirtualMachine	Retrieves the virtual machine associated with this pattern	GET	Association	Tenant

Properties

Name	Description	Data Type
dataCenterName	The name of the data center where this pattern is available	String
dateCreated	The date the pattern was created	String
deleted	Specifies if this pattern is deleted	Boolean
dependantPoolIds	A collection of pool ids where the pattern is being currently used	Collection of String
desktopPoolId	The ID of the desktop pool of this desktop pattern	String
lastUpdated	The last update time and date	String
name	The name of the pattern as known by the hypervisor (e.g. the VM name or the template name)	String
numberOfInstances	The number of instances of this desktop pattern at this location	Integer
previousDesktopPoolId	The pool ID of this pattern prior to migration to another pool	String
templateId	The templateId of the pattern	String

4.5.6 DtGoldPattern

A reserved virtual machine that can be cloned to create virtual desktops. A DtGoldPattern is created by reserving a virtual desktop. DtGoldPatterns are assigned to pools that create virtual desktops from them depending upon the desktop pool properties and desktop model

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
Applications	Retrieves the desktop pools using this gold pattern	GET	Association	Tenant
DataCenters	Retrieves the data centers of this pattern	GET	Association	Tenant
Delete	Deletes the pattern and its associated virtual machine	POST	Action	Tenant
DtGoldPattern	A link to this resource	GET	Self	Tenant
Pools	Retrieves the desktop pools using this gold pattern	GET	Association	Tenant
Rename	Assigns a name to the pattern as would be known in the hypervisor (e.g. the VM name or the template name)	PUT	Action	Tenant
ValidateForGoldPattern	Runs a series of checks against this pattern's VM to confirm that it is valid. Validates that the Gold pattern is in the correct state and is still a valid gold pattern	GET	Action	Tenant
VirtualMachine	Retrieves the virtual machine associated with this pattern	GET	Association	Tenant

Properties

Name	Description	Data Type
companyName	The company name	String
dataCenterName	The name of the data center where this pattern is available	String
dateCreated	The date the pattern was created	String
deleted	Specifies if this pattern is deleted	Boolean
dependantPoolIds	A collection of pool ids where the pattern is being currently used	Collection of String
desktopPools	The desktop pools using this gold pattern	Collection of DtDesktopPool
enabled	Specifies if this pattern can be used for pool creation	Boolean
hostManagerId	The host manager's ID	String
inventoryPath	The inventory path of the gold pattern	String
key	The windows key for the virtual machine referenced by the pattern	String
lastUpdated	The last update time and date	String

md5sum	The MD5 checksum of this gold pattern	String
modified	Specifies if the pattern has been modified	Boolean
name	The name of the pattern as known by the hypervisor (e.g. the VM name or the template name)	String
notes	The notes associated with this gold pattern	String
osType	The operating system type of this gold pattern	String
parentId	The ID of the parent pattern from which this gold pattern is derived	String
templateId	The templateId of the pattern	String
timeZoneId	The time zone ID of this gold pattern	String
username	The local admin username	String
version	Gold Pattern Version	Long
vmUuid	The UUID of the VM referenced by this pattern	String

4.5.7 DtMapping

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
autoGenerated	Indicates mapping was auto generated or not.	Boolean
deleted	Current state of mapping. If pattern is deleted the mapping will be flagged as deleted.	Boolean
pattern	Desktop pattern associated with this mapping	DtStaticDesktopPattern
user	User associated with this mapping	DtUser

4.5.8 DtPatternFilter

This interface is used as a filtering criteria to search desktop patterns.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
desktopmanagerId	Desktop manager Id for the filter	String
goldPatternCapableOnly	Indicates whether filter criteria is set to search only patterns are applicable for gold pattern conversion or not	boolean

poolIds	Pool Ids assigned for the filter	List<Long>
searchName	Search name for the filter	String

4.5.9 DtPoolTask

An asynchronous task performed on a pool. Contains information about the task, the targets, the progress of the task, and the result after the task completes.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DtPoolTask	A link to this resource	GET	Self	Tenant
Refresh	Refreshes the task's status with latest information about the progress of the task	POST	Action	Tenant

Properties

Name	Description	Data Type
desktopPoolId	The pool that the task acts upon	String
percentageComplete	The percentage of the task that is complete	Int
pool	Link to the pool	DtLink
startDate	Start date and time of task	Date
status	The current status of this pool task.	DtTaskStatus
statusDescription	The description of the current state, including reasons for failures	String
type	The type of the pool task	String

4.5.10 DtRemoteApplication

Remote application configured and served in a desktop or session pool

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DtRemoteApplication	A link to this resource	GET	Self	Tenant

Properties

Name	Description	Data Type
commandLine	Command to launch this remote application using CLI	String
iconLocation	Location of the icon that would represent this remote application. Limitation: may not be longer than 2048 characters	String
name	Name of this remote application	String

poolId	Unique identifier of the desktop or session pool that serves this remote application	Long
primaryFileLocation	Location of the primary executable file	String

4.5.11 DtSessionPool

A pool of RDS Servers providing session to users.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
AssignGroup	Assigns the specified user group to this desktop pool	PUT	Action	Tenant
AssignUserToPool	Assigns a specific user to a virtual desktop pool	PUT	Action	Tenant
Delete	Deletes this desktop pool and any relationships this pool had with other resources	DELETE	Action	Tenant
DtSessionPool	Link to this resource	GET	Self	Tenant
GroupsOfPool	Retrieves the groups currently assigned to the pool	GET	Association	Tenant
RemoveGroup	Removes the specified user group from this desktop pool	PUT	Action	Tenant
RemoveUserFromPool	Removes a specific user from a virtual desktop pool	PUT	Action	Tenant
Update	Updates this pool and any new relationships this pool has with other resources	PUT	Action	Tenant
UpdateServerDensity	Updates the existing session pool with a new server density	POST	Action	Tenant
UsersOfPool	Retrieves the users currently assigned to the pool. Users that are not directly assigned to this pool, via a group or only a desktop mapping, will not be returned.	GET	Association	Tenant
VirtualMachines	Retrieves the Virtual Machines in specified desktop pool	GET	Association	Tenant

Properties

Name	Description	Data Type
actualSize	Actual size of the pool (i.e. the actual number of provisioned virtual desktops that are available)	Long
dateCreated	Date the pool was created	Date
defaultPoolProtocol	Default remote display protocol for this pool	DtDisplayProtocol
domainName	Security domain name for the desktops in this pool	String

highlyAvailable	Specifies if the pool has high availability	Boolean
lastUpdated	Date the pool was last updated	Date
name	Pool name	String
patternType	Pattern type of this pool (static or dynamic)	DtPatternType
poolModeMessage	Reason that the pool is online or offline	String
poolOnline	Specifies if the pool is online or offline	Boolean
poolSessionType	Pool session type (e.g. desktop, application, both)	DtPoolSessionType
poolSizeType	Pool size type (e.g. Fixed, Elastic)	DtPoolSizeType
serverCount	Number of servers to be provisioned for this pool	Long
serverDensity	Number of users that can be supported by RDS Server in the pool	Long
sessionBased	Indicates whether the pool is session-based	Boolean
supportedPoolProtocols	Remote access protocols that can be used to access virtual desktops from this pool. This list must be a subset (or the same) as the supported protocols in the pool's DtDesktopModel.	List<DtDisplayProtocol>
usedByDataCenterIds	IDs of data centers where this pool can have desktop instances	Collection
vmRootName	Base name for all virtual machines provisioned from this pool	String

4.5.12 DtStaticDesktopPattern

A virtual desktop with static user assignments. The virtual desktop is available only to users explicitly assigned to it.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
AssignUserToDesktopPattern	Assign a specific user to a static virtual desktop	PUT	Action	Tenant
AssignedUsers	Retrieves all users in the given domain that have been assigned the use of this static desktop pattern	GET	Action	Tenant
DataCenters	Retrieves the data centers of this pattern	GET	Association	Tenant
Delete	Deletes the pattern and its backing virtual machine	POST	Action	Tenant
DtStaticDesktopPattern	A link to this resource	GET	Self	Tenant
Rename	Assigns a name to the pattern as would be known in the hypervisor (e.g. the VM name or the template name)	PUT	Action	Tenant

RemoveUserFromDesktopPattern	Remove a specific user from a static virtual desktop.	PUT	Action	Tenant
ValidateForGoldPattern	Runs a series of checks against this pattern's VM to confirm that it is valid and ready for conversion.	GET	Action	Tenant
VirtualMachine	Retrieves the virtual machine associated with this pattern	GET	Association	Tenant

Properties

Name	Description	Data Type
dataCenterName	The name of the data center where this pattern is available	String
dateCreated	The date the pattern was created	String
deleted	Specifies if this pattern is deleted	Boolean
dependantPoolIds	A collection of pool ids where the pattern is being currently used	Collection of String
desktopPoolId	The ID of the desktop pool of this desktop pattern	String
hostmanagerId	The host manager's ID	String
lastUpdated	The last update time and date	String
modified	Specifies if the pattern has been modified	Boolean
name	The name of the pattern as known by the hypervisor (e.g. the VM name or the template name)	String
osType	The operating system type of this pattern.	String
previousDesktopPoolId	The pool ID of this pattern prior to migration to another pool	String
templateId	The templateId of the pattern	String
vmUuid	The UUID of the VM referenced by this pattern	String

4.5.13 DtVirtualMachine

An active virtual machine used by the end-user as a virtual desktop or virtual server. It provides information about a virtual machine as it is used on a hypervisor.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
PerformOperation	Executes the specified power operation upon this virtual machine	POST	Action	Tenant
DtVirtualMachine	A link to this DtVirtualMachine	GET	Self	Tenant

Properties

Name	Description	Data Type
computePoolId	Compute Pool ID for the virtual machine	String

daaSAgentState	DaaS Agent state for the virtual machine	DtDaaSAgentState
daaSAgentVersion	DaaS Agent Version for the virtual machine	Long
description	Description of the VM	String
goldPatternVersion	Gold Pattern Version of the VM	Long
guestOS	Guest OS for the virtual machine	DtGuestOS
inventoryPath	Inventory path of the VM	String
ipAddress	VM IP address	String
memorySizeMB	Memory size in MB for the virtual machine	Int
name	VM name	String
networkId	Retrieves the Network ID for the virtual machine	String
numCPUs	Number of CPUs	Int
patternId	Pattern ID	String
powerOnDate	Date the last time this VM was powered on	Date
refId	VM reference ID	String
sealDate	Date this VM was sealed on for use as a Gold Pattern	Date
templateId	ID of the template	String
threeDEnabled	State of Soft 3D Graphics enablement	Boolean
type	VM type	String
vmHardwareVersion	VMware vmx version (hardware version) for the virtual machine	String
vmId	VM ID	String
vmLifeState	VM life state	DtVMLifeState
vmPath	VM path	String
vmPowerState	VM power state	DtVMPowerState
vmwareToolsState	VMware Tools State for the virtual machine	DtVMLifeState

4.5.14 DtVirtualMachineFilter

Filtering criteria to search virtual machines.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
searchName	Search name for this filter	String

4.6 DtQuotaManager

A top level entry point to manage tenant quotas.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
createOrUpdateDesktopManagerDMQuota	Sets or updates a collection of Quotas for specified a Desktop Model to the requested amount for specified Desktop Manger	POST	Action	SP
createQuotaFilter	Creates an empty DtQuotaFilter	GET	Action	All
desktopModelQuotas	Retrieves a collection of all desktop model quotas	GET	Association	All
desktopModelQuotasByFilter	Retrieves a collection of desktop model quotas based on the filter parameters	POST	Association	All
protocolQuota	Retrieves the Remote Protocol Quota for this Tenant	GET	Association	Tenant
templateQuota	Retrieves the Template Quota associated with the specified Tenant Datacenter "Tenant Cluster"	GET	Association	All
templateQuotas	Retrieves a collection of all template quotas	GET	Association	All

Properties

NA

4.6.1 DtDesktopModelQuota

Contains information about the number of virtual infrastructure components (like virtual machines) allowed for a tenant.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
dataCenterId	The data center ID of this desktop model quota	String
dateCreated	The date the desktop model quota was created	String

desktopManagerId	The desktop manager ID associated with this desktop model quota.	String
desktopModelId	The unique ID of the desktop model to which this quota is attached	String
lastUpdated	The date the desktop model quota was last updated	String
organizationId	The organization ID of this desktop model quota	String
quota	The quota associated with this desktop model for the associated Tenant Desktop Manager	Long
remainingVMCount	Count for remaining VMs that can be provisioned with this desktop model quota	Long
vmCount	The virtual machine count of this desktop model quota	Long

4.6.2 DtQuotaFilter

Filtering criteria for quota information.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
datacenterId	Datacenter Id for this filter	String
desktopManagerId	Desktop Manager Id for this filter	String
desktopModelId	List of Desktop Model Ids for this filter	List<String>
orgId	Organization Id for this filter	Long

4.6.3 DtRemoteProtocolQuota

Remote protocol quota assigned to tenant clusters (datacenter + organization) on the usage of remote protocols.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
dcId	Unique ID of the datacenter of this remote protocol quota	String
displayProtocol	Display protocol of this remote protocol quota	DtDisplayProtocol
numVms	The quota amount that applies to this display protocol for the given datacenter and organization	Long

orgId	Unique identifier of the organization of this remote protocol quota	Long
protocolId	Unique identifier of the protocol of this remote protocol quota	Long
Unlimited	Specifies if the quota is unlimited	Boolean

4.6.4 DtTemplateQuota

Template quota assigned to a tenant cluster (organization + datacenter).

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
DCId	Unique ID of the datacenter of this template quota.	String
dateCreated	Creation date of this template quota.	Date
dateUpdated	Last update of this template quota.	Date
numTemplates	The maximum number of templates allowed for this tenant cluster.	Long
orgId	Unique ID of the organization of this templates quota.	Long
remainingCount	The remaining available template quota	Long

4.7 DtReportingManager

A top level entry point to manage reporting features in the DaaS Platform.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
AvailableReportDates	Retrieves all available report dates for tenant organization (Tenant only)	GET	Association	Tenant
CreateBillingReportFilter	Creates an empty DtBillingReportFilter to retrieve billing reports	GET	Action	Both
CreateConcurrentUsersFilter	Creates a new filter to retrieve concurrent users report	GET	Action	SP
CreateReportFilter	Creates a generic Report Filter	GET	Action	Both
CreateReportKey	Creates a generic Report Key	GET	Action	Both
CreateUserEventReportFilter	Creates a filter to help retrieve specific groups of user events	POST	Action	Both
CurrentReportDate	Retrieves the active cycle's report date in format yyyyMM	GET	Action	Both

DtReportingManager	A link to this resource	GET	Self	Both
LastCompletedReportDate	Retrieves last month's report date in format yyyyMM	GET	Action	Both
MaxConcurrentUsersReport	Retrieves the maximum concurrent users for a given DtConcurrentUsersFilter	POST	Association	SP
QuotaBillingReports	Retrieves a list of the latest quota billing reports based on the given DtBillingReportFilter	POST	Association	SP
SuperTenantBillingReports	Retrieves a list of super tenant billing reports based on the given DtBillingReportFilter	POST	Association	Both
TenantReports	Retrieves all tenant reports filtered by the given report filter	POST	Association	Both
UserEventReports	Retrieves user event reports based on a given UserEventReportFilter	POST	Action	Both

Properties

NA

4.7.1 DtBillingReportFilter

A filter object that specifies the criterion to retrieve billing reports.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
customerIds	List of customer ids to restrict the results while retrieving billing reports. This must be used only in conjunction with super tenants. The customer ids are URL encoded. (Super Tenant only)	Collection of Strings
organizationIds	List of organization ids to restrict the results while retrieving billing reports. (Service Provider only)	Collection of Longs

4.7.2 DtConcurrentUsersFilter

A filter object that specifies the criterion to retrieve concurrent users.

Scope: Service Provider

Links

NA

Properties

Name	Description	Data Type
beginDate	The start date to filter concurrent users when retrieving reports within a time-window. The date should be in the format - "2013-06-14T12:00:00.000Z"	Date
endDate	The end date to filter concurrent users when retrieving reports within a time-window. The date should be in the format - "2013-06-14T12:00:00.000Z"	Date

4.7.3 DtConcurrentUsersReport

A report of the maximum number of concurrent users in a given time window.

Scope: Service Provider

Links

NA

Properties

Name	Description	Data Type
maxConcurrentUsersCount	The count of the maximum number of users concurrently connected	Long
tenantExceptions	A comma-separated string of tenant organization ids that were not reachable at the time of computing the max concurrent users count	String

4.7.4 DtQuotaBillingReport

Service provider report that provides billing information related to quotas (desktop model quota, session quota, protocol quota and template quota).

Scope: Service Provider

Links

NA

Properties

Name	Description	Data Type
datacenterId	datacenter ID pertaining to this report	String
disabled	Specifies if the organization is currently disabled	Boolean
errorReport	Specifies if this is an error report	Boolean
inUseCount	The count of the amount of quota in use by the organization	Long
organizationId	Organization ID of the super tenant	Long
protocolType	The protocol type of this report, available only if the quotaType is DtQuotaBillingType.PROTOCOL	DtDisplayProtocol

quota	Assigned quota to the organization for this quota type, -1 indicates unlimited quota	Long
quotaId	The ID of the quota associated with this billing record, if applicable	String
quotaType	The type of the quota for this record	DtQuotaBillingType
refId	Reference ID for this report	String
snapshotId	Retrieves the snapshot ID for this report. It is formatted as "YYYYMMddhhmm"	String

4.7.5 DtReportFilter

A filter object that specifies the criterion to retrieve tenant reports.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
reportKeys	A collection of DtTenantReportKey instances that defines the filter	Collection of DtTenantReportKey

4.7.6 DtSuperTenantBillingReport

Collects billing data for super tenants by their customer ids.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
customerId	Sub-tenant customer ID pertaining to this record	String
desktopCount	List of desktop model to the in-use count of those desktop models by this customer in a super tenant. Count for each desktop model is wrapped within DtDesktopCountWrapper instances.	Collection of DtDesktopCountWrapper
organizationId	Organization ID of the super tenant	Long
sessionCount	Count of the number of sessions allocated to this customer	Long

4.7.7 DtTenantReport

Tenant report for billing purposes. Each instance contains information about a single month's activities.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DtTenantReport	A link to this resource	GET	Self	Both

Properties

Name	Description	Data Type
desktopLoginCount	Number of logins per (VDI) desktop	Long
desktopRemoteAppLoginCount	Number of remote application logins serviced by (VDI) desktops	Long
reportKey	ReportKey containing the organization and the date of the report	DtTenantReportKey
sessionLoginCount	Number of logins to session based desktops	Long
sessionRemoteAppLoginCount	Number of remote application logins served by session based desktop	Long

4.7.8 DtTenantReportKey

Specifies the organization ID and report date used to filter tenant reports.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
orgId	Unique ID of an organization	Long
reportDate	The report cycle month for a report in the format yyyyMM	Integer

4.7.9 DtUserEventReport

Contains data pertaining to a user event. See DtUserEvent for all supported event types.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DtUserEventReport	A link to this resource	GET	Self	Both

Properties

Name	Description	Data Type
datacenterName	Name of the datacenter where the event originated	String
displayProtocol	Display protocol associated with the event	DtDisplayProtocol
endpointIPAddress	IP address of the endpoint that originated this event	String
endpointPlatformType	Endpoint type	DtEndpointPlatformType
eventError	Error during the event, if any	DtUserEventError
eventErrorDetails	Details about the event error, if any	String
getVirtualMachineName	Name of virtual machine associated with the event	String
patternId	Pattern Id associated with this event	String
poolName	Name of the pool associated with the event	String
remoteAppName	Name of remote application associated with the event	String
reportTimestamp	Timestamp of the report	Date
userEvent	Event type	DtUserEvent
userGuid	GUID of the user associated with the event	String
virtualMachineId	ID of virtual machine associated with the event	String
vmSessionId	The virtual machine session ID of this event, or 'unavailable' if there's none	String

4.7.10 DtUserEventReportFilter

A filter object that specifies the criterion to retrieve user event reports.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
displayProtocol	The display protocol to filter in the event reports	DtDisplayProtocol
eventTypes	List of DtUserEvent types to filter	Collection of DtUserEvent
includePoolName	Indicates whether the pool name will be included in the output.	Boolean
includeVirtualMachineName	Indicates whether the virtual machine name will be included in the output.	Boolean
numberReportDays	The number of report days from the current day (inclusive) to fetch. One day is interpreted as 24 hours from the current date and time.	Integer
patternId	The pattern Id to filter	String

poolId	The pool Id to filter	String
remoteAppName	Name of the remote application to filter	String
userGuid	The GUID of the user to filter	String
virtualMachineId	The virtual machine's UUID to filter	String

4.8 DtSecurityManager

A top level entry point to traverse the DaaS security object model.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DefaultDomain	Sets a default domain for the local appliance datacenter	PUT	Action	Tenant
Domains	Retrieves a collection of security domains	GET	Association	Both
DtSecurityManager	Link to this resource	GET	Self	Both
Organizations	Retrieves collection of all available organizations	GET	Association	Both
RegisterDomain	Registers a new domain to the local appliance's organization. The registration would enable authentication checks against the directory service specified in the domain. The first domain must be registered using the DtSystemManager's RegisterDomain link.	POST	Action	Both
UserById	Retrieves a DtUser by its userId (GUID)	GET	Action	Both
UsersByIds	Retrieves multiple DtUsers by their userIds (GUID)	POST	map	Both

Properties

NA

4.8.1 DtDomain

A security domain.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DtDomain	A link to this Domain	GET	Self	Both
Groups	Retrieves groups in this domain	GET	Association	Both
GroupsMatchingSubstring	Retrieves groups in this domain matching search string	GET	Association	Both

GroupsMatchingSubstringFromCache	Retrieves groups in tenant cache matching search string	GET	Association	Both
Save	Persist the properties of this domain		Action	Both
Users	Retrieves the user account names registered in this domain	GET	Association	Both
UserByLoginName	Retrieves the user registered to this domain with the given login name	GET	Action	Both
UsersMatchingSubstring	Retrieves user registered to this domain matching search string	GET	Association	Both
UsersMatchingSubstringFromCache	Retrieves users internet cache matching search string	GET	Association	Both

Properties

Name	Description	Data Type
adminGroups	A collection of groups with administrative privileges.	Collection of DtGroup
contextRoot	The context root of the directory server associated with this domain.	String
dateCreated	The date the domain was created domain.	String
directoryAccessPort	The directory server access port for this domain.	String
directoryAccessProtocol	The directory access protocol of this domain.	String
directoryServerIps	A list of directory server IPs used by this domain	Collection of String
directoryServerName	The directory server name of this domain	String
dnsServers	The list of DNS servers for this domain	String
domainSuffix	The domain suffix	String
lastUpdated	The date the domain was last updated	String
maxQueryRange	The maximum number of attributes returned when doing an LDAP query in this domain	String
name	The name of the Domain	String
serviceAccounts	A collection of service accounts of the domain on the directory server	Collection of DtDomainAccount
sysPrepAccounts	A collection of accounts used to prepare images for virtual desktop creation	Collection of DtDomainAccount
userGroups	A collection of groups with user privileges	Collection of DtGroup

4.8.1.1 DtDomainAccount

An account used by a domain, containing information, such as unique identifier and the password.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
type	The type of this domain account	DtDomainAccountType

4.8.1.2 DtGroup

A security group with a role that can be assigned to users to give them privileges.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
assignPool	Assigns this group to a desktop pool	PUT	Action	Tenant
DtGroup	A link to this resource	GET	Self	Both
removePool	Removes a desktop pool assignment from this group	PUT	Action	Tenant

Properties

Name	Description	Data Type
domainName	The container security domain name of this group, for example DEV	String
name	The name of the group, for example cn=enterprise admins,cn=users	String
role	A set of permissions or privileges that can be assigned to a group. There is a one-one association between a group and a role	DtRole

4.8.1.3 DtRole

A set of permissions or privileges that can be assigned to a group.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
creationDate	The date of creation of the role	String
lastUpdate	The date of the last update to the role	String
name	The name of the role	String
permissions	The permissions that a group of users possess when this role is assigned to the group	String

4.8.1.4 DtUser

A security domain user. A user or its group is typically assigned to a desktop pool that serves the virtual desktop for the user. However, a user may or may not have an active session. And a user may or may not have an associated virtual machine.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
assignPool	Assigns this user to a desktop pool	PUT	Action	Tenant
assignStaticDesktop	Assigns a virtual desktop to this user	PUT	Action	Tenant
defaultMapping	Retrieves this user's default desktop pattern or pool	GET	Association	Tenant
defaultMapping	Assigns the default pattern or pool to be used for desktops associated with this user	PUT	Action	Tenant
displayProtocol	Retrieves the display protocol preference of this group for the given desktop pattern	GET	Action	Tenant
displayProtocol	Assigns the given protocol as a preferred display protocol for this user	PUT	Action	Tenant
desktopPatterns	Retrieves the desktop patterns assigned to this user	GET	Association	Tenant
desktopPools	Retrieves the desktop pools assigned to this user [Deprecated]	GET	Association	Tenant
DtUser	A link to this resource	GET	Self	Both

pools	Retrieves the pools assigned to this user	GET	Association	Tenant
removePool	Removes a desktop pool assignment from this user	PUT	Action	Tenant
removeStaticDesktop	Removes a virtual desktop assigned to this user	PUT	Action	Tenant
updateStaticDesktopAssignment	Updates the static desktop mapping of a user from an existing assignment to a new static desktop	PUT	Action	Tenant

Properties

Name	Description	Data Type
domainName	The security domain of the user, for example DEV	String
loginName	The name used by the user to login	String
userDn	The LDAP distinguished name of the user	String

4.8.2 DtOrganization

A completely separate business entity with its own disjoint set of resources.

Scope: Service Provider and Tenant

Links

Name	Description	Method	Relationship	Scope
DesktopModelQuotas	A list of desktop model quotas available to this organization	GET	Association	Both
DtOrganization	A link to this DtOrganization	GET	Self	Both

Properties

Name	Description	Data Type
adminContact	The administrative contact of this organization (consists of email address, name, and phone number)	DtContact
brandedCSSURL	The URL to the CSS file used for this specific organization	String
businessUrl	The business URL of this organization	String
crmURL	The Customer Relationship Management URL of this organization	String
customFields	All custom fields in a key-value pair specific to this organization	Map of String
dateCreated	The date the organization was created, for example 2011-09-01T20:04:27.207Z	String
dateUpdated	The date the organization was last updated, for example 2011-09-01T20:04:27.207Z	String
disabled	Specifies if this organization is disabled	Boolean
diskQuota	The disk quota of this organization	Integer

helpDesk	The help desk contact of this organization	DtContact
jmxPassword	The password used to use the dt-console JMX application for this organization	String
licenseFileURL	The URL to the license file used for this specific organization.	String
name	The name of this organization.	String
superTenant	Specifies if the organization is a super tenant	Boolean
support	The support contact of this organization	DtContact
supportURL	The support URL of this organization.	String
technicalContact	The technical contact of this organization.	DtContact
vmQuota	The virtual machine quota for this organization.	Integer

4.8.2.1 DtContact

Contact details for various functions in an organization, such as admin and support.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
cellPhone	The contact name	String
email	The contact email address	String
name	The contact name	String
phone	The contact phone	String

4.9 DtSessionManager

A top level entry point to manage active sessions.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
ActiveSessions	Retrieves a collection of active sessions based on the criterion in the filter. DtSessionFilter must be posted as an argument.	POST	Action	Tenant
CountIdleSessions	Counts the number of sessions who have been idle for at least idleTime	GET	Action	Tenant

CreateSessionFilter	Creates a session count filter which reduces the number of methods in FabricDelegate	GET	Action	Tenant
CountSessions	Retrieves Long off all active sessions	POST	Action	Tenant
Disconnect	Disconnects an active session (URL parameter is sessionId)	POST	Action	Tenant
Logoff	Logs off an active session (URL parameter is sessionId)	POST	Action	Tenant

Properties

NA

4.9.1 DtActiveSession

This interface represents all active sessions by user, pattern and user, pool and session host.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
DtActiveSession	Link to this resource	GET	Self	Tenant

Properties

Name	Description	Data Type
base64Id	The base 64 Id for the session	String
connectionType	Connection type for the session	String
domain	Domain for the session	String
elementId	Element Id for the session	String
lastActiveTime	The last active time for the session	Date
loginStatus	Login status for the session	String
loginTime	Login time for the session	Date
patternId	Pattern Id for the session	String
userId	User Id for the session	String
vmId	VM Id for the session	String
vmName	VM name for the session	String

4.9.2 DtSessionCountFilter

Filtering criteria for session count information.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
connectionType	Connection type for this filter	String
loginStatus	Login status for this filter	String
patternType	Pattern type for this filter	String
sessionBased	Determines how sessions must be counted - desktop based only/session based only/both	Boolean

4.9.3 DtSessionFilter

Filtering criteria for session information.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
poolId	Pool ID for this session filter	Long
userId	User ID for this session filter	String
vmId	Desktop Manager ID for this session filter	String

4.10 DtSettingsManager

A top level entry point to retrieve feature setting configurations and other setting information.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
getFeatureSetting	Gets the feature setting for the requested feature	GET	Action	Tenant

getMinimumDaaSAgentVersionForGoldPatternConversion	The minimum DaaS agent version required for gold pattern conversion	GET	Action	Tenant
changeBlastClient	Changes the Blast client in use.	PUT	Action	Tenant

Properties

NA

4.10.1 DtFeatureSetting

A feature's enablement and settings. Specifically, this interface tells the caller whether certain features are enabled and if so whether they are required.

Scope: Tenant**Links**

NA

Properties

Name	Description	Data Type
enabled	Specifies if the feature is enabled for use	Boolean
featureName	Feature Name associated with this setting. Value can be: NON_DEFAULT_BLAST_CLIENT, POOL_NETWORK_MAPPING, SOFT_3D_GRAPHICS	String
required	Specifies if the feature is a required component	Boolean

4.11 DtSystemManager

Represents the DaaS Platform and provides various miscellaneous functionality and information about the current installation.

Scope: Service Provider and Tenant**Links**

Name	Description	Method	Relationship	Scope
Certificate	Returns the tenant certificate	GET	Action	Tenant
DtSystemManager	Link to this resource	GET	Self	Both
Login	Performs basic authentication and includes the "Authorization" http header in the response that must be included in all http requests to secure resources and all REST API invocations	POST	Action	Both

Logout	Terminates any session associated with a previous login	DELETE	Action	Both
PlatformInformation	Retrieves the DaaS Platform information including the supported versions by the platform	GET	Top	Both
RegisterDomain	Registers the first domain to the local appliance's organization. The registration would enable authentication checks against the directory service specified in the domain. This call does not require the user to be authenticated as no domain exists at the time of invocation. Domain registration using this method is disallowed if a domain already exists for the organization. In that case, use the authenticated invocation DtSecurityManager's RegisterDomain.	POST	Action	Both
SystemVersionInformation	Gets the version information on the current system.	GET	Top	Both
TermsOfService	Gets the terms of service.	GET	Top	Tenant

Properties

NA

4.11.1 DtPlatform

The DtPlatform resource is the entry point for the API, accessed by the URI /system/platform. Once you login, the first HTTP request you make should access the URI /system/platform.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
defaultDomain	Default domain	String
domainNames	Collection of domain names registered to this organization for this data center	Collection
version	Collection of versions	Collection

4.11.2 DtSystemVersion

Version information on the system.

Scope: Service Provider and Tenant

Links

NA

Properties

Name	Description	Data Type
appVersion	Appliance version of the system	String
buildDate	Build date of the system	String
buildInfo	Build information of the system	String
buildNumber	Build number of the system	int
majorVersion	Major version of the system	int
minorVersion	Minor version of the system	int
patchVersion	Patch version of the system	int
releaseName	Name of the release running on the system	String
svnId	SVN ID of the system	String
version	Full version of the system	String

4.11.3 DtTermsOfService

Information related to whether the logged in user has accepted the terms of service.

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
Accept	Accept the current terms of service	PUT	Action	All
DtTermsOfService	Link to this resource	GET	Self	Tenant

Properties

Name	Description	Data Type
accepted	Indicates whether the logged in user has accepted the terms of service	Boolean
url	URL of the terms of service (or null if no terms of service are specified)	String

4.12 DtTaskManager

Scope: Tenant

Links

Name	Description	Method	Relationship	Scope
CreateTaskFilter	Creates an empty Task Filter	GET	Action	Tenant
Tasks	Retrieves all tasks that match the specified filter	POST	Association	Tenant

Properties

NA

4.12.1 DtTaskFilter

Filters tasks by various criteria.

Scope: Tenant

Links

NA

Properties

Name	Description	Data Type
fromDate	Earliest date from which tasks should be returned	Date
limit	Maximum number of tasks to be returned	Int
poolId	Pools for which tasks should be returned	Long
taskStatus	Task statuses for which tasks should be returned	Collection
toDate	Latest date from which tasks should be returned	Date

5 Enumerated Data Types

This chapter lists the enumerated types that are the legal values for certain resource attributes.

5.1 DtComputePoolType

Represents the list of compute pool types that can be discovered by the DaaS Platform

Properties

Name	Description
ESX	Direct ESX hypervisor host
VCENTER_CLUSTER	vCenter cluster
VCLLOUD_VDC	vCloud virtual datacenter

5.2 DtDisplayProtocol

Represents the list of display protocols that can be used to access virtual desktops from remote clients.

Properties

Name	Description
ICA	Citrix's Independent Computing Architecture
NX	The unique ID of the desktop model to which this license entitlement is applied
RDP	Remote desktop protocol
RGS	
VNC	Virtual network computing
PCOIP	Teradici's PC-over-IP
OTHER	
UNKNOWN	

5.3 DtDomainAccountType

Specifies the domain's account type. Value is one of the following:

- SERVICE
- SYS_PREP

5.4 DtEndpointPlatformType

Represents the platform types of the endpoint (end-user) device used to access a virtual desktop or remote application. Value is one of the following:

- ANDROID
- IOS
- LINUX
- MAC
- OTHER_MOBILE
- OTHER_NON_MOBILE
- THIN_CLIENT
- UBUNTU
- UNKNOWN
- WINDOWS

5.5 DtNetworkType

Indicates the type of network. Value is one of the following:

- VLAN - VLAN network
- SDN - software defined network

5.6 DtPatternType

Specifies the types of patterns.

Properties

Name	Description
A	(App) A pattern that describes a desktop or application served by an app server that describes a non-VM desktop machine
D	(Dynamic) A pattern that describes a dynamic virtual desktop machine
G	(Gold) A pattern based off of a template from the hypervisor
S	(Static) A pattern that describes a static virtual desktop machine
U	U (User) A pattern that describes a non-VM desktop machine

5.7 DtPoolSizeType

Specifies the types of pool sizes. Value is one of the following:

- Elastic
- Fixed
- Test

5.8 DtQuotaBillingType

Specifies the types of quota that can be used for billing reports. Value is one of the following:

- DESKTOP_MODEL
- SESSION
- TEMPLATE
- PROTOCOL

5.9 DtTaskStatus

Specifies the current status of this pool task. Value is one of the following:

- FAILED
- OTHER
- RUNNING
- SUCCESSFUL

5.10 DtUserEvent

Specifies the user event types that are reported by the platform. Values are described below.

Value	Description
ALLOCATE	VM or RDS session is allocated to a user successfully
ALLOCATE_FAILURE	VM or RDS session fails to be allocated to a user
DISCONNECT	User disconnects from a VM or RDS session
LOG_OFF	User logs off of a VM or RDS session
LOG_ON	User logs on to a VM or RDS session
RECONNECT	User reconnects to an existing VM or RDS session
SESSION_TIMEOUT	VM or RDS session times out due to lack of activity
UNKNOWN	

5.11 DtUserEventError

Specifies the type of user event errors that can occur when desktop allocation fails. Values are described below.

Value	Description
BROKER_PARAM_ERROR	Internal error when a parameter required for allocation is incorrect or missing
CONNECTION_TYPE_MISMATCH	Session to a full desktop or a remote application currently exists and attempt has been made to connect with the other type (full desktop or remote application)
FAILED_TO_LOCK_VM	VM for allocation could not be locked for exclusive access
DESKTOP_UNAVAILABLE	VM or RDS session cannot be allocated because either no VMs in a pool are available or an unrecognized error occurred
DESKTOP_NOT_RUNNING	VM for allocation was not powered on
GUEST_OS_NOT_RUNNING	Operating system of the VM is not running
VM_TOOLS_NOT_INSTALLED	VM tools have not been installed on the VM for allocation
VM_TOOLS_NOT_RUNNING	VM tools are not running on the VM for allocation
IP_ADDRESS_UNKNOWN	IP address of the VM to be allocated is not known or has not been reported by the hypervisor
IP_ADDRESS_UNREACHABLE	IP address of the VM to be allocated is not reachable
VM_STATE_ERROR	Allocation state of the VM is not in the required state (generally, "AVAILABLE")
AGENT_STATE_ERROR	DaaS Agent of the VM is not in the required state ("ACTIVE")
INCOMPATIBLE_PROTOCOL	Reconnect to an existing VM or RDS session and requested protocol is not compatible with the one previously used to establish the session. User must be logged off of the existing session before proceeding with the new session.
PCOIP_AGENT_ERROR	PCoIP agent is not active on the VM or there was an error with the PCoIP gateway on the VM
RAM_SERVICE_FAILURE	RAM service failure is when a desktop cannot be reached through the dtRAM appliance

5.12 DtVMLifeState

Specifies the various states of the virtual machine. Value is one of the following:

- CLONE_FAILED
- CLONING
- DESTROYING
- JOINING_DOMAIN
- NOT_IN_DOMAIN
- OFF
- READY

- STARTING
- UNKNOWN

5.13 DtVMPowerState

Specifies the states of power for a VM. Value is one of the following:

- POWER_STATE_POWERING_OFF
- POWER_STATE_POWERING_ON
- POWER_STATE_RESETTING
- POWER_STATE_SUSPENDING
- POWERED_OFF
- POWERED_ON
- SUSPENDED
- SYS_PREP