UForge AppCenter Admin Documentation

Release 3.6

FUJITSU

February 24, 2017

This documentation is intended for Administrators of the UForge AppCenter. It covers an architecture presentation, installation instructions, as well as all the necessary steps to configure, manage and monitor your AppCenter once installed.

Note: There are multiple options for reading this documentation - click on the link at the lower left hand corner for these options.

Contents:

Architecture Considerations

The following sections describe in detail things to consider when deploying a UForge platform.

UForge Platform Overview

UForge is a scalable multi-tenant platform. UForge can be split into the following distinct parts.

- UForge Server – This contains all the business logic of UForge, handling all incoming user requests.
- Meta-data SQL Store – A database holding all the configuration information and data of the platform.
- LDAP Service – LDAP holding user authentication and access information
- IDM Service – Authentication and authorization module
- Generation Cluster – A grid engine for scheduling and executing image generations
- Event-bus – Rabbit MQ
- Proxy cache - Squid

The UForge platform can be deployed on physical machines or in a virtualized or cloud environment.
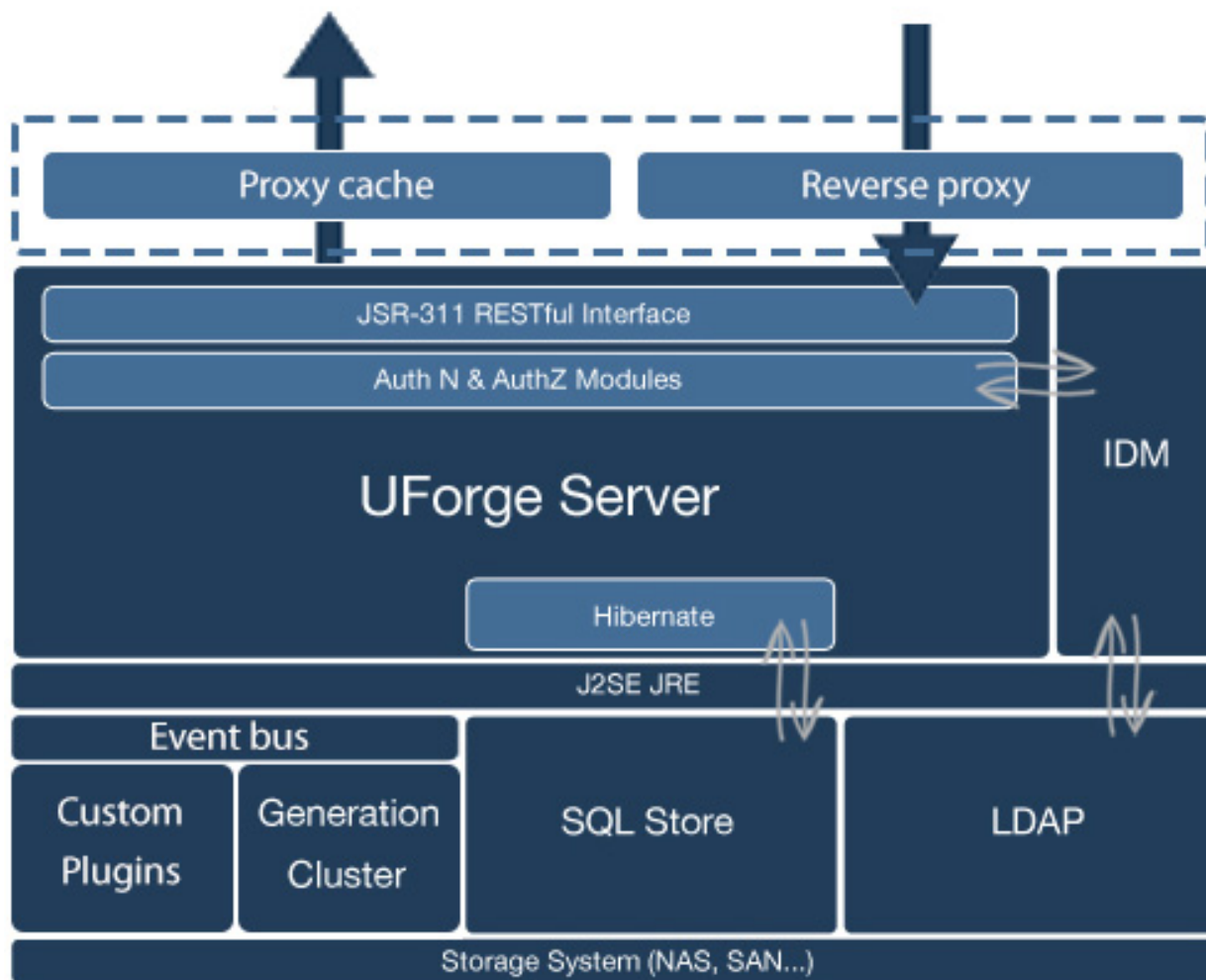
UForge Server. The UForge Server is a RESTful (Representational State Transfer) web service built on top of Java and using the JSR-311 reference implementation (project Jersey). UForge Server is based on the design principles of REST and Resource Orientated Architecture (ROA). Resources are references with a unique global identifier (URI). UForge Server uses the semantics of the HTTP protocol to manipulate these resources. The HTTP response codes are used to determine whether a user's request was treated successfully or not.

Information is returned to the client in either XML or JSON, depending on the `Accept-Type` header attribute used by the client. If no `Accept-Type` header is provided, XML is returned by default. To ensure security, communication with the UForge Server is done via HTTPS. UForge Server provides authentication (AuthN) and authorization (AuthZ) modules. These modules can be customized by the customer to provide or use an alternative mechanism for authentication and authorization. By default these modules communicate with the IDM service. This service determines whether a request has the correct authentication and the correct access.

The UForge Server interacts with the SQL Store using **Hibernate**. Hibernate is a high-performance Object/Relational persistence and query service. Hibernate maps from Java classes to database tables and from Java data types to SQL data types. It provides data query and retrieval facilities, providing a buffer between the two data representations and enables a more elegant use of object orientation on the Java side. The UForge Server is deployed inside a web/application server container. UForge, by default, uses Tomcat as the application server container.

Proxy Cache. UForge AppCenter includes a proxy cache. The proxy used is Squid. It is used for caching and

---

centralizes all outgoing traffic. This improves the performance of UForge, specifically for the population of distribution repositories.

IDM Service. The UForge Identity management module is based on Apache Syncope (APL2 license). The module allows management of auditing (reports), policies, roles, users, tasks and entitlements. At present UForge uses this module for authentication and role-based access control (authorization). The persistence store for the UForge IDM is the SQL Store (described below) and there is a resource connector to the LDAP Service for storage of roles/users and entitlements in an ldap repository backend. The IDM service is deployed inside a web/application server container. UForge, by default, uses Tomcat as the application server container.

This IDM Service can be extended to provide a wider identity and access management (IAM) integration to an existing enterprise or corporate IAM system or to a brand new standards compliant IAM system(s) by using open source industry standard resource connectors.

LDAP. This service is an industry standard powerful AuthN ldap (v3) pure java server, based on the open source ForgeRock's OpenDJ offering. This can be run in single instance mode or multi-master replication mode for robustness.

SQL Store. This is a relational database holding all the meta-data of UForge. Meta-data includes such items as:

- Appliance information including which operating system packages are included, install profile, middleware and configuration information.

- Images that have been generated from an appliance

- Images that have been published to a virtual or cloud platform

- Package information for an operating system

- Third party project software components

By default MariaDB is used as the SQL Store.

Generation Cluster. Image generation is I/O intensive and may take several minutes to complete. Consequently an HPC cluster is used to execute image generation jobs. There are two parts to this cluster:

- One or more compute nodes to execute a generation job

- A resource management system or batch scheduler that manages the reservation and access to the compute nodes

The resource management system holds its configuration information inside the meta-data SQL Store. The generation cluster is based on an open source project called OAR that is heavily used in production clusters of over 5000 nodes capable of handling over 5 million jobs.

Event-bus. UForge AppCenter uses RabbitMQ as an event bus. This event bus allows UForge administrators to track a number of user events on the platform. For more information on managing Rabbit MQ see: http://rabbitmq.com

UForge Clients. UForge provides a client, called UForge Portal that connects to UForge Server via HTTPS. This provides an interactive, visual experience when designing appliances or application stacks that can be deployed on physical, virtual and cloud environments. Consequently, UForge does not need to provide any presentation information to the client, greatly reducing the information (and therefore bandwidth) sent to the client.
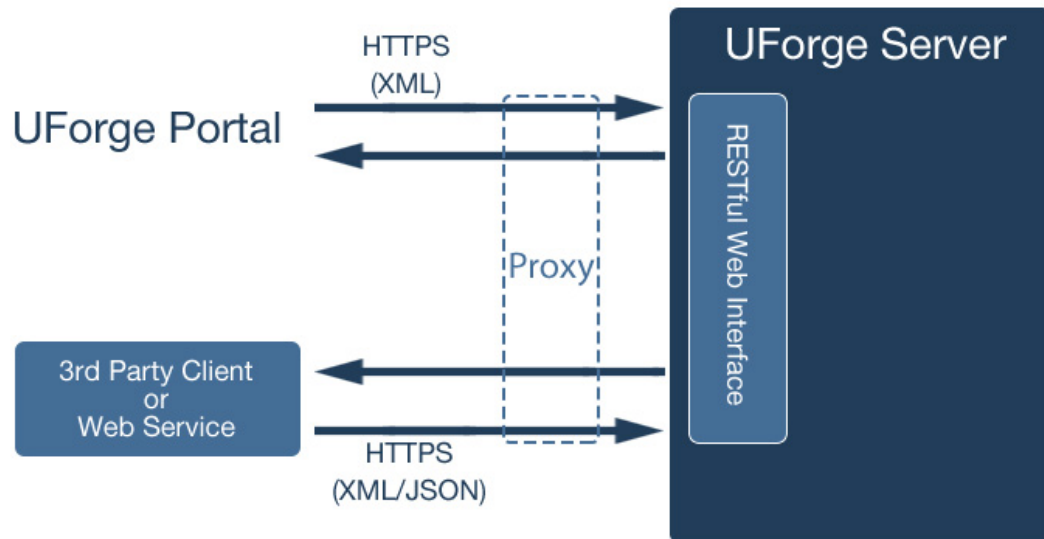
UForge allows users to implement their own clients to interact with UForge. UForge provides a RESTful API, allowing businesses to create a mashup and expose certain functionalities of UForge in their websites and portals. UForge provides two SDKs:

- Java

- Python

Users can use other languages to communicate with UForge.

Repository Caching

UForge AppCenter uses a repository cache in which it stores OS repositories as new images are generated. This means that the cache is empty when the AppCenter is installed and will be populated as images are generated by users. The cache ensures that all the versions a user needs are always available to generate images. As users generate images, the AppCenter connects to the official repositories to get the repositories and stores them in the cache.
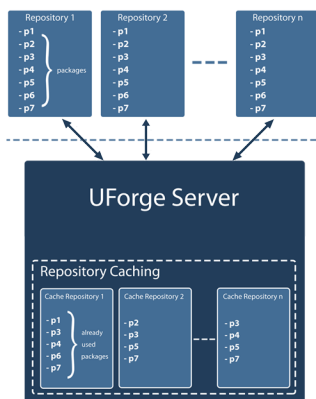
Infrastructure Setup

UForge can be installed either on physical machines or in a virtual or cloud environment. The minimal installation requirements for UForge are:

- One physical or virtual machine where UForge will be installed
- A NAS or SAN for storage

UForge AppCenter Node Prerequisites

The UForge AppCenter components can be run on one physical or virtual machine, or can be distributed over several physical or virtual machines for scaling and reliability.

The UForge AppCenter requires the following hardware:

- CPU: 64-bit, 8 or more cores
- RAM: 16GB or more
- Local Hard Drive: 400GB
- NAS/SAN Storage: 200 GB (though this might be much more depending upon the usage)

This is the minimum for an "all in one" solution. For more information, refer to *Minimum Software Topology*.

Reliability

Fault tolerance is an important consideration for large-scale deployments. UForge platform has several types of data that must be replicated.

- The meta-data stored in the SQL store. The meta-data SQL store is replicated by using either master-slave or clustering. Both these configurations are supported by Percona Server. These configurations help scale-out the system and provide a level of fault tolerance if one of the database servers fails.
- The LDAP service data can be replicated over multiple LDAP instances via MMR (Multi-Master Replication)
- Binary data including operating system packages, project packages, uploaded software packages, license files, generated images and logo images. The binary data is stored on a storage system. This can be on a local filesystem of the database or on a NAS or SAN. This data is transparently replicated using RAID techniques.

In order to make the web service tier fault tolerant, multiple web servers can be deployed and load balanced. The administrator may also wish to have multiple load balancers in case the load balancer itself fails.

Security

UForge communicates with clients using HTTPS to ensure a secure connection. However, when deploying UForge, other security measures should be considered:

- Add a firewall in front of the web service tier, to only expose the HTTPS port.
- Provide a logical sub-network to protect the database, LDAP, storage and generation cluster (DMZ).

Storage Considerations

The UForge AppCenter requires a variable amount of storage, depending, among other things, on the number of image generations and users. During the configuration it is important to size the local storage the various node instances making up the deployment and, if any, the size of the shared remote storage.

Local storage is used for installing the UForge service software and free disk space for log files.

> **Warning:** Logs for UForge, OpenDJ, Tomcat, and syncope are stored under /opt and NOT under /var. Under /var/log you will find a symlink to the /var location where the logs are stored. Therefore you should allow enough space under /var for these logs.

The SQL Store uses the local storage to store all the database meta-data. Finally a compute node used to generate an image requires local free disk space to generate one or more images from an appliance template.

A remote storage system (NAS or SAN) is used to store:

- Operating system cache repositories (this will be empty at the beginning and will grow as images are generated)
- Project catalog package binaries
- Binaries uploaded by users using UForge
- Generated images (copied from a compute node once the generation is complete)

For small deployments, the remote storage system can be on a local filesystem of the generation cluster.

Operating System Cache Repositories and Projects. The OS repositories are no longer stored in the UForge AppCenter. UForge AppCenter uses cache repositories in order to generate appliance templates. Therefore its size will grow and is completely variable. What consumes space are the images generated and the binaries uploaded in MySoftware. You can limit the space used by setting user quotas (refer to *Setting Quotas*).

The size of projects also vary, however they are usually in the 10 to 100 MB range. Consequently they take up a very small percentage of the entire disk space.

Image Generation. Compute nodes require local storage to create the disk images during the generation phase of an appliance template. To calculate the local storage required by a compute node, the generation of an image needs to be understood. There are six phases to the generation of an image:

- Phase 1: Check if the packages are in the cache. If they are not, download them and save to cache.
- Phase 2: Create a virtual disk for package installation.
- Phase 3: Install the packages, binaries and files to the disk (at this stage the local copy of the packages are deleted to save space).
- Phase 4: Convert the disk to the required target format.
- Phase 5: Compress the created disk.
- Phase 6: Copy the image to a more permanent storage (the remote storage so that the image can be downloaded or published by the user). At this stage the created disk is deleted.

At any given time during the construction of an image, the compute node requires three times as much space as the final image being generated. Since a compute node can generate more than one image simultaneously, the local disk storage for a compute node can be calculated as follows:

> average image size x simultaneous generations x 3

Image sizes vary from 300 MB to 12 GB. Note this is the disk size required to install all binary packages. A UForge compute node uses 'sparse' filesystem so that 'free' disk space required by a virtual machine does not need to be allocated during the generation of that virtual machine.

For a compute node configured to generate 10 images simultaneously and assuming that an average image size is 8 GB, the minimum local storage required by a compute node is:

8 x 10 x 3 = 240 GB

User Disk Space. Each user on UForge can generate images and also upload their own software. The user software is stored on the remote storage. Once an image has been generated it is also stored on remote storage allowing the user to download or publish the image directly to a virtual or cloud platform at a later stage. User uploaded software is usually a small percentage of the disk space compared to generated images. Note, that as the user has the appliance template, an image can be regenerated at any time, therefore, images can be deleted to save disk space. The total disk space required to store uploaded software and user images can be calculated as follows:

user disk quota x total number of users

For example, a UForge platform providing all the operating systems UForge supports for 100 users, each having a quota of 18 GB (equivalent to having a maximum of 15 stored images of 1GB each and 3GB of uploaded software), the remote storage required is: 1880 GB (1.88 TB). When calculating, you should consider an addition 10% margin, plus 20 GB for UShareSoft bits (installer, ISO, etc).

Depending on the replication policy used for the remote storage, the total remote storage may have to be larger. Note that it may not be necessary to replicate all the information stored in the remote storage. Only important information must be replicated, including:

- the uploaded user software

- operating system repositories (this may be optional if there are backups of the operating systems elsewhere.)

Images generated are usually not replicated, as the image can be regenerated from the appliance template stored in the SQL Store.

Scalability through Partitioning

UForge was designed from the ground up to scale to meet the needs of businesses and service providers with 100,000s of users. The key to scaling is partitioning. Effective partitioning is based on leveraging "locality of reference" for both processing and data – if certain servers are specialized to solve a subset of the bigger problem, then the essential code and date are more likely to be in memory or close at hand. Partitioning techniques include vertical partitioning of functional tasks and horizontal partitioning of data and associated processing. Partitioning also helps to implement security to the platform.

Partitioning is increased by other distributed system techniques like automated replication, load balancing and failover.

Vertical Partitioning allows complex processing tasks to be divided into subtasks that can be independently optimized and managed. Vertical partitioning in UForge primarily consists of off-loading or splitting the I/O intensive generation tier, web service tier and database tier.

This allows the administrator to scale-up independently the number of CPUs, RAM and disk size for each of the tiers.

Horizontal Partitioning is crucial for large scale deployments. UForge is horizontally partitioned for larger deployments where thousands of users are required to interact simultaneously with the system.

One of the big bottlenecks, however, in such architectures is the database. When scaling out the web service tier, the number database reads and writes increases, and therefore the database becomes saturated.

Firstly UForge uses Db sharding. This basically consists of splitting up the data into buckets (shards) that can be stored on more than one database instance. The second is caching (memcached) which is an in-memory key-value store for small chunks of arbitrary data from results of database calls. This reduces the amount of potential reads directly into the database.

To help scale out further you can also set up a database cluster, providing multiple database instances to the web service tier (the default database service does not support clustering).

Such bottlenecks can also be reduced by scaling-up (vertical partitioning) where more RAM and CPU is provided to the machine.

The generation cluster is intrinsically scalable, allowing the administrator to easily add new compute nodes to scale-out the number of simultaneous generations. OAR also provides the ability to deploy multiple schedulers and to configure them in master-slave mode.

Image Generations

Image generations are very I/O intensive compared to CPU, consequently only 1 core per simultaneous generation is required. Therefore for 10 simultaneous generations, a total of 10 cores are required for the compute notes.

The generation capacity of UForge depends on the total number of subscribers using the service. From experience, the ratio between generation capacity per week and the number of subscribers can be calculated as:

number of subscribers x 0.022 = generation capacity/week

This is assuming that UForge is used 24 hours a day. If the service is only used in one geography, then users will typically use the platform within an 8 hour period. Therefore the generation capacity will have to be multiplied by 3 (as the other 2/3 of the day, no generations will take place). For an 8 hour day, the generation capacity per week required for a certain number of subscribers can be calculated as:

number of subscribers x 0.066 = number of generations/week

On average, a generation takes about 5 minutes (this depends upon the size of the image being created and whether it requires to be compressed). For each core used to generate images, the total generation capacity per week can be calculated as:

12 x number of hours x 7 = generation capacity per core

For an 8 hour period:

12 x 8 x 7 = 672/week (per core)

For an 24 hour period:

12 x 24 x 7 = 2016/week (per core)

Estimating Scan Size

It is difficult to estimate sizing of scans. However, the following guidelines may be useful to plan your first scan:

average scan size x number of scans

Scan sizes vary from 100 KB to 12 GB

You should note that the data that will be scanned does not include operating systems known by UForge. For example, Debian or other OS data will not be included in the scan size.

Also, if you have already run a scan, only the delta information will be included.

Deployment Example

How to organize your UForge configuration depends on the specific customer needs. This may include:

- The number of users as well as the number of simultaneous connections to the platform

---

- The number of simultaneous generations

- The number of projects in the project catalog

- The SLA of UForge

- Whether the service needs to be reached over multiple sites

- Whether UForge is exposed to 3rd party customers and partners, or for internal use only

Note that UForge can be deployed on physical machines or on a virtual or cloud platform. The word 'node' describes either the specification of a physical machine or a virtual machine instance running in the virtual or cloud environment.

Network Topology

In order to optimize scalability, you should use a dedicated network subnet for UForge service.

If you want to deploy a public UForge service, you can use a public IP address for each component.

We recommend using a firewall with the NAT technology to map a public IP address to a private IP address. This allows you to use public IP addresses only for the Web Front-end and the Web Service. The other services stay hidden from the rest of the world.

For a private network, you can use a NAT translation, but it is better to use the firewall in gateway mode and filter the traffic. This way you can open ports on the different services.

By default UForge provides a proxy service, exposing the user interface (UI), REST web service tier and a download URL allowing external cloud platforms download generated machine images. This is configured automatically using the information you provide as part of the configuration phase (see *Configuring UForge*). These values can also be configured manually, for more information see *Modifying the UForge Platform External URL Endpoints*.
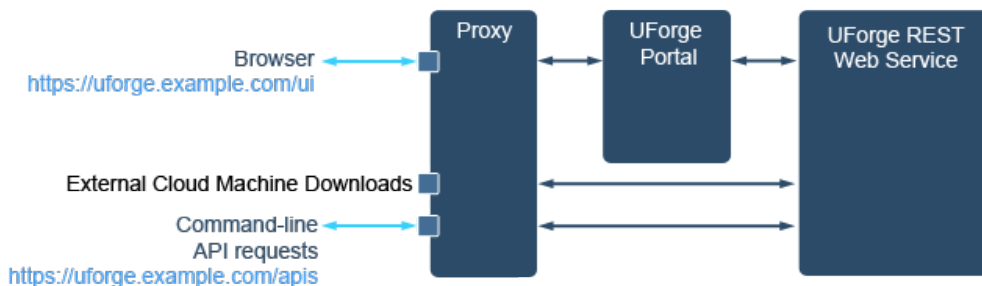
Minimum Software Topology

For the minimum software topology, use all services without security or fault tolerance.

For each of these components we recommend the following minimal (hardware or virtual) specification:

Component	RAM	CPU	Hard Drive	Comment
Web Front-End	3 GB	2	15 GB	
Web Service	3 GB	2	15 GB	
Generation Cluster (scheduler)	2 GB	1	15 GB	
Compute Node	4 GB	2	30 GB	If you have a good NAS/SAN you reduce the disk space to 15 GB if you map the directory /space/REPOS with the NAS/SAN
SYNCOPE Webservice	2 GB	2	15 GB	
Database	4 GB	4	100 GB	If you have a good NAS/SAN you reduce the disk space to 15 GB if you map the directory /space/REPOS with the NAS/SAN
LDAP	2 GB	2	15 GB	
NAS/SAN	2 GB	2		

It may be more logical to group the DB, LDAP and Generation Cluster (scheduler) on one node. Typically you can also group the Webservice and SYNCOPE Webservice.

Note: When a user creates an appliance, the packages are stored locally in the UForge cache repository, which is stored on the DB. Therefore, depending on the number of appliances created and OS used, you may need to adjust the DB size.

Note: If you are using high availability and you choose to split the Webservice and SYNCOPE on separate nodes, then you need a shared NAD/SAN (/tmp/userdata).

Security Options

To increase the security of the UForge service, you can add replication in master/slave of the database component and the LDAP component.

The minimal requirement is 1 slave for the database and 1 slave for the LDAP service. Of course, you can add more slaves for each of these services.

See the specification of these options:

Component	RAM	CPU	Hard Drive	Comment
Database Slave	2 GB	1	70 GB	If you have a good NAS/SAN you reduce the disk space to 15 GB if you map the directory /var/lib/mysql to the NAS/SAN
LDAP Slave	1 GB	1	15 GB	

High Availability

High Availability can be attained with the redundancy of all services. However, the database and the LDAP service have not yet been tested with the clustering mode.

For the moment we recommend using the "security options" for these services. This is a point of failure.

All other services can be clustered. For clustering a service you will need a pool of load balancers (physical or virtual).

Component	RAM	CPU	Hard Drive	Comment
Load Balancer	2 GB	2	15 GB	

All other services keep the same configuration, as seen above.

UForge Installation

To install a fully functioning UForge platform, you must install and configure the UForge services as well as populate the UForge Repository with the operating systems you wish to build your server templates from.

This section covers:

UForge Installation Overview

To install a fully functioning UForge platform, we provide the following 2 ISOs:

- UForge Setup ISO – this includes all the necessary elements to have a functioning "UForge in a Box"

- ISO which contains the skeleton of the distribution installers in case you want to create ISO images with UForge. This only needs to be installed if you want to create ISO images.

Installation Checklist

Before you start deploying UForge, ensure that you have all the following:

- UForge Setup ISO

- Your activation credentials (ID and activation key) provided by your vendor

- Architecture of the deployment (number of nodes and networking topology. See *Network Topology*)

- The necessary system requirements (see *Storage Considerations*)

- Your SSL certificates, key and chaining certificates, and all files corresponding to the following entries in /etc/httpd/conf.d/ssl.conf:

    - SSLCertificateFile

    - SSLCertificateKeyFile

    - SSLCACertificateFile

    - SSLCertificateChainFile (might be empty)

Installation Steps

UForge installation has three distinct phases:

Step 1: Install the UForge AppCenter for each node of the platform.

Step 2: Configure UForge AppCenter using the UForge Deployment Wizard.

Step 3: Additional configuration steps, as described in *Further AppCenter Configuration*.

UForge Repository Setup

The UForge Repository is a storage area containing:

- Operating system packages and updates
- Project catalog binaries
- User `My Software` binaries
- Generated images from users using the platform

The operating system and project binaries are separated from the `My Software` binaries and the images generated.

> **Warning:** When new projects are populated by the administrator after the initial install, they are copied in the same location as `My Software` and generated images.

The UForge Setup Disk contains all the operating system and default project binaries information that needs to be copied to the UForge Repository. This repository can be shared storage or in the case where the entire UForge platform is being installed on one machine, can be on the local disk.

If you want to have the UForge Repository setup on a local disk, you must first install UForge prior to setting up UForge Repository. Otherwise, if this is shared storage, the UForge Repository can be setup independently.

The UForge Repository must be setup properly prior to completing the final configuration step. You must first populate the UForge database with the operating system package meta-data before you run the configuration phase described in *Configuring UForge*.

You can set up the UForge Repository either:

- on a shared storage
- on a local storage

UForge Repository on Shared Storage

When using shared storage NFS is used to share the information between the various UForge nodes. To setup the NAS or SAN for the UForge Repository you must create two shared directories, one for the operating system data and the other for all the user data (`My Software` and images generated).

To setup the shared storage:

1. Log in to the machine where the NFS server is running

2. Create the operating system directory, for example: /volume1/DISTROS

   The following NFS options are required:

   ```
   *(rw,async,no_wdelay,no_root_squash,insecure_locks,anonuid=0,anongid=0)
   ```

3. Create the user data directory, for example: /volume1/USER_DATA

   The following NFS options are required:

```
*(rw,async,no_wdelay,no_root_squash,insecure_locks,anonuid=0,anongid=0)
```

4. Check the mount points:

```
mount 192.20.777.205:/appstore_qualif_data/ /mnt
su - tomcat
cd /mnt
```

5. Execute the command:

```
touch test
```

6. If it returns:

```
touch: cannot touch `test': Permission denied
```

Then execute the following commands (as root)

```
cd /mnt
chown -R tomcat:tomcat .
```

7. Confirm you can create a file on /mnt as user tomcat. Then perform the following:

```
cd ~
umount /mnt
```

UForge Repository on Local Storage

When using local storage, UForge must already be installed, but not configured. Create two directories one for the operating system data and the other for all the user data (for example `My Software` and images generated).

1. Create the operating system directory, for example: `/space/DISTROS`

2. Create the user data directory, for example: `/space/USER_DATA`

3. Copy the operating system information from the UForge setup ISO to the DISTROS sub-directory:

```
$ rsync -a --progress /<disk directory>/DISTROS/ /space/DISTROS/
```

Note: It may take up to an hour to copy all of the data.

---

4. If you want to create ISO images with UForge then you must copy the ISO skeleton files and mount them to your system, as follows:

Installing from an ISO

To install UForge from the ISO image:

1. Attach the ISO to a VM or burn the ISO to a DVD (for installing to a physical machine, note the machine will require to have a DVD disk drive).

2. Boot the system from the ISO.

3. Choose the default menu proposed by the installer.

---

4. Choose the keyboard layout you want to use.
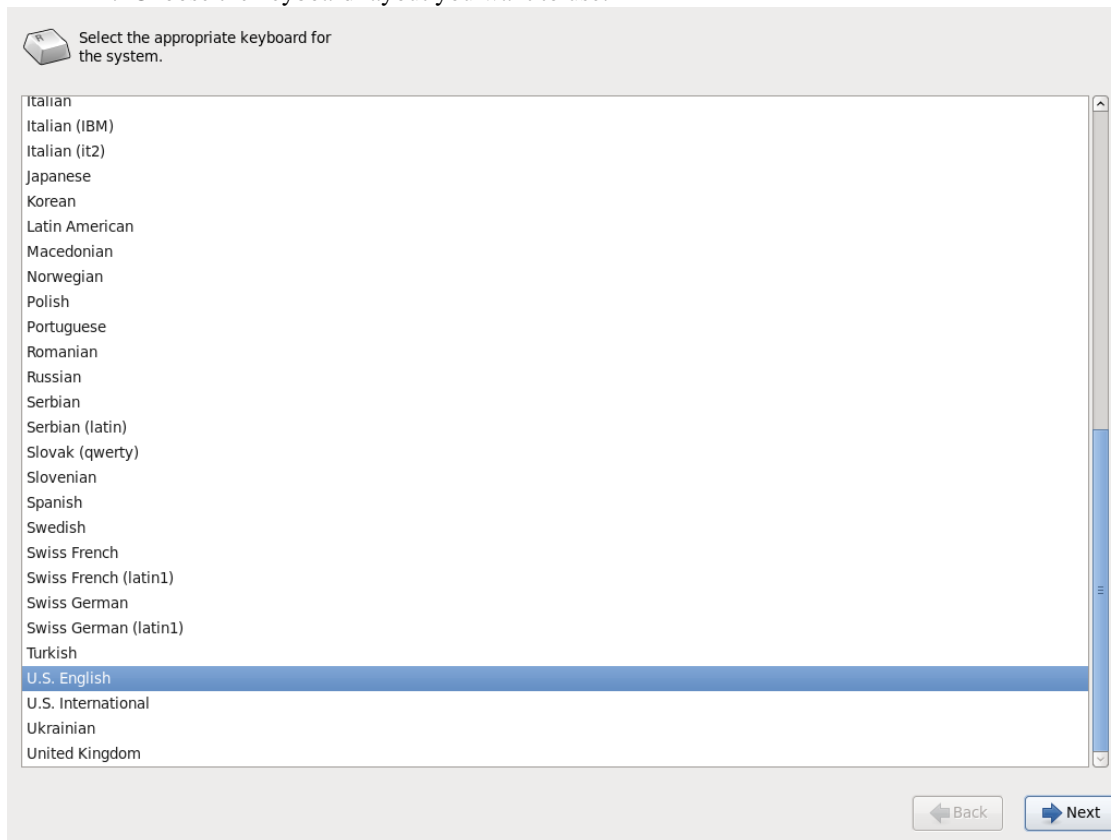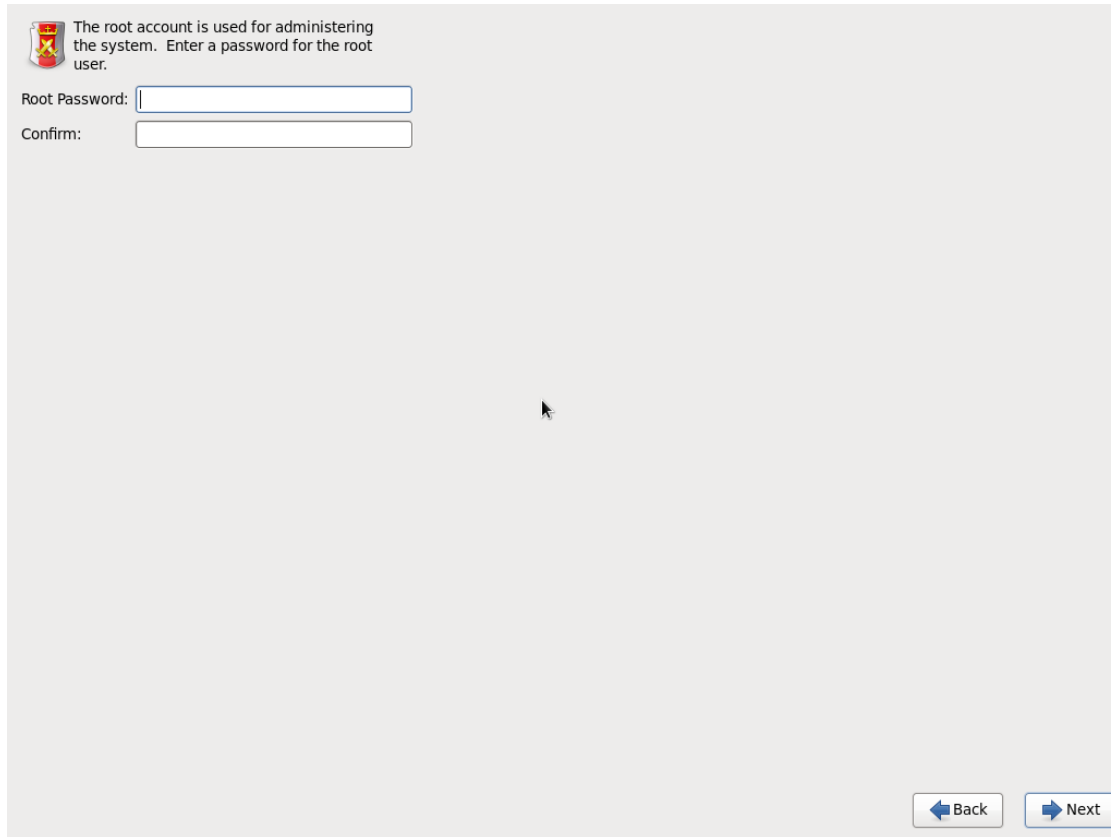
5. Set the timezone.
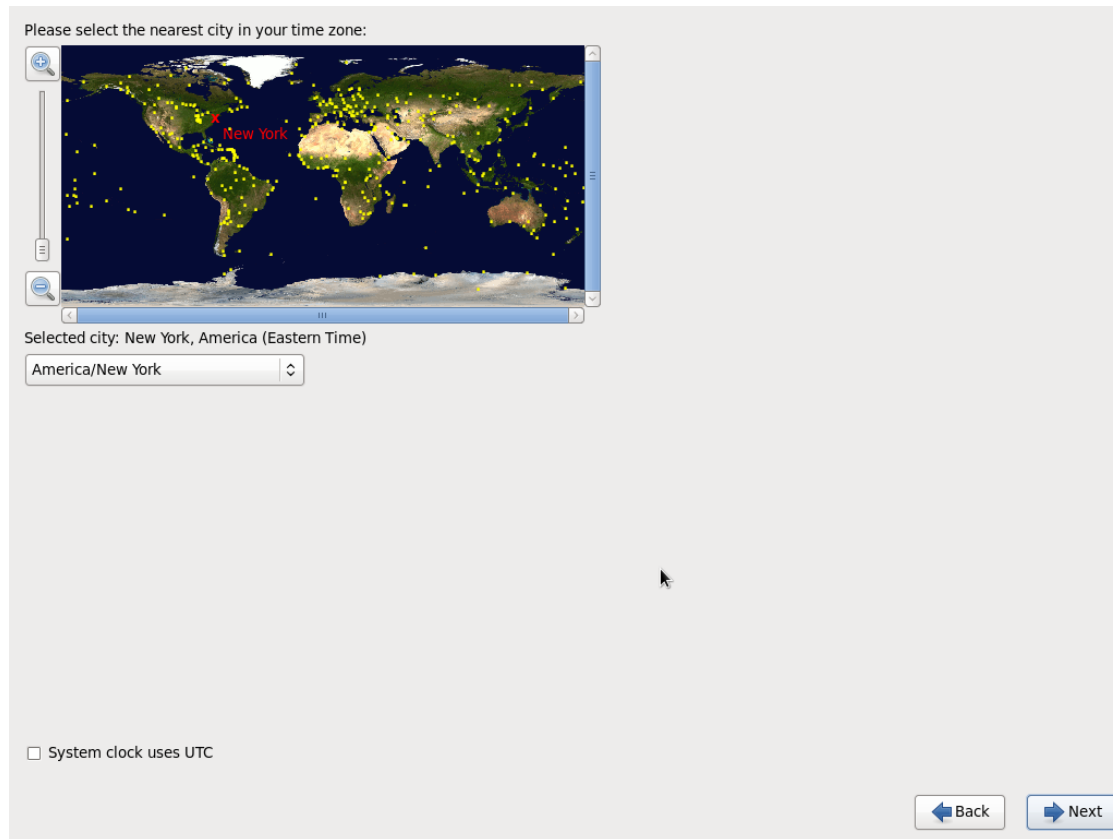
6. Set the root linux password.

7. Wait for the install to finish, then reboot the system and detach the ISO from the VM or remove the DVD from the disk drive.
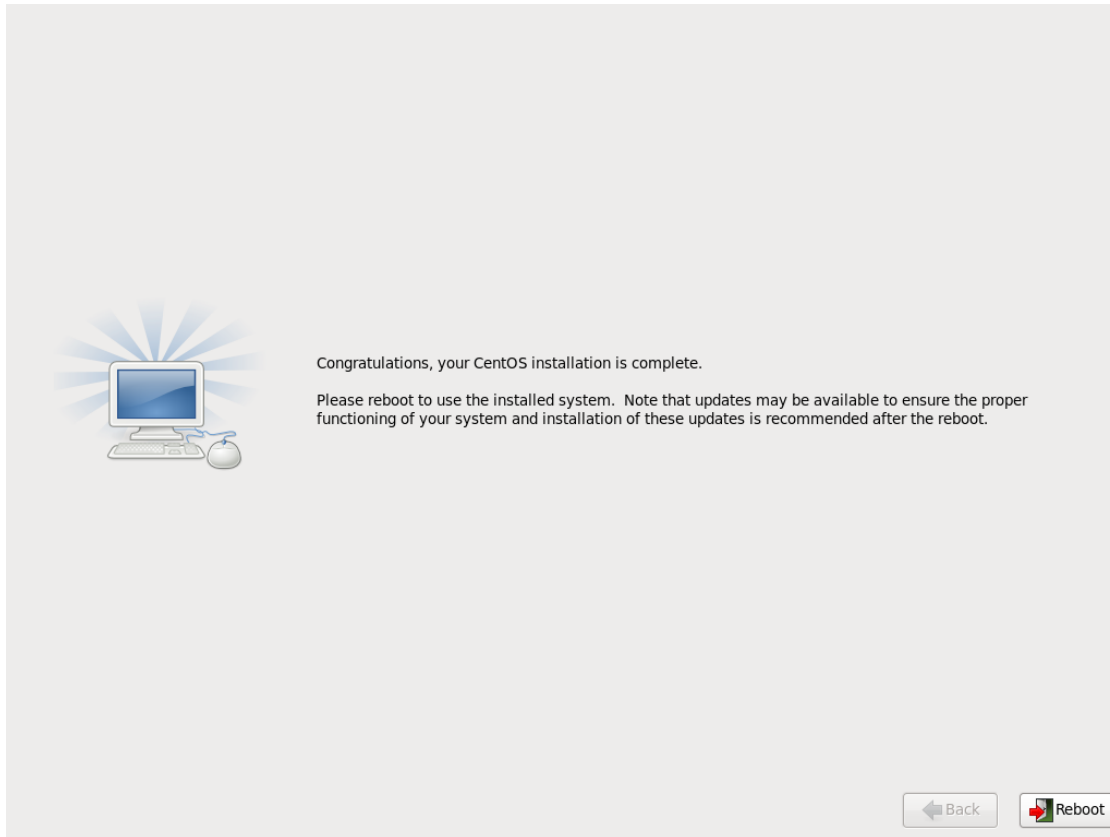
8. Once the system reboots, accepts the EULA license agreement, select the keyboard and timezone.

For a multi-node installation, repeat these steps for every physical or VM instance you wish to install. The installation phase is complete, you are now ready to configure the UForge AppCenter, see *Configuring UForge*.

Configuring UForge

Once the installation is complete on all the nodes you wish to use for the UForge AppCenter, you are now ready to configure all the UForge AppCenter services. This is done via the UForge Deployment Wizard that helps guide you through the final steps of the installation process.

Note: The LVM VolGroup Name name should be unique. The default when installing UForge will have a format similar to `vg_uss_150910-lv_uss_150910`. If your scanned instance has the same volume group name, or if you set up advanced partitioning with the same name, you will get an error when migrating.

Launching Deployment Wizard

To launch the UForge Deployment Wizard, use your browser and go to one of the nodes that have been installed:

http://<ip address of the node>:9998/deployments

Note: The deployment wizard is still using Flex technology, therefore you will require to have flash player installed on your browser. Steps are being taken to remove this dependency in future versions of the product.

Fill in the wizard, note that all the fields are mandatory.

1. Enter the Organization name. This is the name of the default Organization. The organization groups all the operating systems, formats and users for the platform.

2. Enter an email address. This is the email address of the root administrator of UForge. All administration email notifications are sent to this email address.

3. Set the root administrator's password. The password must be at least five characters long, including alphanumeric characters and the following special characters `!#$%&'()*+,-/:;<=>?@[\]/^_{|}~`. Spaces are not allowed at the beginning or end.

4. Provide a username, email and password for an initial user account to be created. This user account will have administration access rights for the default Organization.

5. Provide the UForge activation key and credential information to be able to receive UForge updates. If you do not have this information, please contact your vendor.

6. Select the operating systems and formats you would like to install. When you select an OS, a default mirror location is indicated. This is used for synchronization of packages and distributions. You can either accept the default values or modify the value to synchronize with another mirror.

7. Set the internet connection. By default, UForge expects to have a direct connection to the internet. If you de-select this option, you will need to enter the proxy hostname and port.

8. Set the SMTP Server to use for sending email notification messages created by UForge. If you want to use a SMTP relay, then also indicate the relay hostname and port number.

9. Click `next` to continue.

10. Enter the Web Server IP address, external hostname and the database IP address.

Note: The external hostname is used to construct two external URL endpoints. The first as the external URL endpoint of the user interface, and the other for REST API calls and command-line usage. The external hostname should normally be a fully qualified hostname. For example, if the external hostname is uf.example.com, then the following URL endpoints are created:

- User interface URL endpoint: https://uf.example.com/uforge
- Command-line URL endpoint: https://uf.example.com/apis

These external URL endpoints can be changed after the initial configuration is complete, refer to *Modifying the UForge Platform External URL Endpoints* for more information.

---

11. Select if OS and image storage should be local or remote. OS storage will be used for distributions, which image storage will include user data such as images, projects, mysoftware and other user data created with UForgeNow.

    - If you choose to use a remote storage, indicate the NFS server with mount point and you should enter the full path for the OS directory e.g. `/DISTROS`.

    - If you select remote storage for the image store, you have to make sure that the path `USER_DATA` exists, with the correct permissions i.e. tomcat:tomcat. Refer to *UForge Repository on Shared Storage*, step 4, Check mount points, for more details.

12. Indicate the generation cluster compute node hostnames. You can add additional compute nodes by clicking the add button.

13. Once you have finished the configuration, click the `Deploy` button.

Note: The deployment may take a few hours, depending on the number of operating systems you have chosen.

---

14. Once the deployment is complete, click `Finish`.

Warning: As part of the deployment phase, the wizard logs all the steps of the configuration. These logs though include the administration passwords and other sensitive data you have used to configure this platform. It is important that these logs do not remain on the machine once the deployment is over. To do this:

```
$ cd /var/log/UShareSoft
$ /bin/rm -rf oas-deploy
```

Configuring Ports

The following ports need to be configured for your UForge AppCenter.

For outgoing:

- 20 and 21
- 22 for SSH
- 443 and 80

For incoming:

- 80 and 443
- 22

In addition, communication ports between UForge and the cloud platform to which you will publish the images have to be open and depend entirely on your cloud platform configuration, see *Cloud Platform Default Ports* for more information.

Configuring NTP

Some cloud platforms will reject uploading machine images, if the HTTP request date is in the future of the target cloud platform. To ensure proper function of UForge, please edit 'server' directives in `/etc/ntp.conf` if UForge servers cannot connect to NTP servers on the internet.

Warning: If you wish to contact NTP servers on the internet, then port 123 (UDP) should be opened on your firewall.

Cloud Platform Default Ports

To allow UForge to register machine images, you must ensure that there is network connectivity to the cloud platforms you wish to push the machine images to. The table below provides the default port numbers of the cloud platforms supported, and indicates whether UForge uploads the machine image or requests the cloud platform to download.

Warning: For private cloud platforms, the port number may not be the default port number indicated.

Cloud Platform	Protocol	Default Port Number	Method of Registration
Abiquo	TCP	80 / 443	DOWNLOAD
AWS	TCP	443	UPLOAD
Azure	TCP	443	UPLOAD
CloudStack	TCP	8080	DOWNLOAD
Flexiant	TCP	443 and 4442	DOWNLOAD
Eucalyptus	TCP	8773	UPLOAD
Google Compute Engine	TCP	443	UPLOAD
Nimbula	TCP	80 / 443	UPLOAD
OpenStack	TCP	9292 and 5000	UPLOAD
vCloud Director	TCP	80 / 443	UPLOAD
vCenter	TCP	80 / 443	UPLOAD

Testing the Deployment

Once the configuration phase is complete, you may wish to carry out some basic sanity tests to ensure that the UForge AppCenter is running normally:

Step 1: Check if the web service is operational

Use the values in the uforge.conf to contact the web service and expect a `200 OK` response.

Get the values from the uforge.conf and add them to some environment variables (you could also just manually view the uforge.conf)

```
$ eval `grep '^UFORGE_WEBSVC_\|^UFORGE_GF_INTERNAL_IP\|^UFORGE_GF_HTTP_PORT\|^UFORGE_GF_WEBSVC_ROOT_C
```

Run a simple http request (using basic authentication) using curl

```
$ curl http://$UFORGE_GF_INTERNAL_IP:$UFORGE_GF_HTTP_PORT/$UFORGE_GF_WEBSVC_ROOT_CONTEXT/users/$UFORG
--verbose

* About to connect() to 10.0.0.240 port 9090 (#0)
*   Trying 10.0.0.240... connected
* Connected to 10.0.0.240 (10.0.0.240) port 9090 (#0)
> GET /ufws-3.0/users/root HTTP/1.1
> User-Agent: curl/7.20.1 (x86_64-redhat-linux-gnu) libcurl/7.20.1 NSS/3.12.8.0 zlib/1.2.3 libidn/1.1
> Host: 10.0.0.240:9090
> Accept: */*
> Authorization:Basic root:welcome
>
< HTTP/1.1 200 OK
< X-Powered-By: Servlet/3.0 JSP/2.2 (GlassFish Server Open Source Edition 3.1.1 Java/Sun Microsystems
< Server: GlassFish Server Open Source Edition 3.1.1
< Last-Modified: Thu, 03 May 2012 08:32:18 GMT
< ETag: "ef286bf07b8d18928287e12cb122ccf2"
< Content-Type: application/xml
< Content-Length: 5477
< Date: Thu, 03 May 2012 08:33:11 GMT
...<rest of the body removed>
```

Step 2: Check to see if the database is running

The database service should be running and available on the port `3306` and that the database table is present. The Percona Server instance should have `usharedb` and `oar`

```
$ service mysql status
MySQL running (22661)                                         [  OK  ]
```

Get the values from the auth.conf and add them to some environment variables (you could also just manually view the auth.conf)

```
$ eval `grep '^UFORGE_DB' /etc/UShareSoft/auth.conf`
$ echo "show databases" | mysql -f -N -u $UFORGE_DB_ADMIN_LOGIN -p$UFORGE_DB_ADMIN_PASSWORD -h db
information_schema
mysql
oar
performance_schema
usharedb
```

Step 3: Check the generation cluster resources

Check that all the cluster resources are available ('alive'). On each compute node

```
$ oarnodes | awk '/resource_id/ {n=$NF} /state : Suspected/ {printf "oarnodesetting -s Alive -r %s\n'
```

This should return without any output.

Further AppCenter Configuration

This section assumes that you have completed the installation of your UForge platform. Once it is installed and configured, you can:

Modifying the UForge Platform External URL Endpoints

There are three external URL endpoints for the UForge platform, namely:

- URL endpoint to access the UForge Portal (user interface)

- URL endpoint to access directly the REST web service for command-line tools and REST API calls

- URL endpoint for cloud platforms to download machine images from UForge. This URL endpoint is not used by end users, but only by cloud platforms that request to download machine images, rather than UForge uploading those machine images
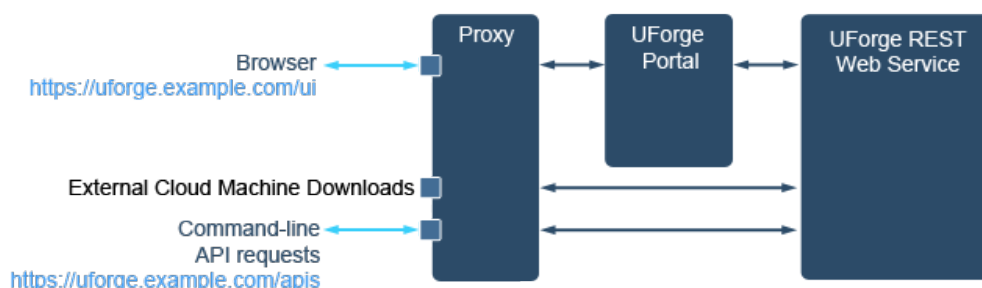
These URL endpoints are automatically created based on the external hostname provided during the initial configuration of the UForge platform (see *Configuring UForge*). These URL endpoints can be changed by updating certain variables in the `/etc/UShareSoft/uforge/uforge.conf` file.

The UForge Portal URL endpoint is constructed using the following variables:

https://<UFORGE_PROXY_INFOS>/<UFORGE_UI_ROOT_CONTEXT>

The URL endpoint for direct REST web service access is constructed using the following variables:

https://<UFORGE_PROXY_INFOS>/<UFORGE_API_ROOT_CONTEXT>

The download URL endpoint is constructed using the `UFORGE_IAAS_DOWNLOAD_URL` variable.

If you wish to use `http` rather than `https` (not recommended) then you require to set the following variable in the uforge.conf file:

```
UFORGE_PROXY_USE_SSL = false
```

For example, if you set the following variables in `uforge.conf`, will result in the following external URLs:

```
UFORGE_PROXY_INFOS = hq.example.com:5666
UFORGE_UI_ROOT_CONTEXT = /ui
UFORGE_API_ROOT_CONTEXT = /apis
UFORGE_IAAS_DOWNLOAD_URL = http://hq.example.com:5777/downloads
UFORGE_PROXY_USE_SSL = true
```

Resulting external URLs:

```
* UForge Portal: https://hq.example.com:5666/ui
* REST URL endpoint: https://hq.example.com:5666/apis
* Machine Image downloads (for external cloud platforms): http://hq.example.com:5777/downloads
```

To update the external URLs:

1. Update the `/etc/UShareSoft/uforge/uforge.conf` file for each node with the updated variables you wish.

2. Launch the following two scripts (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Adding a Compute Node

You can add a new OAR compute node which was instantiated from UForge but which was not configured as part of the initial deployment as follows.

1. Make a snapshot of the UForge Server (to be able to come back to the state without the additional OAR compute node).

2. Initial setup: `oar-server` and `oarnode1` to `oarnodeN` already configured. Name oarnodeX and IP-oarnodeX respectively the name and IP address of the new node to be added to the UForge "cluster".

Note: The following commands are run on the first existing oarnode, for example oarnode1 until stated otherwise.

---

3. Copy over `/etc/hosts` from oarnode1 to oarnodeX with mods

4. Get first local network interface (eth0)

```
ITF=`cat /proc/net/dev | grep : | cut -d ':' -f 1|grep -v lo | tr -d ' '| head -1`
```

5. Get IP address associated with this interface

```
IP=`/sbin/ip -o addr show $ITF 2>/dev/null| grep 'inet ' | awk '{print \$4}' |
sed -e 's?/.*??'`
```

6. Modify and copy /etc/hosts

```
sed -e 's/\<oarnode1\>/oarnodeX/g' /etc/hosts | awk -v ip=$IP -v name=oarnode1
'{print}END{printf "%s %s\n",ip,name}' | ssh IP-oarnodeX dd of=/etc/hosts
```

---

7. Copy over uforge.conf file and others

```
rsync -a /etc/UShareSoft/uforge/uforge.conf IP-oarnodeX:/etc/UShareSoft/uforge/
rsync /etc/oar/oar.conf IP-oarnodeX:/etc/oar/
rsync /etc/ssh/sshd_config IP-oarnodeX:/etc/ssh
rsync -a ~oar/.ssh IP-oarnodeX:~oar
```

8. Run the following commands on all oarnode1 .. oarnodeN and oar-server but **not** oarnodeX. This adds new node in all machines /etc/hosts

```
awk 'BEGIN{d=1}/\<oarnodeX\>/ {d=0}{print}END{if(d==1){print "IP-oarnodeX
oarnodeX"}}' /etc/hosts > /tmp/hosts.NEW
diff -q /tmp/hosts.NEW /etc/hosts >/dev/null
if [ $? -ne 0 ]; then
rsync -a /etc/hosts /etc/hosts.SVG-`date +"%Y-%m-%d"`
cp /tmp/hosts.NEW /etc/hosts
rm -f /tmp/hosts.NEW
fi
```

9. Run the following commands on oarnodeX

```
chmod 666 /etc/oar/oar.conf
/opt/UShareSoft/uforge/conf/oar_user_setup.sh
cp /opt/UShareSoft/uforge/tmpl/nfs.tmpl /etc/sysconfig/nfs
cp /opt/UShareSoft/uforge/tmpl/mountisos_init /etc/init.d/mountisos; chkconfig
--add mountisos
for s in ntpd tomcat mysql httpd oar-server openstack-glance-api
openstack-glance-registry oas oas-deploy; do service $s stop; chkconfig --levels
0123456 $s off ; done >/dev/null 2>&1
ntpdate pool.ntp.org ; service ntpd start
for s in oar-node ntpd postfix mountisos; do chkconfig --levels 2345 $s on; done
service ntpd stop ; ntpdate pool.ntp.org ; chkconfig ntpd on ; service ntpd
start
service mountisos start
service oar-node start
service sshd restart
/opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh -f >/dev/null 2>&1
```

10. Run the following commands on oar-server to create new resources on oarnodeX from existing oarnode1 resources

```
/usr/bin/oarnodes | /bin/awk '/network_address=oarnode1/
{s=$0;gsub(".*nature=","",s);gsub(",.*","",s);printf "/usr/sbin/oarnodesetting
-a -h oarnode3 -p cpuset=0,nature=%s\n",s}' | sh
```

You can also use a remote disk space of the compute node to generate multiple machine images in parallel by mounting the /space directory with a NAS or SAN.

Removing a Node

In order to remove a node, run the following command on the UForge server:

```
/usr/bin/oarnodes | /bin/awk "/resource_id/ {n=\$NF} /network_address=$
{REMOVENODE}/ {printf \"/usr/sbin/oarnodesetting -s Dead -r %s ; sleep 2;
/usr/sbin/oarremoveresource %s\n\",n,n}" | sh
```

Configure Apache and Tomcat Web Services to use SSL Certificate

It is highly recommended that all communication with UForge is done via HTTPS. After the initial installation of UForge, neither the HTTP server (Apache) nor the application server (Tomcat) have yet been configured to use a SSL certificate and allow HTTPS.

To configure both servers to use an SSL certificate:

1. Log in as root to the machine running the UForge Apache and Tomcat Web Services.

2. Copy the SSL certificate files locally to the machine. Note that you should have three or four files, for example:

    - SSLCertificateFile: server.crt.pem

    - SSLCertificateKeyFile: server.key.pem

    - SSLCACertificateFile: CA.crt.pem

    - SSLCertificateChainFile: intermediate.CA.crt.pem (this one is optional)

   You need to build a self contained certificate as follows:

   ```
   $ cat server.crt.pem CA.crt.pem > server_CA_chain.crt.pem
   ```

   or:

   ```
   $ cat server.crt.pem intermediate.CA.crt.pem CA.crt.pem > server_CA_chain.crt.pem
   ```

   Note that `.pem` files contain the following type of data for certificate files:

   ```
   $ cat server.crt.pem
   -----BEGIN CERTIFICATE-----
   MIIHJTCCBg2gAwIBAgIDB25YMA0GCSqGSIb3DQEBBQUAMIGMMQswCQYDVQQGEwJJ
   ...
   aW9uIEF1dGhvcml0eTADAgECGmRMaWFiaWxpdHkgYW5kIHdhcnJhbnRpZXMgYXJl
   V4XfZvZtrRcZ
   -----END CERTIFICATE-----
   ```

   And the following data for the key file:

   ```
   $ cat server.key.pem
   -----BEGIN PRIVATE KEY-----
   MIIEvwIBADANBgkqhkiG9w0BAQEFAASCBKkwggSlAgEAAoIBAQDaRIAE7wrKbS9T
   ...
   GdIr+qaNjk+eZLVsuPsAvwPlsWI/Cip7Zqygtvvviteyen0VZbLpRJgbbjXqh9GwP
   G33VnWF89pfm5FNRu3WHIf8Ukw==
   -----END PRIVATE KEY-----
   ```

   If this is not the case, refer to the OpenSSL documentation on how to convert certificate and key files from one format to another.

3. Put the following entries in `/etc/httpd/conf.d/ssl.conf`:

    - SSLCertificateFile `/etc/pki/tls/certs/server.crt.pem`

    - SSLCertificateKeyFile `/etc/pki/tls/private/server.key.pem`

    - SSLCertificateChainFile `/etc/pki/tls/certs/intermediate.CA.crt.pem`

    - SSLCACertificateFile `/etc/pki/tls/certs/CA.crt.pem`

4. Verify the permissions and ownerships of these files

```
$ ll -d /etc/pki/tls/certs/server.crt.pem /etc/pki/tls/private/localhost.key /etc/pki/tl
drwxr-xr-x. 2 root root 4096 Sep 25 12:05 /etc/pki/tls/certs/
-rw-------. 1 root root 1188 Sep 25 12:05 /etc/pki/tls/certs/server.crt.pem
drwxr-xr-x. 2 root root 4096 Sep 25 12:05 /etc/pki/tls/private/
-rw-------. 1 root root  887 Sep 25 12:05 /etc/pki/tls/private/server.key.pem
```

5. (Re)start the httpd server:

```
$ service httpd restart
```

If the server does not start, this may be because of a bad certificate, key or CA certificate file. In this case, check the appropriate logs in `/var/log/httpd`.

6. Verify the validity of the certificates:

```
$ openssl s_client -connect localhost:443
...
Verify return code: 0 (ok)
---
Ctrl-C or Ctrl-D to leave openssl client
```

If there is a problem with the certificate you might get outputs like:

```
$ openssl s_client -connect localhost:443
...
Verify return code: 18 (self signed certificate)
---
```

or

```
$ openssl s_client -connect localhost:443
...
Verify return code: 21 (unable to verify the first certificate)
---
```

7. Verify the certificate:

```
$ openssl s_client -showcerts -connect <ip-of-the-uforge-web-service-machine>:<port>
```

Or you can also use same openssl client command used for the Apache server in step 6.

To verify that the new certificate is correct and if the Tomcat service is accessible from the outside, go to http://www.digicert.com/help/ and type the public name or IP address of your web service.

Note that there is no way to specify another port than HTTPS (443) on this page therefore you might need to add an iptables redirection rule like:

```
$ iptables -t nat -A PREROUTING -i eth0 -p tcp --dport 443 -j REDIRECT--to-port 9191
```

Configuring UForge Behind Enterprise Proxy

Once your UForge platform deployment is complete you can configure SMTP proxy. To configure SMTP:

1. In `/etc/UShareSoft/uforge/uforge.conf` you can modify:

- UFORGE_RELAY_HOST=
- UFORGE_RELAY_PORT=
- UFORGE_RELAY_USER=
- UFORGE_RELAY_PASSWORD=

These can be empty.

2. Run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

Configuring the Web Service

Each UForge Web Service instance runs inside a Tomcat application server. The web service has the following basic configuration information that is stored in a central configuration file: `uforge.conf`. The main configuration attributes for the web service are:

- External hostname used for incoming user connections
- External hostname for downloading images (this uses port `80` allowing all cloud platforms to be able to communicate with UForge for publishing images)
- Internal IP address used to connect with the other UForge services
- HTTP port (default: `8080`)
- HTTPS port (default: `8443`)
- Administration console port (default: `4848`)
- Administration console credential information (user and password)
- Root context of the web service (for example `/uforge`)

When installing UForge via the deployment wizard some of the configuration attributes can be decided by the administrator. The deployment wizard also creates the uforge.conf file with all the configuration information.

To view the uforge.conf file:

1. Log in to the web service node as root:

```
$ ssh root@<ip address of the node>
```

2. Open the uforge.conf file:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
```

3. After making appropriate changes in these files, you should run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

For more information on Tomcat, see http://tomcat.apache.org

Configuring the Database

UForge uses the MariaDB database to store all the UForge meta-data and user information. The web service communicates with the database using hibernate. When installing UForge using the deployment wizard, one database instance is configured.

Note: By default no mechanism is configured to backup the contents of the UForge database. MariaDB can be configured as a cluster or in master-slave mode to provide reliability and to have a replicate of the data. Regular backups of the database should also be done.

The MariaDB database has the following basic configuration information that is stored in a central configuration file: `uforge.conf`. The main configuration attributes for the database are:

- Administration credential information (user and password)

- Address of the uforge database host.

When installing UForge via the deployment wizard some of the configuration attributes can be decided by the administrator. The deployment wizard also creates the `uforge.conf` file with all the configuration information.

If you decide to change database configuration information including the password, then you must also update the `auth.conf` and `uforge.conf` files with the correct information on all the nodes of the platform.

To view the `uforge.conf` or `auth.conf` files:

1. Log in to the web service node as root:

```
$ ssh root@<ip address of the node>
```

2. Open the uforge.conf file or auth.conf files:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
$ vi /etc/UShareSoft/auth.conf
```

3. After making appropriate changes in these files, you should run the following command on all the nodes (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

For more information on MariaDB, see http://www.mariadb.com

Configuring the Scheduler

The scheduler can be configured to throttle the maximum number of:

- generations allowed in parallel per compute node.

- publishes allowed in parallel per compute node. A `publish` is when a user wishes to upload and register a generated image to a particular cloud environment.

- shares allowed in parallel per compute node. A `share` is when a user requests to share a template that they have in their private workspace.

- live system scans allowed in parallel per compute node. This is used for migrating live systems from one environment to another.

By default, after installing UForge using the deployment wizard, UForge will have X compute nodes. The scheduler is configured to allow 1 generation that has been chosen during the deployment on each compute node.

You can re-configure the scheduler to have different scheduling policies for different job types for each compute node. This is done be declaring a resource with a `nature` in OAR. Each `nature` corresponds to a specific job type. For example by declaring 4 resources with the nature ID `0` for compute node `node1` will configure the scheduler to allow up to 4 parallel generations. The following table shows the mapping between the different job types and the nature ID.

Job Type	Nature ID
Image Generation Job Type	0
Publish Image Job Type	1
Share Job Type	2
System Scan Job Type	3
Update cache repo (cron job)	4
Update_repos_pkgs.sh (Spider) (cron job)	5

Viewing Current Resources

To view all the current resources in the scheduler, log in to the oar scheduler node as root and run the command `oarnodes`:

```
$ ssh root@<ip address of the node>
$ oarnodes
network_address : compute1.example.com
resource_id : 1
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=0, network_address=iso

network_address : compute1.example.com
resource_id : 4
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=1, network_address=vm,
```

This provides the basic information of each resource including:

- `network_address`: the compute node this resource is attached to

- `resource_id` is the ID of the node

- `state` is the current state of the resource

- `properties`: the main properties of the resource, including the nature ID that determines the job type this resource is providing

Adding or Updating a Resource

To add or update a resource, first log in to the oar scheduler node as `root`.

To add a resource to compute node `node1` allowing to generate an image:

```
$ oarnodesetting -a -h node1 -p cpuset=0,nature=0;
$ oarnodes
network_address : node1
resource_id : 92
state : Alive
properties : deploy=NO, besteffort=YES, cpuset=0, desktop_computing=NO, nature=0, network_address=vm,
```

To change a current resource (resource_id: 92) to a different job type (for example publish images):

```
$ oarnodesetting -h node1 -r 92 -p nature=1;
```

Removing a Resource

To add or update a resource, first log in to the oar scheduler node as `root`.

To remove a resource from the compute node, run the following commands:

```
$ oarnodesetting -s Dead -r <resource_id>
$ oarremoveresource <resource_id>
```

Deleting a Job

In case of a problem, you may want to delete a job which is stuck in a waiting state.

In this case, run:

```
$ oardel <job_id>
```

Managing the Watchdog Services

UForge has a set of watchdog services that carry out housekeeping tasks on a regular basis. These are:

- Cleanup Tickets: Task to remove generated images that are no longer attached to a parent image ticket in the database. This happens when the user "deletes" the generated image from their account. By default this is run once a day at 04:10AM.

- Update Distribution Packages: Task to regularly search the operating system repositories for any new updates and synchronize metadata into the UForge database. By default this is run every hour.

- Reset OAR Resources: Check the OAR scheduler resources and ensure that their state is "Active". By default this is run every 5 minutes.

Each of these housekeeping tasks are registered as a cron job in the first database node of the UForge platform to schedule the task to be run periodically. The frequency of these tasks can be changed.

To change the frequency of these housekeeping tasks, you need to update the crontab. Each line of a crontab file represents a job and is composed of a CRON expression, followed by a shell command to execute. The syntax is:

`dw month day hr min` followed by the command to be executed

Where:

- `dw` is the day of the week (0 - 6) (0 is Sunday, or use names)

- `month` is 1 - 12

- `day` is day of the month (1 - 31)

- `hr` is the hour (0 - 23)

- `min` is minutes (0 - 59)

To view these cron jobs, log in to the oar scheduler node as root and view the cron jobs:

```
$ crontab -l
*/5 * * * * /opt/UShareSoft/uforge/cron/reset_oar_resources.sh
10 2 * * * /opt/UShareSoft/uforge/cron/cleanup_tickets.sh
10 3 * * * /opt/UShareSoft/uforge/cron/cleanup_scans.sh
42 * * * * /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
05 * * * * /opt/UShareSoft/uforge/cron/update_repos_local_cache.sh
1 8 * * * /opt/UShareSoft/uforge/cron/drop_caches.sh
```

To update the crontab, log in to the oar scheduler node as root and edit the crontab

```
$ crontab -e
```

Cron Job Guidelines

There is no specific order to be respected when running cron jobs. These jobs are not inter-dependent.

```
*/5 * * * * /opt/UShareSoft/uforge/cron/reset_oar_resources.sh
```

This needs to be launched on a regular basis to avoid having issues with the OAR scheduler computation nodes. In fact, it happens, following network issue or other, that nodes move to state "Suspected". This job tries to fix that.

This job executes very quickly and does not take resources on the machine. It is set by default to 5 minutes but this can be changed.

```
10 2 * * * /opt/UShareSoft/uforge/cron/cleanup_tickets.sh
```

When a user deletes a machine image persisted on the NAS, only the metadata is removed from the database to avoid using a webservice thread to delete the file. This could take time and should be done asynchronously.

This job goes through the NAS and the database and checks which directories could be removed.

This job could potentially take a long time and be IO-intensive. It is highly recommend to execute it when there is not a lot of activity on the platform. We have set this at 2:10AM because there is not a lot of activity on our platform at that time of day.

This script could be launched several times in a day depending on the size of the infrastructure. For example, if the NAS is not so big and if there are a lot of images created and deleted per day, it might be a good to launch it several times a day.

```
10 3 * * * /opt/UShareSoft/uforge/cron/cleanup_scans.sh
```

Same as for /opt/UShareSoft/uforge/cron/cleanup_tickets.sh

```
42 * * * * /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

This mechanism launches UForge Spider to crawl the packages from the registered repositories.

It is important to keep these repositories up to date so that:

- when a user creates a new template, the latest package updates are listed

- when a user checks the appliance library, the number of updates available are listed

- the image generations is faster. If this is not done on a regular basis, when launching a generation, the repositories of the template distribution will be updated and the user will have to wait longer.

Also, if a repository the platform is connected to deletes packages (e.g. because of newer package version – this is not the case of UShareSoft official repositories), having the latest packages available is important.

```
05 * * * * /opt/UShareSoft/uforge/cron/update_repos_local_cache.sh
```

Since UForge 3.5.1, the UForge platform does not download all the packages from all the repositories listed. Instead, only the necessary packages are downloaded "on demand".

In the case of repositories that remove packages (because of newer packages), it is important to be able to reproduce a machine image with the same packages even though these packages no longer exist on the remote repository.

The local partial copy of the repository is registered in the platform as another repository. Each time new packages are downloaded for a repository, the local directory is marked "to be refreshed". When `update_repos_local_cache.sh` is executed, it checks all the local repo directories tagged as "to be refreshed" and executes the tool to update the native distribution repository (for rhel : createrepo with options).

It is important to execute regularly if the repositories use removed packages.

On extremely large platforms, it could take time and be IO-intensive.

If this command fails, usually it will only have an impact several days later (depending on the removing-package-repo policy with package removal). For example, if you generate a machine image with a sticky package version 1.2.3 on NTP. Let's consider NTP is on a repository that removes packages. You generate a machine image and NTP gets downloaded. update_repos_local_cache.sh. You generate the machine image again. No issue. Three days later, 1.2.4 version is released and 1.2.3 is removed from the remote repository. In that case, you will no longer be able to generate as the package in not in the remote repository, nor in the cache.

```
1 8 * * * /opt/UShareSoft/uforge/cron/drop_caches.sh
```

This calls native Linux commands to free up some memory on the platform.

If this commands fails it means the platform (not UForge but the machine itself) is in bad shape. It has no direct consequence on the UForge platform (only side effect: usually, issue with memory).

Tuning the Services

You can set the priority of tasks on the UForge platform in the file `services_conf.json`.

You can run COMMAND with an adjusted value for the nice value, which affects process scheduling. This allows you to fine tune the order in which services are treated by the platform and allows you to improve performances.

Modifying a Configuration Profile

UForge includes a configuration file written in json that includes a set of profiles. These profiles contain a set of parameters to deal with resource priority (CPU, disk) and set different properties.

The configuration file is located at `/etc/UShareSoft/uforge/services_conf.json`

For each profile, you can modify the following parameters:

- `ionice`: set io priority of a process
- `nice`: set cpu priority of a process
- `sudo`: give sudo access or not to a process
- `jvm`: set java system properties and options

The following is a basic example of a configuration profile, named `profile1`. The parameters defined in `profile1` will be applied to the process specified by exec.

```
"profile1": {
  "exec": "/opt/UShareSoft/uforge/tool/demo.sh",
  "ionice": {
  "class": 2,
  "level": 7
},
"jvm": {
```

```
  "options": [
    "-Xmx2048m",
    "-Xms768m",
    "-XX:PermSize=256m",
    "-XX:MaxPermSize=1024m"
  ],
  "sys-properties": {
    "jna.library.path": "/opt/UShareSoft/uforge/lib"
  }
},
  "nice": {
    "niceness": 15
  }
}
```

Launching a Configuration Profile

To launch the process in a specific profile, run the script `runjob.py` located in /opt/UShareSoft/uforge/bin/

This sets all the properties you have defined in the profile and forwards all the arguments.

Command usage:

```
$ /opt/UShareSoft/uforge/bin/runjob.py profile_name arg1 arg2
```

where:

- `profile_name` is the name of the profile

- `arg1...argn` is the list of arguments to forward to the process this profile is to be applied to (defined in "exec" element)

Including an Override

You can include a profile that will override a process in a profile.

The following is an example of a profile that includes an override profile called `low_prio`

```
"my_process": {
  "exec": "/opt/UShareSoft/uforge/tools/demo.bin",
  "include": "low_prio",
  "sudo": true
  },
```

Using the Event Bus

UForge can be easily extended to interact and integrate with other products and services via an event bus service. Whenever a `POST`, `PUT` or `DELETE` request to UForge, a corresponding event is sent to the event bus (RabbitMQ). Custom plugins can be built to listen for these events and trigger custom business logic or call out to other 3rd party systems.

The event bus service is based on RabbitMQ. Custom plugins are known as "consumers".

Custom plugins (RabbitMQ consumers) can be written in many different languages including:

- Java

- Ruby

- C

- Python

- Erlang

- PHP

- Node.js etc

RabbitMQ is a message broker. In essence, it accepts messages from "producers" and delivers them to "consumers". In-between, it can route, buffer and persist messages, known as the "queue".

When a `POST`, `PUT` or `DELETE` request is sent to the web service, then UForge creates a message (the producer) and posts this to a routing service, called an exchange. This exchange is configured to publish messages to the UForge queue. When writing a custom plugin (consumer), it registers itself to this queue.

Administrators can also reconfigure the event bus routing service to adapt to more complex workflows. This is done in the event bus administration UI and use the root account of the UForge service, at:

```
http://youripaddress:15672
Login: uss-admin
Password: administration password
```

Refer to the RabbitMQ documentation for more information: https://www.rabbitmq.com/documentation.html

Writing a Custom Plugin (Consumer)

Prior to writing a custom plugin, it is important to understand the producer messages created by the UForge AppCenter. When a request is sent to the web service (`POST`, `PUT` or `DELETE`) the web service creates a producer message with the contents of the DTO (data transfer object) that is sent as a response to the request. The attributes for each DTO is described in the *APIs xsd file <apis:apis-index>*.

When creating a plugin, you will have access to all the attributes in a message. The plugin will contain the custom business logic required.

By default, this mechanism is used to send notification emails. For examples, refer to the RabbitMQ tutorials: https://www.rabbitmq.com/getstarted.html

Registering a Custom Plugin to Event Bus

To register a custom plugin, you must provide:

- The event bus URI, can be found in the UForge configuration file: `/etc/UShareSoft/uforge/uforge.conf`

- The name of the queue

This information can be found in the RabbitMQ administration console.

1. Login to the administration console:

```
http://youripaddress:15672
Login: uss-admin
Password: administration password
```

2. Go to the Queues Tab, you will see the available queues. The default queue is `distrotools`.

Email Notification Service

The UForge AppCenter provides an email notification service using RabbitMQ to notify the super-user administrator and users via email for certain types of information.

Email notifications are triggered for the administrator when:

- There is a server template generation failure.
- A new user has been created on the platform (or in the case where this is manual, a request for a new user account to be created).
- A user has reported an abusive comment that requires moderation.

Email notifications are triggered to the end users when:

- A new user account has been created using their email address. This includes information on their credential information to access UForge.
- For all the templates the user is "following", emails are sent when the template is updated or a new comment has been added to the template.

Changing Email Address for Notifications

During the initial install of the UForge AppCenter, you must set the email address for the administrator. If you need to change the administrator's email address then you must update the uforge.conf file for each node that comprises the UForge AppCenter. UForge allows you to have a different email address for:

- new registrations (user account creations) to UForge
- all other administrator emails (generation failures etc)

You can also provide a no-reply email address when an email is sent.

To change the email address:

1. Log in to the node as root and edit the `uforge.conf` file:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
```

2. Update the following variables:

    - `UFORGE_REGISTRATIONS_EMAIL`: to receive notifications on new user accounts being created
    - `UFORGE_POSTMASTER_EMAIL`: to receive all other email notifications (errors etc)

3. Run the script to force UForge to use the new `uforge.conf` file, this will restart certain UForge services (if multi-node the following order should be respected: compute notes, db nodes, web service nodes):

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

Customizing the Email Templates

You can modify or internationalize the information sent during an email notification. UForge provides a set of default templates for each email type sent by the system. The templates are stored in: `/opt/UShareSoft/uforge/tmpl`

> **Warning:** When UForge is upgraded all the templates in the default directory will be overwritten. To ensure that any custom templates are restored during an update, a copy must be made of the custom template.

User account management emails:

- `ForgotPasswordEmail.tmpl` – email sent to the user when resetting their account password
- `SubscriptionAdminEmail.tmpl` – email sent to the administrator when a new user account has been created
- `SubscriptionUserEmail.tmpl` – email sent to the user requesting a new UForge account

Error emails:

- `ImageGenerationErrorMessage.tmpl` – email sent to administrator when an image generation failure occurs
- `WebServiceErrorsMessage.tmpl` (not used today)

Marketplace notification emails:

> **Note:** These are only applicable if your Uforge AppCenter is connected to a Marketplace. Otherwise, disregard.

- `AppStoreNotificationNewComment.tmpl` – email sent to the user watching the template where a new comment is added
- `AppStoreNotificationTemplateNew.tmpl` – email sent to the user watching the template where a new version of the current template is available
- `AppStoreNotificationTemplateUpdate.tmpl` – email sent to the user watching the template where the template is updated by the publisher
- `AppStoreTemplateReportAbuse.tmpl` – email sent to the administrator when a user reports an abuse for a comment in a template
- `AppStoreTemplateReportAbuseConfirm.tmpl` – confirmation email sent to the user reporting an abuse

The following keywords are reserved for substituting information into the emails:

- `#USER#` - The user name of the person carrying out the request
- `#EMAIL#` - The email of the user carrying out the request
- `#PASSWD#` - The password of the new user account created or during a "forgotten password" request

Specific keywords for Marketplace (App Store) email notifications:

- `#COMMENT#` - The comment added to a template in the Marketplace

- `#REPORTER#` - The user name of the person carrying out the request (used for adding a comment or reporting an abuse)

- `#REPORTEREMAIL#` - The user name of the person carrying out the request (used for adding a comment or reporting an abuse)

- `#NBREPORTEDABUSE#` - The total number of times this comment has been flagged as abusive

- `#CMDACCEPTABUSE#` - The command to use to remove the comment from the Template in the Marketplace

- `#TEMPLATENAME#` - The name of the template in the Marketplace

- `#PUBLISHDATE#` - The date the template was published in the Marketplace

- `#TEMPLATEVERSION#` - The version of the template published in the Marketplace

- `#ORGNAME#` - The organization name where the Marketplace resides

- `#PUBLISHEREMAIL#` - The email address of the publisher (to the Marketplace)

UForge uses the directory /var/opt/UShareSoft for custom files. If this directory exists, as part of the upgrade process and custom files will be restored.

> **Warning:** When UForge is upgraded all the templates in the default directory will be overwritten. To ensure that any custom templates are restored during an update, a copy must be made of the custom template.

Therefore to change an email template:

1. Log in to the node as root then go to the template directory:

```
$ cd /opt/UShareSoft/uforge/tmpl
$ vi AppStoreNotificationNewComment.tmpl
```

2. Change the contents of the template and rename using the extension for the new language, if appropriate.

3. Copy the new template to all the other nodes of the UForge AppCenter.

4. Save a copy of the new template to protect against an upgrade overwriting the custom template:

```
$ mkdir -p /var/opt/UShareSoft/uforge/tmpl
$ cp /opt/UShareSoft/uforge/tmpl/AppStoreNotificationNewComment.tmpl /var/opt/UShareSoft/uforge/
```

5. Instantiate the following changes by running the following command:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

Customizing UForge Authentication for SSO

You need to build a custom AuthN/AuthZ module in order to perform Single Sign On to the UForge Platform.

The goal is to gain authenticated and authorized access to the RESTful UForge Webservice underlying methods. These methods are protected by a SecurityFilter and a SecurityContext using what is called a filter chain. The UForge Platform uses JSR 311(JAX-RS: The JavaTM API for RESTful Web Services), JSR 250 (Common Annotations for the JavaTM Platform) and Jersey (RESTful Web Services in Java).

To gain access through this filter chain to the UForge Webservice methods, you first need to be "Authenticated" (to prove who you are) and then be "Authorized", that is, be granted entitlements to perform or access certain methods of the UForge Webservice.

So, in order to connect to the UForge Webservice you first have to pass through the "authentication filter chain" of the web service. These filters are defined by default in web.xml

By default there are two filters:

- BasicAuthenticationFilter: the Basic method from RFC2617

- APIKeyAuthenticationFilter: the UForge internal APIKey authentication.

You should create an implementation of the interface IAuthenticationFilter returning an object implementing IUser-Authentication if authentication is ok, an exception if the authentication is invalid and null if your filter can not handle the request authentication.

The following is a basic example of an implementation of the RFC.

```java
package com.usharesoft.authentication;
import com.sun.jersey.api.core.HttpRequestContext;
import com.sun.jersey.core.util.Base64;
import com.sun.jersey.spi.container.ContainerRequest;
import com.usharesoft.common.messages.User;
import com.usharesoft.db.DbAccess;
import com.usharesoft.db.DBException;
import com.usharesoft.services.IDMService;
import com.usharesoft.services.ServicesContext;
import com.usharesoft.services.exceptions.UForgeException;
import org.apache.log4j.Logger;
import org.hibernate.Criteria;
import org.hibernate.criterion.Restrictions;
import java.net.URI;
import java.util.List;
import java.util.regex.Matcher;
import java.util.regex.Pattern;

public class BasicAuthenticationFilter implements IAuthenticationFilter {
    private static final Logger logger = Logger.getLogger(BasicAuthenticationFilter.class);
    private static final Pattern CREDENTIALS = Pattern.compile("^basic\\s+(?.*)$", Pattern.CASE_INSEN
    private static final Pattern USER_PASS = Pattern.compile("^(?[^:]*):(?.*)$", Pattern.CASE_INSENSI
    @Override
    public IUserAuthentication filter(ContainerRequest containerRequest) throws UForgeException {
        logger.trace("Getting Auth from Http Header");
        return getUserAuthentication(containerRequest);
    }
    protected IUserAuthentication getUserAuthentication(HttpRequestContext containerRequest) throws U
        /*
         * Get the authentication information from the HTTP header Return an
         * empty string if no authorization information is found
         */
        List authorizationList = containerRequest.getRequestHeaders().get("Authorization");
        if (authorizationList == null || authorizationList.isEmpty()) {
            return null;
        }
        /*
         * Get the first value as there should only be one value here as we will
         * only support Basic authentication for the moment Remove the Basic Tag
         */
        String credentials = authorizationList.get(0);
        logger.trace("Credentials: " + credentials);
        /*
         * Decode credentials
         */
```

```java
        Matcher matcher = CREDENTIALS.matcher(credentials);
        if (!matcher.matches()) {
            logger.debug("Not matched Authorization header: " + credentials);
            return null;
        }
        /*
         * Consider that we match
         */
        /*
         * Decode base64 credentials
         */
        String base64basicCredentials = matcher.group("credential");
        if (!Base64.isBase64(base64basicCredentials)) {
            logger.warn("Invalid Base64 basic-credentials: " + base64basicCredentials);
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.AUTHENTICATION.INVALID");
        }
        String basicCredentials = Base64.base64Decode(base64basicCredentials);
        /*
         * Decode user-pass
         */
        matcher = USER_PASS.matcher(basicCredentials);
        if (!matcher.matches()) {
            logger.warn("Invalid basic-credentials: " + basicCredentials);
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.AUTHENTICATION.INVALID");
        }
        /*
         * Compute fields
         */
        String userId = matcher.group("userId");
        String userName;
        String targetUserName = null;
        if (UserAuthentication.isCompositeUserTargetUser(userId)) {
            userName = UserAuthentication.getCompositeUser(userId);
            targetUserName = UserAuthentication.getCompositeTargetUser(userId);
        } else {
            userName = userId;
        }
        String password = matcher.group("password");
        /*
         * Sanity checks
         */
        if (password == null) {
            logger.warn("Invalid password");
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.AUTHENTICATION.INVALID");
        }
        if (userName == null) {
            logger.warn("Invalid user");
            throw new UForgeException(UForgeException.UNAUTHORIZED, "ERROR.AUTHENTICATION.INVALID");
        }
        return getUserAuthentication(containerRequest.getRequestUri(), userName, password, targetUser
    }
    protected IUserAuthentication getUserAuthentication(URI requestUri, String userName, String passw
        /*
         * Check with IDM
         */
        ServicesContext.get().getService(IDMService.class).checkUserAuth(userName, password);
        /*
         * Grab users
```

```
        */
        User user;
        User targetUser = null;
        DbAccess db = ServicesContext.get().getService(DbAccess.class);
        try {
            Criteria userCriteria = db.getDbManager().newCriteria(User.class);
            userCriteria.add(Restrictions.eq("loginName", userName));
            user = UserAuthentication.getUser(db, userCriteria);
            if (targetUserName != null) {
                Criteria targetUserCriteria = db.getDbManager().newCriteria(User.class);
                targetUserCriteria.add(Restrictions.eq("loginName", targetUserName));
                targetUser = UserAuthentication.getUser(db, targetUserCriteria);
            }
        } catch (DBException e) {
            throw new UForgeException(UForgeException.DB_ERROR, e);
        }
        logger.debug("Basic Authentication is OK");
        return new UserAuthentication(user, targetUser);
    }
}
```

There are two ways to provide your authentication filter:

- modifying the webservice `web.xml`. You will need to modify the web.xml template by replacing the `com.usharesoft.authentication.AuthenticationFilters` value by your filter classname if you want only your authentication

- using the `@Provider` annotated class (the order can be important if the two authentication methods are used in the same request)

You can add a filter, but you cannot remove the default authentication filters, nor choose the order. To add your filter, use the following Jersey annotation:

```
package my.company.authentication

import javax.ws.rs.ext.Provider;
implements com.usharesoft.authentication.IAuthenticationFilter;

@Provider
class SuperAuthenticationFilter implements IAuthenticationFilter {
...
}
```

Allowing https Repositories with Self-Signed Certificate

> **Warning:** The use of self-signed certificate can create security risks.

The following command allows you to add self-signed certificates (exposed by your repositories) as trusted into the UForge server.

1. Download the SSL self-signed certificate from the server (repository) you want to populate using the following command:

```
$ echo -n | openssl s_client -connect localhost:443 | sed -ne '/-BEGIN CERTIFICATE-/,/-END CERTI
```

2. Recreate the CA file on your UForge platform (containing all trusted CA):

```
$ cp /tmp/cert.crt /etc/pki/ca-trust/source/anchors/
$ update-ca-trust
$ update-ca-trust enable
```

3. Create a link to use the system java keystore instead of the jre default keystore:

```
$ ln -s /etc/pki/java/cacerts /usr/java/latest/jre/lib/security/cacerts
```

The default password is `changeit`.

Populating Database with OS Packages

Open source operating system versions are taken from the official repository mirror or the UForge repository cache. Proprietary operating systems such as Red Hat Enterprise Linux are not; therefore it is the responsibility of the end customer (or reseller if they have correct agreements in place to re-distribute an operating system) to have the original ISO images of the operating system in questions. Refer to *Hosting Proprietary Packages*.

To enable UForge to generate images based on the operating system it needs all the meta-data of the packages comprising the operating system. This meta-data includes the location in the storage of the package as well as dependency information that is used during generation. Furthermore, certain specific UForge packages must be populated for this operating system.

Note: Custom repositories are supported in UForge. They are treated like other OS packages.

Warning: When using UForge, you have to comply with the license agreement of OSes and software which UForge handles, in particular:
- **Publishing OS image of RHEL (Red Hat Enterprise Linux) subscription to public cloud** Cloud provider has to be CCSP (Certified Cloud & Service Provider) and you must register to Red Hat Cloud Access. For more details, please confirm with cloud provider.
- **Scanning server** You have to check whether the licenses of OS and software which the source machine contains allow you to use them on the destination server which you are migrating to. If the source machine contains rpm packages which Red Hat provides, you must register repository with these rpm packages to UForge. Unless you register repository, UForge automatically regenerates rpm packages which the source machine contains, and regenerated packages are NOT supported by Red Hat.
- **Handling Microsoft Windows** Refer to *Microsoft Windows and UForge*.

Note: When installing a major version, all minor versions will be included. If you want to restrict to only a few minor versions, you will have to follow this procedure for each minor version you want to install.

Warning: If you are going to use the migration feature for RHEL or CentOS, then it is highly recommended to register the major version repositories.

In order to add an operating system in your UForge AppCenter you must:

1. Connect to one of your UForge platform instances

2. Create the OS in the organization.

3. Create the repository. This includes the official repository (see *Official UForge Tool Repositories*) as well as the specific UForge tool repository (see *Specific UForge Tool Repositories*). This is covered in steps 6 and 7 in the section *Example for Adding CentOS*.

---

4. Link the distribution to the repository.

5. Launch spider to fill the repository with the packages.

Official UForge Tool Repositories

The following is a list for all the distributions that can be used to create an official repository. These will be used in step 6 of the examples below.

    Ubuntu (example 10.04)

- http://distros-repository.usharesoft.com/ubuntu/lucid/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid multiverse restricted universe main

- http://distros-repository.usharesoft.com/ubuntu/lucid-security/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-security multiverse restricted universe main

- http://distros-repository.usharesoft.com/ubuntu/lucid-backports/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-backports multiverse restricted universe main

- http://distros-repository.usharesoft.com/ubuntu/lucid-updates/mirror/bouyguestelecom.ubuntu.lafibre.info/ubuntu/ lucid-updates multiverse restricted universe main

    Debian (example version 6)

- http://distros-repository.usharesoft.com/debian/squeeze/mirror/ftp.fr.debian.org/debian/ squeeze contrib non-free main

- http://distros-repository.usharesoft.com/debian/squeeze-updates/mirror/ftp.fr.debian.org/debian/ squeeze-updates contrib non-free main

- http://distros-repository.usharesoft.com/debian/security/squeeze/updates/mirror/security.debian.org/ squeeze/updates main contrib non-free

    CentOS (example CentOS 6.7)

- http://distros-repository.usharesoft.com/centos/6.7/updates/x86_64

- http://distros-repository.usharesoft.com/centos/6.7/extras/x86_64

- http://distros-repository.usharesoft.com/centos/6.7/os/x86_64

    OpenSUSE (example version 12.2)

- http://distros-repository.usharesoft.com/opensuse/distribution/12.2/repo/oss/

- http://distros-repository.usharesoft.com/opensuse/distribution/12.2/repo/non-oss/

- http://distros-repository.usharesoft.com/opensuse/update/12.2/

    Scientific Linux (example version 6.6)

- http://distros-repository.usharesoft.com/scientificlinux/6.6/x86_64/os/

- http://distros-repository.usharesoft.com/scientificlinux/6.6/x86_64/updates/fastbugs/

- http://distros-repository.usharesoft.com/scientificlinux/6.6/x86_64/updates/security/

    RedHat Enterprise Linux

- You need to use your own repository.

Specific UForge Tool Repositories

The following is a list of specific UForge tool repositories that can be added. These will be used in step 7 of the examples below.

CentOS (example version 6, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/centos/6/x86_64/

Red Hat Enterprise Linux (example version 6.2, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/rhel/6.2/x86_64/

OpenSUSE (example version 12.1, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/opensuse/12.1/x86_64/

Scientific Linux (example version 6, arch x86_64):

- http://distros-repository.usharesoft.com/usharesoft/scientificlinux/6/x86_64/

Debian (example version 8, arch x86_64) [arch=amd64]:

- http://distros-repository.usharesoft.com/usharesoft/debian/ jessie main

Ubuntu (example version 14.04, arch x86_64) [arch=amd64]:

- http://distros-repository.usharesoft.com/usharesoft/ubuntu/ trusty main

Adding RPM Type OSes

The following sections give examples for adding CentOS and RedHat Enterprise Linux. They can be adjusted for your particular version, and are applicable to OpenSUSE and Scientific Linux.

Example for Adding CentOS

The following is a concrete example to begin the population of CentOS 6.5 64bit:

1. Connect to UForge:

```
$ ssh root@<your UForge instance>
```

2. In order for the following commands to be generic you can set some variables in your environment.

```
$ . /etc/UShareSoft/uforge/uforge.conf
ADMIN=$UFORGE_WEBSVC_LOGIN ; PASS=$UFORGE_WEBSVC_PASSWORD
```

3. Run the following CLI command in order to create the distribution:

```
$ uforge org os add --name CentOS --arch x86_64 --version 6.5 -u $ADMIN -p $PASS
```

4. Enable the new operating system for the organization. The following command enables CentOS 6.5 in the default organization:

```
$ uforge org os enable --name CentOS --version 6.5 --arch x86_64 -u $ADMIN -p $PASS
```

5. Enable the user to use the operating system. The user must be a member of the organization. This step can be done later.:

```
$ uforge user os enable --account root --name CentOS --version 6.5 --arch x86_64 -u $ADMIN -
```

6. Create the distribution repository. The following example shows the creation of an official CentOS repository. However, you can also create a repository based on the UForge official repository as shown later.

   For example, for the CentOS 6.5 repository:

   ```
   $ uforge org repo create --name "CentOS 6.5 os" --repoUrl http://vault.centos.org/6.5

   Success: Created repository with url [http://vault.centos.org/6.5/os/x86_64/] to defa
   ```

   The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance. The `--repoUrl` can be either `http://` or `file://`.

   > **Warning:** You must use the `--officiallySupported` flag for all officially supported OSes. If you do not include this argument the packages will not appear in the install profile of appliances built with the corresponding operating system. Do not use `--officiallySupported` for distributions that are part of the core distribution. For example, epel or vmwatools are not officially part of the distribution, therefore you should not use `--officiallySupported` when adding such repositories.

   http://distros-repository.usharesoft.com/ is an official public repository that users can use to populate the distributions. Official repositories such as Ubuntu and Debian periodically delete some package versions. In the http://distros-repository.usharesoft.com/ repository, package versions are never deleted. This can facilitate investigations on older systems.

7. You must then add the specific UForge tool repository. The repository to attach for `CentOS` (example version 6, arch x86_64) is the following:

   - http://distros-repository.usharesoft.com/usharesoft/centos/6/x86_64/

   For example:

   ```
   $ uforge org repo create --name "CentOS 6.5 os" --repoUrl http://distros-repository.usha
   ```

   > **Note:** For a complete list of the different repositories that can be attached, refer to *Specific UForge Tool Repositories*.

8. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

   ```
   $ uforge org repo os attach --name CentOS --arch x86_64 --version 6.5 --repoIds 354 -u $ADMI
   ```

   The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

9. Populate repository packages:

   ```
   $ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
   ```

   > **Note:** This procedure may take a long time.

10. To verify if the procedure is terminated, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/<directory name with date>/s
```

The procedure is complete when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

11. Create OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/runjob.py sorter_low_prio -d CentOS -v 6.5 -a x86_64
```

Example for Adding RedHat Enterprise Linux

The following is a concrete example to begin the population of RedHat Enterprise Linux version 7, 64bit:

1. Connect to UForge:

```
$ ssh root@<your UForge instance>
```

2. In order for the following commands to be generic you can set some variables in your environment.

```
$ . /etc/UShareSoft/uforge/uforge.conf
ADMIN=$UFORGE_WEBSVC_LOGIN ; PASS=$UFORGE_WEBSVC_PASSWORD
```

3. Run the following CLI command in order to create the distribution:

```
$ uforge org os add --name "RedHat Enterprise Linux" --arch x86_64 --version 7 -u $ADMIN -p
```

4. Enable the new operating system for the organization. The following command enables CentOS 6.5 in the default organization:

```
$ uforge org os enable --name "RedHat Enterprise Linux" --arch x86_64 --version 7 -u $ADMIN
```

5. Enable the user to use the operating system. The user must be a member of the organization. This step can be done later.:

```
$ uforge user os enable --account root --name "RedHat Enterprise Linux" --arch x86_64 --vers
```

6. Create the distribution repository. The following example shows the creation of an official RedHat Enterprise Linux repository.

```
$ uforge org repo create --name "RedHat 7" --repoUrl http://<your-repo> --type RPM --off
```

The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance. The `--repoUrl` can be either `http://` or `file://`.

> **Warning:** You must use the `--officiallySupported` flag for all officially supported OSes. If you do not include this argument the packages will not appear in the install profile of appliances built with the corresponding operating system. Do not use `--officiallySupported` for distributions that are part of the core distribution. For example, epel or vmwatools are not officially part of the distribution, therefore you should not use `--officiallySupported` when adding such repositories.

7. You must then add the specific UForge tool repository. The repository to attach for RedHat Enterprise Linux version 7 arch x86_64 is the following:

   • http://distros-repository.usharesoft.com/usharesoft/rhel/7/x86_64/

For example:

```
$ uforge org repo create --name "UShareSoft RedHat 7" --repoUrl http://distros-repository
```

Note: For a complete list of the different repositories that can be attached, refer to *Specific UForge Tool Repositories*.

8. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

```
$ uforge org repo os attach --name "RedHat Enterprise Linux" --arch x86_64 --version 7 --rep
```

The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

9. Populate repository packages:

```
$ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

Note: This procedure may take a long time.

10. To verify if the procedure is terminated, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/<directory name with date>/sp
```

The procedure is complete when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

11. Create OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/runjob.py sorter_low_prio -d "RedHat" -v 7 -a x86_64
```

Adding DEB Type OSes

The following section give an example for adding Ubuntu. It is also applicable for Debian.

Example for Adding Ubuntu

The following is a concrete example to begin the population of Ubuntu 10.04 64bit:

1. Connect to UForge:

```
$ ssh root@<your UForge instance>
```

2. In order for the following commands to be generic you can set some variables in your environment.

```
$ . /etc/UShareSoft/uforge/uforge.conf
ADMIN=$UFORGE_WEBSVC_LOGIN ; PASS=$UFORGE_WEBSVC_PASSWORD
```

3. Run the following CLI command in order to create the distribution:

```
$ uforge org os add --name Ubuntu --arch x86_64 --version 10.04 -u $ADMIN -p $PASS
```

4. Enable the new operating system for the organization. The following command enables Ubuntu 10.04 in the default organization:

```
$ uforge org os enable --name Unbuntu --version 10.04 --arch x86_64 -u $ADMIN -p $PASS
```

5. Enable the user to use the operating system. The user must be a member of the organization. This step can be done later.:

```
$ uforge user os enable --account root --name Unbuntu --version 10.04 --arch x86_64 -u $ADMI
```

6. Create the distribution repository. The following example shows the creation of an official Ubuntu repository.

```
$ uforge org repo create --name "Ubuntu x86_64 lucid-main" --repoUrl "[arch=amd64] http:

$ uforge org repo create --name "Ubuntu x86_64 lucid-security" --repoUrl "[arch=amd64] h

$ uforge org repo create --name "Ubuntu x86_64 lucid-backports" --repoUrl "[arch=amd64]

$ uforge org repo create --name "Ubuntu x86_64 lucid-updates" --repoUrl "[arch=amd64] ht
```

The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance. The `--repoUrl` can be either `http://` or `file://`.

> **Warning:** You must use the `--officiallySupported` flag for all officially supported OSes. If you do not include this argument the packages will not appear in the install profile of appliances built with the corresponding operating system. Do not use `--officiallySupported` for distributions that are part of the core distribution. For example, epel or vmwatools are not officially part of the distribution, therefore you should not use `--officiallySupported` when adding such repositories.

The syntax of the repoURL for Debian based OSes follows that of the sources.list file.

See https://wiki.debian.org/SourcesList and https://wiki.debian.org/Multiarch/HOWTO (section Setting up apt sources)

Typically, a correct value for the repoURL parameter is either

- http://httpredir.debian.org/debian jessie main

- http://ftp.riken.go.jp/Linux/ubuntu/ precise-security multiverse restricted universe main

Users may also want to restrict per architecture. For example:

```
[arch=amd64] http://distros-repository.usharesoft.com/ubuntu/ ...
```

http://distros-repository.usharesoft.com/ is an official public repository that users can use to populate the distributions. Official repositories such as Ubuntu and Debian periodically delete some package versions. In the http://distros-repository.usharesoft.com/ repository, package versions are never deleted. This can facilitate investigations on older systems.

7. You must then add the specific UForge tool repository. The repository to attach for `CentOS` (example version 6, arch x86_64) is the following:

- http://distros-repository.usharesoft.com/usharesoft/ubuntu/

For example:

```
$ uforge org repo create --name "UShareSoft Ubuntu x86_64 lucid" --repoUrl "[arch=amd64]
```

Note: For a complete list of the different repositories that can be attached, refer to *Specific UForge Tool Repositories*.

---

8. Attach repository to the distribution as follows for each repository (your own repository and the UShareSoft tool repository):

```
$ uforge org repo os attach --name Ubuntu --arch x86_64 --version 10.04 --repoIds 354 -u $AD
```

The `--repoIds` specified here are the space-separated "id" of previously created repositories, shown by command `uforge org repo list -u $ADMIN -p $PASS`.

9. Populate repository packages:

```
$ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

Note: This procedure may take a long time.

---

10. To verify if the procedure is terminated, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/<directory name with date>/sp
```

The procedure is complete when you see the line `INFO` ends with `Entering CheckForRepositoriesUpdates->terminate()`

11. Create OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/runjob.py sorter_low_prio -d Ubuntu -v 10.04 -a x86_64
```

Hosting Proprietary Packages

Proprietary packages, such as Red Hat Enterprise Linux are not delivered as part of the UForge repository. You must have the original ISO images of the operating system in questions and follow the steps below.

For example, to add a Red Hat repository:

1. Mount the iso into `/mnt` (on the works node)

2. Create the appropriate directory layout under `/tmp/USER_DATA/repos/` for example: `/tmp/USER_DATA/repos/RHEL/6.5/x86_64/`

3. Copy all the contents of the DVD into `/tmp/USER_DATA/repos/RHEL/6.5/x86_64/`

4. If the repository does not already contain a repodata folder, you must create it inside the package directory:

```
$ cd /tmp/USER_DATA/repos/RHEL/6.5/x86_64/
$ createrepo .
```

5. Create a file in `/etc/httpd/conf.d` called `repos.conf`. The file should contain the following:

```
Alias /repos /tmp/USER_DATA/repos

<Directory /tmp/USER_DATA/repos>
```

```
        Options +Indexes
    </Directory>
```

6. Run the following from the command line:

```
service httpd restart
```

7. Create the repository using the UForge CLI as follows:

```
$ uforge org repo create --name "RHEL 6.5 os" --repoUrl "http://MACHINE_IP/repos/RHEL/6.5/x86_64
```

The `--name` specified here is the "tagname" that will be shown in the UI when creating an appliance.

8. Attach the repository to the distribution as follows:

```
$ uforge org repo os attach --name RHEL --repoIds 954 -u $ADMIN -p $PASS
```

9. Populate the repository packages:

```
$ /opt/UShareSoft/uforge/cron/update_repos_pkgs.sh
```

This procedure may take a long time.

10. To verify if the procedure is terminated, run the following command:

```
$ tail -f /tmp/USER_DATA/FactoryContainer/logs/repos/spider/<directory name with date>/spider.st
```

The procedure is terminated when you see the line:

```
INFO  CheckForRepositoriesUpdates:275 - Entering CheckForRepositoriesUpdates->terminate()
```

11. Create an OS profile based on packages (minimal, server, etc.):

```
$ /opt/UShareSoft/uforge/bin/runjob.py sorter_low_prio -a x86_64 -d RHEL -v 6.5
```

Updating an Existing UForge Deployment

All the UForge components are delivered as native RPM packages. We maintain a custom repository of the UForge platform. All updates are added to this repository. The update mechanism of a remote UForge platform uses the standard "yum" command-line package management utility.
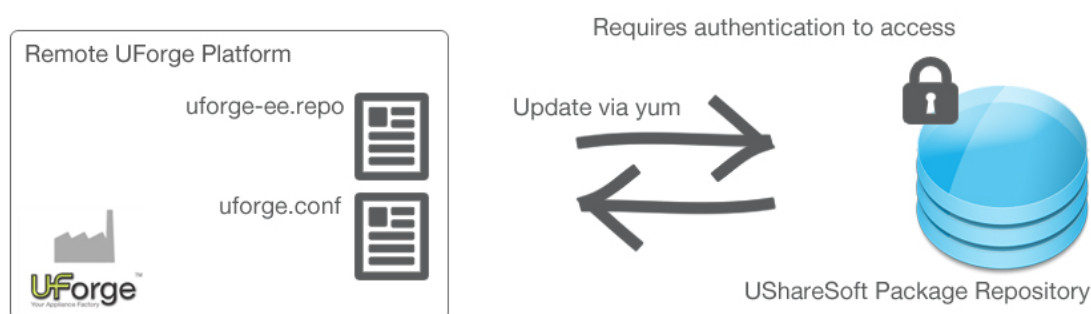
In order to interact with the official package repository, you must already have an active UForge license (stored in Fujitsu's database) and corresponding authentication credentials, set during the initial install of the platform. The authentication credentials are stored in the uforge.conf file and used in the uforge-ee.repo file. To view these files:

```
$ vi /etc/UShareSoft/uforge/uforge.conf
UFORGE_PRODUCT_ACCESS_USER=username
UFORGE_PRODUCT_ACCESS_PASSWORD=password
```

Note: You should never update the following file: `/etc/yum.repo.d/uforge-ee-repo`

The request to update uses these credentials via HTTPS to Fujitsu who then determines whether you have the access rights to update the platform.

You must also have set up the yum repo file to authorize UForge updates using configure_yum_repos.sh as follows:

```
$ /opt/UShareSoft/uforge/conf/configure_yum_repos.sh -u <uss account user> -p <uss account password>
```

The (optional) parameter <uforge install type> can take the values uforge-ee or uforge-dev. By default uforge-ee is used.

To update the platform, use the "yum" command-line tool as follows:

Note: Running `yum update` may also update OS packages from CentOS official repository. You should accept all

the updates because UForge is qualified based on the latest packages.

```
$ yum update

Loaded plugins: presto
UForge-ee-uforge                                               |  951 B     00:00
UForge-ee-uforge/primary                                       | 1.9 kB     00:00
UForge-ee-uforge
                                                                                  4/4
UForge-ee-uforge-client                                        |  951 B     00:00
UForge-ee-uforge-client/primary                                | 1.0 kB     00:00
UForge-ee-uforge-client
                                                                                  4/4
fedora/metalink                                                | 2.6 kB     00:00
fedora                                                         | 4.3 kB     00:00
fedora/primary_db                                              |  13 MB     00:36
updates/metalink                                               | 1.9 kB     00:00
updates                                                        | 4.7 kB     00:00
updates/primary_db                                             | 6.4 MB     00:21
Setting up Update Process
Resolving Dependencies
--> Running transaction check
---> Package MySQL-client.x86_64 0:5.5.23-1.linux2.6 set to be installed
---> Package MySQL-server.x86_64 0:5.5.23-1.linux2.6 set to be installed
---> Package MySQL-shared.x86_64 0:5.5.23-1.linux2.6 set to be installed
---> Package glassfish.noarch 0:3.1-2 set to be updated
---> Package perl-Compress-Raw-Zlib.x86_64 0:2.030-1.fc13 set to be updated
---> Package perl-Test-Simple.noarch 0:0.94-1.fc13 set to be updated
---> Package perl-parent.noarch 1:0.223-3.fc13 set to be updated
---> Package perl-threads.x86_64 0:1.81-1.fc13 set to be updated
---> Package uforge.noarch 0:3.2.5-0 set to be updated
---> Package uforge-client.noarch 0:3.2.5-0 set to be updated
--> Finished Dependency Resolution

Dependencies Resolved

================================================================================================
 Package                        Arch          Version            Repository           Size
================================================================================================
Installing:
 MySQL-client                   x86_64        5.5.23-1.linux2.6  UForge-ee-mysql       14 M
     replacing  MySQL-client.x86_64 5.5.17-1.linux2.6
 MySQL-server                   x86_64        5.5.23-1.linux2.6  UForge-ee-mysql       40 M
     replacing  MySQL-server.x86_64 5.5.17-1.linux2.6
 MySQL-shared                   x86_64        5.5.23-1.linux2.6  UForge-ee-mysql      1.7 M
     replacing  MySQL-shared.x86_64 5.5.17-1.linux2.6
Updating:
 glassfish                      noarch        3.1-2              UForge-ee-glassfish   85 M
 perl-Compress-Raw-Zlib         x86_64        2.030-1.fc13       updates               57 k
 perl-Test-Simple               noarch        0.94-1.fc13        updates              116 k
 perl-parent                    noarch        1:0.223-3.fc13     fedora                13 k
 perl-threads                   x86_64        1.81-1.fc13        updates               47 k
 uforge                         noarch        3.2.5-0            UForge-ee-uforge  38 M

Transaction Summary

================================================================================================
Install       3 Package(s)
Upgrade       7 Package(s)
```

```
Total download size: 204 M
Is this ok [y/N]: y

<traces removed for readability>


....

Complete!
```

The RPM packages will be replaced and the services will be reconfigured to correctly update the platform. If you have a multi-node UForge platform, then this command must be run on all the nodes. The updates should be run in the following order:

1. database node

2. compute node(s)

3. web service and Portal nodes

Run the following CLI command in order to know if Squid is running:

```
$ service squid status
```

If squid is stopped, run the following command-line

```
$ service squid start
```

Going Back to a Previous Version of a UForge Deployment

Note: Using `yum downgrade` to return to a previous version of UForge is not supported.

Even though the `yum downgrade` command works from a packaging perspective, it will not roll back possible changes done to the database (especially the database schema).

Moreover, there are additional factors that may lead to fatal errors, including configuration or properties files which are not considered by `yum downgrade`, or possible changes of users and their permissions on the file system.

If you need to downgrade a UForge deployment to a previous version, you should create a snapshot of your machine prior to the upgrade.

Retrieving Data from UForge

Before retrieving data from resellers.usharesoft.com using the lftp command from a UForge instance do the following:

1. Verify if the UForge instance is running in a virtualized infrastructure with security rules by default (AWS, OpenStack, ...). Ports 20, 21 (as well as 22 for SSH) must be allowed for outgoing traffic.

2. Due to the new proxy mechanism you must run:

```
export ftp_proxy=""
```

This ensures that squid does not interfere with ftp transfer.

Sending a Request to UForge

As the UForge Web Services are RESTful, clients communicate via the standard HTTP(S) protocol. That means you can easily construct request URLs that will work on the command line and in your code.

All UForge requests (with some exceptions) require authentication information as part of the request. There are several ways to communicate with UForge:

- Using API keys – A public and secret key is used to construct the URL. This URL will contain public key and a signature that authenticates the request.

- Basic Authentication – Where the login name and password are provided in the requesting HTTP(S) headers.

- Custom – UForge provides AuthN and AuthZ modules that can be customized to provide other authentication mechanisms (refer to Customizing UForge Authentication for SSO).

All request URLs start with the hostname of where UForge is running, the port where UForge is listening for incoming requests, the service name and version number. This is known as the BASE URL.

Even though UForge accepts HTTP requests, it is highly recommended for security reasons that HTTPS requests be used. HTTP requests should only be used for debugging purposes. Sensitive information will be exposed using HTTP.

UForge expects certain headers containing authentication information to be present as part of the URL request. UForge also accepts other header information, for example, to specify response content type and caching.

The following is an example of a request sent to a UForge AppCenter with hostname 10.0.0.20 using cURL to get the user `myUser`. Note that basic authentication is used for clarity.

```
$ curl "http://10.0.0.20:9090/ufws/users/myUser" -H "Authorization: Basic myUser:password" -H "Accept
        * About to connect() to 10.0.0.20 port 8080 (#0)
        * Trying 10.0.0.20... connected
        * Connected to 10.0.0.20 (10.0.0.20) port 8080 (#0)
        > GET /ufws/users/myUser HTTP/1.1
        > User-Agent: curl/7.19.7 (universal-apple-darwin10.0) libcurl/7.19.7 OpenSSL/0.9.8r zlib/1.2
        > Host: 10.0.0.20:8080
        > Accept: application/xml
        >

        < HTTP/1.1 200 OK
        < X-Powered-By: Servlet/2.5
        < Server: Sun GlassFish Enterprise Server v3.1.2
        < Last-Modified: Thu, 08 Aug 2013 19:52:13 GMT
        < ETag: "80f76a81b033572861260548dd748bb3"
        < Content-Type: application/xml
        < Transfer-Encoding: chunked
        < Date: Thu, 21 Jul 2011 17:02:10 GMT
```

```
        <

* Closing connection #0
<?xml version="1.0" encoding="utf-8" standalone="yes"?>
<user>
...omitted for clarity
</user>
```

Backup Overview Guidelines

> **Warning:** Fujitsu is not responsible for any customer data loss. The database backup techniques highlighted in this document are standard best practices used by the industry.

The goal of this section is to highlight some best practices on how to correctly backup the UForge data. Before you begin you might want to read:

Backup Overview Guidelines

> **Warning:** Fujitsu is not responsible for any customer data loss. The database backup techniques highlighted in this document are standard best practices used by the industry.

There are two zones of UForge data that should be backed up:

- UForge database service.
- UForge user data (user uploaded software etc)

Database replication and backup are sometimes confused as achieving the same goal. Many people believe that by having replication, the data within the database is being stored elsewhere and therefore in the event of a serious problem, the data can be recovered. However, with replication, any accidental changes or deletions are also replicated to the other systems.

Database replication is meant for high-availability of your service, in other words building a redundant database service. If a problem occurs with one database instance, then having a second database with the data replicated, this instance can continue providing the database service (even if the service is degraded).

Database backup is used to take a snapshot of the database data (either a complete or incremental snapshot) and storing this information in an archive.

When thinking about your database strategy, you should consider the following goals:

- The backup should be an internally consistent snapshot (this may seem obvious, but if you do not lock the database or turn the database off before doing the snapshot then the data may be inconsistent).

- Keep the service running during the backup (in production systems this will be mandatory, however for less-critical platforms this may not be a necessary goal)

- Make the backup automated (via cron or using some 3rd party tools)

Backup Recommendations

> **Warning:** Fujitsu is not responsible for any customer data loss. The database backup techniques highlighted in this document are standard best practices used by the industry.

Here are some general recommendations and guidelines when doing database backup.

Never Back Up Databases to Local Disk

If the database server crashes, especially due to a hardware problem or a severe OS problem, the local drives may not be available. In the event of a hardware disaster, if the archives and current copy is on the shared filesystem allowing someone to quickly build a new server and start doing restores while resuscitating the server.

Back up Databases to a Fileshare, then Back the Share to Tape

Tape drives these days are fast enough that the vendors like to say DBAs should go straight to tape, and they are technically right: tape backup & restore speed is not a bottleneck. However, there is a limited number of drives available. This, however, gives you an extra layer of backup protection in case something goes wrong with the shared filesystem.

Back Up to a Different NAS/SAN

This is not possible for everyone, however, in extremely rare occasions where the production NAS/SAN does go down then at least the backup array was not affected as it is on a different NAS/SAN than the production system.

Consider Raid 10 SATA

Depending on the backup windows, multiple database servers may be writing backups to that same share simultaneously. Raid 10 gives better write performance than raid 5, (using fiber channel backup drives is even better but cost-prohibitive). Raid 10 on SATA gets a good balance of cost versus backup throughput.

Using Master-Slave Replication for Database Backup

To ensure consistent backups of the data and to limit the impact of the actual backup process on the service performance, a common approach is to use Master-Slave replication. This is an easy way to provide a complete running copy of the data on another database instance (called the slave). Consequently you can then stop the slave instance to do the backups while leaving the other instance (the master) to continue providing the database service. Master-slave replication is also the first implementation of high-availability.

The configuration of master-slave replication is out of scope of this document, but we recommend the following reading:

- How to Set Up Replication, MySQL documentation
- High Availability MySQL Cookbook
- MySQL High Availability by O'Reilly

You can now safely run the same commands provided in section *UForge Databases Basic Backup* on the slave database instance.

To automate the entire process, a script can be written and executed by a cron job at the frequency you wish to back up the database (nightly backups are recommended).

If you want to add high-availability, then we recommend using MySQL Cluster. This is a separate MySQL product that is not distributed with UForge.

The following zones of UForge data should be backed up:

UForge Databases Basic Backup

UForge has three databases to store all its data, namely:

- LDAP (OpenDJ) for storing user credential information
- IDM (Syncope using MariaDB) for storing user entitlements and roles
- UForge (MariaDB) for storing everything else for UForge features.

MariaDB provides many different types of backup methods you can choose from, including:

- Doing a hot backup (for example with Percona XtraBackup)
- Copying Table files
- Delimited-Text backups
- Using the mysqldump or mysqlhotcopy commands
- Creating incremental backups by enabling the binary log
- Creating a backup from a slave
- Making backups using a file system snapshot

For more details on all these back up strategies, refer to: http://dev.mysql.com/doc/refman/5.5/en/backup-methods.html

MariaDB holds both UForge and IDM data. All this information is located under:

```
/var/lib/mysql
```

By copying this information, you are taking a snapshot of the entire database.

LDAP (OpenDJ) has its own tool for doing backup, called `backup`. To ensure data consistency across the entire platform, we recommend you backup all the databases at the same time. Here is an example of how to back up the MariaDB (holding IDM and UForge information) and LDAP databases. Note, we are also stopping the web service instances, this ensures we do not generate connection error messages in the logs:

```
$ service tomcat stop
$ service mysql stop
$ rsync -a --delete-before /var/lib/mysql /tmp/dir_to_backup/mysql
$ /opt/OpenDJ/bin/backup -a -c -d /tmp/dir_to_backup/ldap
$ rsync -a --delete-before /tmp/dir_to_backup/* root@/tmp/dir_to_backup/
$ service mysql start
$ service tomcat start
```

As the backup destination does not change, you are creating a replication of the entire database. The rsync command only copies the incremental differences each time.

The danger here is that if during that time there is damaged data, when rsync is run then this corrupted data is copied and there is no possible way to recuperate the old data. To overcome this, we also create archive backups of the entire database.

```
$ service tomcat stop
$ service mysql stop
$ f=`date +%y%m%d`
$ l=`echo "$f-ldap-backup.tgz"` #what to name ldap backups
$ f=`echo "$f-mysql-backup.tgz"` #what to name backups
$ tar -czf /tmp/$f /var/lib/mysql/*
$ /opt/OpenDJ/bin/backup -a -c -d /tmp/ldap_backup
$ tar -czf /tmp/$l /tmp/ldap_backup
$ rsync -a /tmp/$f root@<BACKUP-DESTINATION>
```

```
$ rsync -a /tmp/$1 root@<BACKUP-DESTINATION>
$ service mysql start
$ service tomcat start
```

This only works if the database instance is stopped to ensure a consistent dump of the database. Otherwise you get a corrupt, inconsistent backup.

The issue with this method is if you have only one database instance, then you are effectively stopping your service to do the backup. To overcome this you need to use replication, for example master-slave.

Note: For LDAP, the service does not need to be stopped.

We recommend you use a mounted disk, SAN or NAS to store your backup archives.

Basic Restore

Restoring the database is a simple copy using rsync back to the database directory. Note the use of slashes (/) is important. You should stop the web service to ensure we do not generate connection error messages in the logs.

```
$ service tomcat stop
$ service OpenDJ stop
$ service mysql stop
$ rsync -a --delete-before root@<BACKUP-DESTINATION>/<ldap tarball> /tmp
$ tar -xvf /tmp/<ldap tarball>
$ /opt/OpenDJ/bin/restore -d /tmp/<ldap dir>/userRoot
$ /opt/OpenDJ/bin/restore -d /tmp/<ldap dir>/tasks
$ /opt/OpenDJ/bin/restore -d /tmp/<ldap dir>/config
$ /opt/OpenDJ/bin/restore -d /tmp/<ldap dir>/schema
$ rsync -a --delete-before root@<BACKUP-DESTINATION>/mysql/ /var/lib/mysql
$ service OpenDJ start
$ service mysql start
$ service tomcat start
```

User Data Backup

UForge also stores information that is uploaded by the users of the platform. These are:

- packages, binaries and files that are uploaded in `My Software` or `Projects`
- images (photos, appliance logos etc)
- boot scripts
- licenses (attached to projects or My Software)

All this information is stored in the following directory: `/tmp/USER_DATA`

Like the database, this is important information that must be backed up on a regular basis. The same mechanism can be used for back up as the database, namely using rsync as highlighted in *UForge Databases Basic Backup*.

You do not need to stop the service, but the danger is that if during that time there is damaged data, when rsync is run then this corrupted data is copied and there is no possible way to recuperate the old data.

The time it will take to back up the data will depend on the size of your data and the connection between the two servers.

Manage Services

The following sections cover information regarding managing the UForge services.

Starting and Stopping the Application Server

Tomcat is registered as an operating system service. Each time a web service node is rebooted, the application server is restarted automatically as part of the node boot sequence. To manually stop or start the web service you must be the unix root user of the node.

Stopping the web service

```
$ service tomcat stop
```

Starting the web service

```
$ service tomcat start
```

Restarting the web service

```
$ service tomcat restart
```

Starting and Stopping the Database

The database is registered as an operating system service. Each time a database node is rebooted, MariaDB is restarted automatically as part of the node boot sequence. To manually stop or start the database you must be the unix root user of the node. Log in to the database node as root:

```
$ ssh root@<ip address of the node>
```

Stopping the Database

```
$ service mysql stop
```

Starting the Database

```
$ service mysql start
```

Restarting the Database

```
$ service mysql restart
```

Manage Resources

The following sections cover information regarding managing the UForge resources.

Viewing the Installed OSes

All the open source operating system versions are delivered as part of the UForge repository. Proprietary operating systems are not.

To get the complete list of the currently supported operating systems on the UForge platform use the command `uforge os list`.

Log in to one of the UForge instances:

```
$ uforge os list -u $ADMIN -p $PASS
Getting operating systems ...
Success: Found the following operating systems
+---------------+---------------+--------------+--------+---------+---------+------------------+
| Distribution  | Version       | Architecture | Access | Visible | Default | Release Date     |
+---------------+---------------+--------------+--------+---------+---------+------------------+
| CentOS        | 5             | i386         |   X    |    X    |         | NK               |
| CentOS        | 5.3           | i386         |   X    |    X    |         | 2009-03-17       |
| CentOS        | 5.3           | x86_64       |   X    |    X    |         | 2009-03-18       |
| CentOS        | 5.4           | i386         |   X    |    X    |         | 2009-10-01       |
| CentOS        | 5.4           | x86_64       |   X    |    X    |         | 2009-10-01       |
| CentOS        | 5.5           | i386         |   X    |    X    |         | 2010-04-27       |
| CentOS        | 5.5           | x86_64       |   X    |    X    |         | 2010-04-27       |
| CentOS        | 5.6           | i386         |   X    |    X    |         | 2011-03-07       |
| CentOS        | 5.6           | x86_64       |   X    |    X    |         | 2011-03-22       |
| CentOS        | 5.7           | i386         |   X    |    X    |    X    | 2011-08-29       |
.... rest omitted for clarity
```

All the open source operating system versions are delivered as part of the UForge repository. Proprietary operating systems such as RedHat Enterprise Linux or older operating system versions (that have been EOL'd) are not; therefore it is the responsibility of the end customer (or reseller if they have correct agreements in place to re-distribute an operating system) to have the original ISO images of the operating system in questions.

Note: You can only add an operating system version that is officially supported by the UForge AppCenter and has been certified by Fujitsu.

Viewing the Enabled OSes

To get a list of the operating systems that are currently enabled on your UForge platform use the command `uforge org os list`.

Login to one of the UForge instances:

```
$ uforge org os list -u $ADMIN -p $PASS
Getting operating systems ...
Success: Found the following operating systems
+---------------+---------------+-------------+-------+--------+--------+------------------+
| Distribution  | Version       | Architecture | Access | Visible | Default | Release Date     |
+---------------+---------------+-------------+-------+--------+--------+------------------+
| CentOS        | 6.3           | i386        |   X   |        |        | 2012-07-01       |
| CentOS        | 6.3           | x86_64      |   X   |        |        | 2012-06-26       |
| CentOS        | 6.4           | i386        |   X   |        |        | 2013-03-01       |
| CentOS        | 6.4           | x86_64      |   X   |   X    |        | 2013-03-01       |
.... rest omitted for clarity
```

Adding an OS to an Organization

Older operating system versions (that for example have been EOL'd) or proprietary operating systems such as Red-Hat Enterprise Linux are not automatically populated at the installation phase. Population of such operating system versions must be done manually after the initial installation of UForge is complete.

Note: You can only add an operating system version that is officially supported by the UForge platform and has been certified by Fujitsu.

Before adding an OS to the organization you must retrieve data from the Fujitsu repository. For more information refer to *Populating Database with OS Packages*.

In order to add an OS to an organization you must do the following:

1. Add the distribution to the organization.

2. Create the repositories.

3. Attach the repository to the distribution.

4. Launch Spider to fill the repositories.

This must be done for each version of an OS. For example CentOS 6.5 i386. It is not possible to do this for all CentOS versions at once.

For example, to add CentOS 6.5 i386 to the default org:

In order to add the OS to a specific distribution, you will need to specify `--org <value>` for each of the commands in the steps below.

1. Add the distribution to the organization, using the official name and version.

```
$ uforge org os add --name CentOS --version 6.5 --arch i386 -u $ADMIN -p $PASS
Getting default organization ...
Success: Add operating system OK
```

2. Create the repositories.

```
$ uforge org repo create --name CentOSplus --repoUrl http://vault.centos.org/6.5/centosplus/i386
Getting default organization ...
Success: Created repository with url [http://vault.centos.org/6.5/centosplus/i386] to default or
+------+---------------+-----------+-------------------------------------------+------+
|   Id | Off. Supported | Name      | Url                                       | type |
+------+---------------+-----------+-------------------------------------------+------+
|  355 |               | CentOSplus | http://vault.centos.org/6.5/centosplus/i386 | RPM  |
```

3. Attach the repository to the distribution

```
$ uforge org repo os attach --name CentOS --version 6.5 --arch i386 --repoId 355 -u $ADMIN -p $P
Getting default organization ...
Success: Attach distribution to repository [355]
```

4. Launch Spider and other population steps. Refer to *Populating Database with OS Packages*.

When adding CentOS, Debian and RedHat, the major versions are automatically marked as Milestones when the distribution is added to the platform. For more information on Milestones, refer to *Creating and Managing Milestones*.

Removing OSes and Distributions

You cannot remove an OS from an organization once added. You can only disable it, in which case it can no longer be used. To disable a distribution, for example CentOS for all users of an organization you can specify only the OS name, in which case all the versions will be removed:

```
$ uforge org os disable --name CentOS -u $ADMIN -p $PASS
```

If you only want to remove a specific version of a distribution (for example CentOS 5), run:

```
$ uforge org os disable --name CentOS --version 5 -u $ADMIN -p $PASS
```

Creating Custom OS Profiles

UForge provides a set of OS profiles. If you want to create another OS profile, you can create one using UForge's graphical user interface.

Note: You can delete packages but we do not guarantee that your OS profile will be functional.

To create a new OS profile:

1. Under the `Administration` tab, click `OS Profiles`.

2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.

3. Double-click on the operating system you want to administer. This will provide the current list of profiles this operating system has.

4. Select the profile which will serve as the base for your custom OS profile and click create.

5. Enter your profile name and click create.

6. Enter a description (mandatory).

7. Select the package(s) you want to add to your OS profile and click `save`. You can search for the package you want to add. The packages that you can add are listed in the top list. The bottom list is the list of packages that are currently part of the profile.

8. Click `Save`.

Editing Custom OS Profiles

If you have created a custom OS profile, you can edit it at any time. You cannot modify a profile provided by UForge.

To edit your OS profile:

1. Under the `Administration` tab, click `OS Profiles`.

2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.

3. Click on the operating system you want to administer. This will provide the current list of profiles this operating system has.

4. Click on the profile you want to edit.

5. You can edit the name and description from the Overview page.

6. To modify the packages included in your profile, click OS Packages. The packages that you can add are listed in the top list. The bottom list is the list of packages that are currently part of the profile. Select the package(s) you want to add or remove to your OS profile and click `save`. You can search for the package you want to add.

7. Click `Save`.

Allowing Access to Image Formats

Image formats can be managed at the level of the organization or of the user.

An administrator who wants to add format rights to several users within an organization should create the formats at the level of the organization and then add them to a subscription profile. In this case, all new users created with the given subscription profile will have access to the formats.

In order to add a format you must create a target format and target platform.

Note: Without a target platform, the image cannot be published.

The administrator can add or remove image formats for a specific user account using the command-line interface, as described in *Adding Formats to a Specific User*.

Listing Formats

In UForge AppCenter, in order for a user to use a given format, a Target Format must be created and enabled for the given user and organization. Moreover, if the user wants to use a given Target Format for publishing an image, then a Target Platform must also be created, enabled and associated to the Target Format.

Note: As Target Platforms and Target Formats are created at the org level but then must be enabled for a given user, the list of available formats can differ for users of the same organization.

In order to list the formats available for a given *org*, use the command `org targetformat list`.

In order to list the formats available for a given *user*, use the command `user targetformat list`.

Adding Formats to an Organization Using the CLI

To add access to a format to an organization, you must:

1. Create a target format category using `uforge org category create`

2. Create target format using `uforge org targetformat create`

3. If the target format type is Cloud, create a target platform using `uforge org targetplatform create`. This target platform is necessary to publish image generated with the target format. The target platform will allow to create a cloud account associated

4. If the target format type is Cloud, add the target format to the target platform using `uforge org targetplatform addTargetFormat`.

5. (Optional) Add the format to a subscription profile using `uforge subscription targetFormat add`. To add a target platform use either:

   - `uforge subscription targetformat add`
   - `uforge subscription targetplatform add`

   Note: In order to force the changes to apply to all users (even those already created), use the option `--allusers`. For example:

   ```
   $ uforge subscription targetformat add --targetformat ovf qcow2 vbox --allusers --name sub --url
   ```

6. Enable the target format for the organization using `uforge org targetformat enable`.

7. Enable the target format using `user targetformat enable`.

Once the format is created in an organisation, you can either:

- add it to a group of users by adding it to the subscription profile (refer to *Adding Formats Using Subscription Profile*)

- add it to an individual user (refer to *Adding Formats to a Specific User*)

Adding Image Formats from the GUI

You can create your own image formats from the UForge GUI. To do so:

1. Create a target platform.

2. Create a target format.

Create a Target Platform

To create a target platform:

1. From the `Administration` tab, click on `Target Platforms`.

2. In the top right-hand, click on `new target platform`.

3. Enter the name of the target platform.

4. Select the type from the drop-down menu.

5. Optionally you can click on the plus (+) to add a logo.

6. If you do not want the target platform to be visible immediately, click on the check box next to `Enable` to deselect.

7. Click `create` in the top right to complete the creation.

Create a Target Format

To create a target format:

1. From the `Administration` tab, click on `Target Platforms`.

2. In the top right-hand, click on `new target format`.

3. From the drop-down menu, choose the target format category and click the next arrow button.

4. Enter the name of the target format.

5. Select the type and the image format from the drop-down menus.

6. Optionally you can click on the plus (+) to add a logo.

7. If you do not want the target format to be visible immediately, click on the check box next to `Enable` to deselect.

8. On the `Tooltips` page enter the Credentials, image and publish information.

9. On the `Target Platforms` page you can attach your target format to a target platform. To add the target format to a target platform, select the target platform from the button table and click the up arrow. Your target format will be attached to all the target platforms listed in the top table will be part of the part format you are creating.

10. Click next to complete.

Microsoft Windows and UForge

> **Warning:** UForge user must acquire Windows licenses in order to handle Windows OSes in UForge. When publishing Windows OS image or scanning Windows server, you have to confirm usage conditions of cloud provider and virtualization software which you publish to or scan.

Within UForge, Microsoft Windows is treated differently from other Linux/UNIX operating systems. In fact, Windows is not bundled with packages. Consequently, it is not possible to create standard (package based) OS Profile as for all the other supported distributions.

Instead, UForge uses a Golden Image as a profile. A Golden Image is an image that has been by the customer (see *Creating a Golden Image*) that contains the basic installation of the Windows version and some extra files. You can generate Golden Images at any time.

A Golden Image can be between 5 to 10 Gb, depending on the selected version.

You will need Golden Images to create Windows appliance templates. If you want to incorporate a Windows update, then you need to create and install a new set of Golden Images. You can create Golden Images yourself.

> **Note:** A good knowledge of Microsoft Windows is required to create your own Golden Images.

Generating all the profiles available (in one language) takes roughly 4 to 7 hours depending on the machine/network performance. You can regenerate Golden Images as often as you like, based on your individual needs. However, it is recommended that you regenerate only for specific updates—these updates will be in the Golden Image and you will not need to run package updates. When you generate a Golden Image the updates are the ones at the moment at which the Golden Image is generated.

Within UForge, the Golden Image used when you create appliances will be the last Golden Image created. In future releases, the different Golden Images will appear as Milestones.

Once the Golden Image is created, you will need to

1. Save it to the proper location (as described in *Storing Golden Images on the NAS*).

2. Store the golden images (all profiles in one language) as described in *Storing Golden Images on the NAS*. You will need about 40Gb of disk space on the UForge NAS.

3. Add the Golden Image to your UForge AppCenter, as described in *Adding a Golden Image to UForge AppCenter*.

Supported Microsoft Windows Versions

UForge supports Microsoft Windows Server 2008 R2 (this distribution is only 64 bits), 2012 and 2012R2.

Microsoft delivers 4 versions:

- DataCenter
- Standard
- WebServer
- Enterprise

For each version, there is a Full release and a Core release (without a Desktop).

These versions are available in French, English and Japanese. Unfortunately, one version of Microsoft Windows can NOT support multiple languages.

Restrictions

The following UForge features are not supported with appliances based on Microsoft Windows:

- Package selection at the OS level (however, users can add software via MySoftware or Projects)
- Windows migration possible as blackbox only
- Windows ISO format not supported

Using Microsoft Windows Key Mechanism

Microsoft Windows Operating System requires a key validation. UForge generates images without a key in it. Users have 30 days to enter their key once the Appliance is booted.

Workflow for Windows Golden Image

The following graphic illustrates the workflow to obtain and install Windows Golden Images.
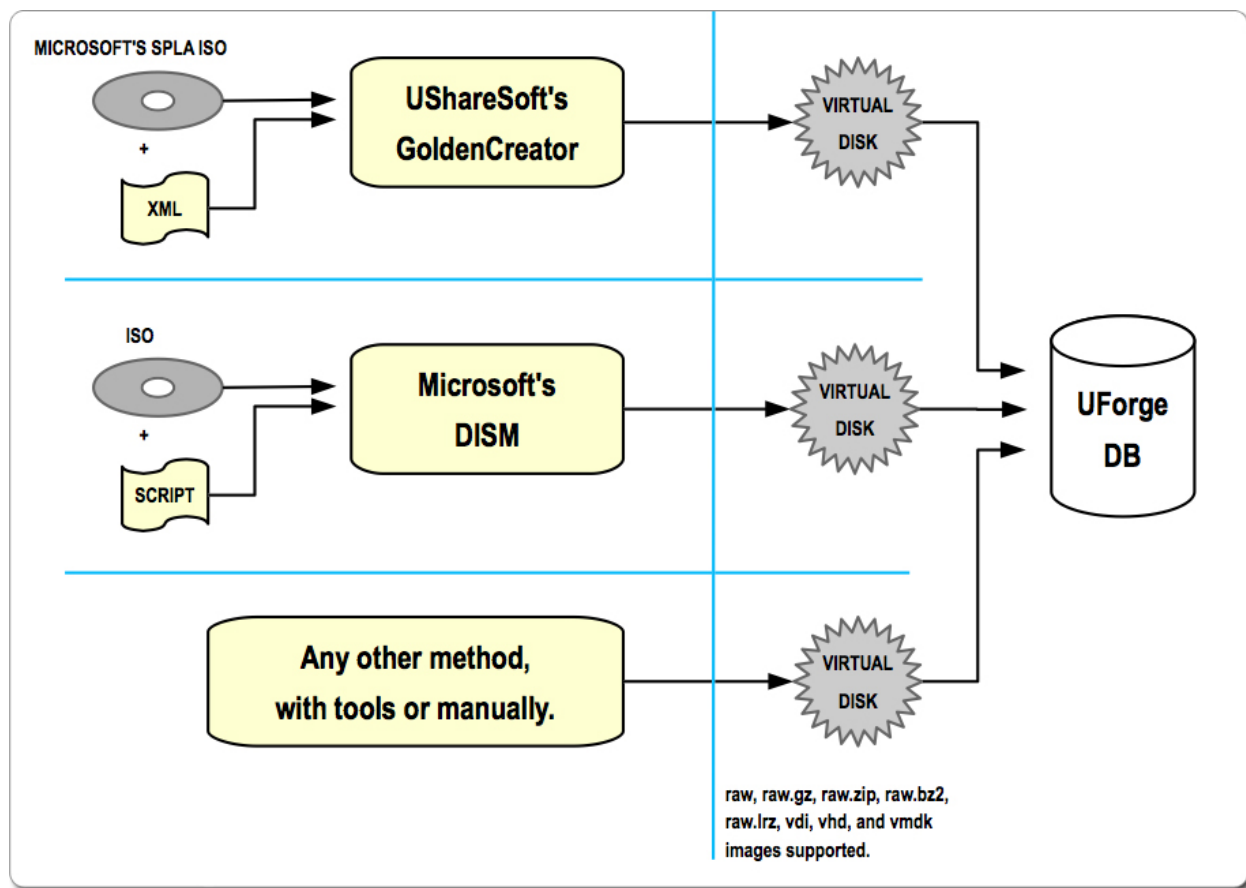
First Installation of Microsoft Windows

To install Microsoft Windows, follow the instructions below. This example is for Microsoft Windows Server 2008R2. If you wish to install for Microsoft Windows Server 2012 or 2012 R2, replace the string `Server2008R2` with the version you want to install in all commands below.

1. First population of Windows:

```
$ org os add --name Windows --arch x86_64 --version Server2008R2
```

2. Activate Windows Server2008R2 inside the UForge organization:

```
$ org os enable --name Windows --arch x86_64 --version Server2008R2
```

3. Activate Windows Server2008R2 for root user (and potentially other preexisting users):

```
$ user os enable --name Windows --arch x86_64 --version Server2008R2 --account root
$ user os enable --name Windows --arch x86_64 --version Server2008R2 --account <OTHER_USER_NAME>
```

Note: UForge does not manage Windows updates as it does for Linux. Each time you want to have an update for Windows, you will need to create and install a new set of Golden Images.

Listing Existing Golden Images

In order to view a list of existing golden images installed on your UForge run:

```
$ org golden list --name Windows --arch x86_64 --version Server2008R2
```

Creating a Golden Image

To create a new Golden Image, you will need to:

1. Ensure the following two partitions exist. These partitions are created by default during a standard Windows installation. There must be no other partitions.

   - System partition. This one is hidden, created automatically during installation of Windows Server.

   - Drive `C:`

2. The following Windows features must be installed as Administrator:

   - ServerCore-WOW64

   - NetFx2-ServerCore

   - NetFx2-ServerCore-WOW64

   - NetFx3ServerFeatures

   - NetFx3

   To install these features, you can either go to the `Control Panel` under `Programs and Features` or open a command prompt window as Administrator and run the following Windows commands:

```
$ start /w dism /online /enable-feature /all /featurename:ServerCore-WOW64
$ start /w dism /online /enable-feature /all /featurename:NetFx2-ServerCore
$ start /w dism /online /enable-feature /all /featurename:NetFx2-ServerCore-WOW64
$ start /w dism /online /enable-feature /all /featurename:NetFx3ServerFeatures
$ start /w dism /online /enable-feature /all /featurename:NetFx3
$ start /w dism /online /enable-feature /all /featurename:NetFx3-ServerCore
$ start /w dism /online /enable-feature /all /featurename:NetFx3-ServerCore-WOW64
```

   Note: If any of the above commands fail with an error indicating that the feature is non-existant, ignore the error and continue.

3. Install gtk-sharp-2.12.10.win32.msi.

   - You can download it from `http://download.mono-project.com/gtk-sharp/gtk-sharp-2.12.10.win32`

   - Rename `gtk-sharp-2.12.10.win32.msi.old` to `gtk-sharp-2.12.10.win32.msi`

   - Run the following command: `gtk-sharp-2.12.10.win32.msi`

4. We recommend that you run Windows Update to ensure that the latest updates are pre-installed in the Golden Image.

5. Optionally, you can also add the following customizations:

   - Modify the registry

   - Extra software installation

- User creation

6. Optionally, you can free several gigabytes of space by cleaning up windows updates installers.

> **Warning:** After this optimization you may not be able to uninstall some of the Windows updates.

```
$ dism /online /Cleanup-Image /StartComponentCleanup /ResetBase
```

7. If you have Service Packs installed, you can free up some space by executing the following command, which will merge the Service Pack installer to the operating system.

> **Warning:** After this optimization, you will not be able to uninstall the Service Pack.

```
$ dism /online /Cleanup-Image /SPSuperseded
```

8. You can optionally perform optimizations in size for the compressed raw virtual disk image. To do so, you must:

   (a) Before the sysprep step, use the Microsoft Sysinternals tool called sdelete.exe (or sdelete64.exe) with option -z in a command line for all partitions, example:

   ```
   $ sdelete -z C:
   ```

   (b) After finishing the golden image (after sysprep at the last step), but before compressing the .raw with gzip or lrzip, perform the following command to the .raw virtual disk image:

   ```
   $ cp --sparse=always image.raw newimage.raw
   ```

   This will copy the image file but skip the zeros, so the .raw image will be as sparse as possible, also helping the compression program.

   ```
   $ mv -f newimage.raw image.raw
   ```

9. For Windows 2008R2 create a file as follows. Note that the admin user name may be different depending on the environment. Please replace `Administrator` in the script with the appropriate one.

```
mkdir C:\Windows\Setup\Scripts
notepad C:\Windows\Setup\Scripts\SetupComplete.cmd
---
net user Administrator /logonpasswordchg:yes
---
```

10. For Windows 2012 and 2012R create a file as follows. Note that the admin user name may be different depending on the environment. Please replace `Administrator` in the script with the appropriate one.

```
mkdir C:\Windows\Setup\Scripts
notepad C:\Windows\Setup\Scripts\SetupComplete.cmd
---
@echo off
if not exist C:\etc\UShareSoft\no_console (
    net user Administrator /logonpasswordchg:yes
)
---
```

`changepasswd.bat` is specified in `Unattend.xml`. The script is launched only when the image has no console, just after `uforge-install-config` before displaying desktop.

```
        notepad C:\uforge\changepasswd.bat
        ---
        @if exist C:\etc\UShareSoft\no_console (
            @title Changing Administrator password
            echo Please provide new Administrator password.
            net user Administrator *
        )
        ---
```

11. Open a command prompt window as an administrator and go to the `%WINDIR%\\system32\sysprep` directory. Then run:

```
$ sysprep.exe /generalize /oobe /shutdown /unattend:c:\path-to-sysprep\Unattend.xml
```

Note: This will shutdown the machine. Do not boot the machine again!

12. You can now compress the golden images by running:

```
$ gzip image.raw
```

You can now save your golden image on the NAS.

Example of Unattend File for Windows 2008R2

The following is an example of an unattend file to be used when creating a golden image for Windows 2008R2.

```xml
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
    <settings pass="oobeSystem">
        <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyT
            <OOBE>
                <HideEULAPage>true</HideEULAPage>
                <NetworkLocation>Work</NetworkLocation>
                <ProtectYourPC>3</ProtectYourPC>
                <SkipUserOOBE>true</SkipUserOOBE>
            </OOBE>
            <UserAccounts>
                <AdministratorPassword>
                    <Value>Welcome@UShareSoft</Value>
                    <PlainText>true</PlainText>
                </AdministratorPassword>
            </UserAccounts>
        </component>
        <component name="Microsoft-Windows-International-Core" processorArchitecture="amd64" pub
            <InputLocale>0411:00000411</InputLocale>
            <SystemLocale>ja-JP</SystemLocale>
            <UILanguage>ja-JP</UILanguage>
            <UILanguageFallback>ja-JP</UILanguageFallback>
            <UserLocale>ja-JP</UserLocale>
        </component>
    </settings>
    <settings pass="specialize">
        <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyT
            <ProductKey>XXXXX-XXXXX-XXXXX-XXXXX-XXXXX</ProductKey>
            <ComputerName />
```

```
            </component>
            <component name="Microsoft-Windows-DNS-Client" processorArchitecture="amd64" publicKeyTo
                <DNSDomain />
                <UseDomainNameDevolution>true</UseDomainNameDevolution>
            </component>
        </settings>
        <settings pass="generalize">
            <component name="Microsoft-Windows-PnpSysprep" processorArchitecture="amd64" publicKeyTo
                <PersistAllDeviceInstalls>false</PersistAllDeviceInstalls>
                <DoNotCleanUpNonPresentDevices>false</DoNotCleanUpNonPresentDevices>
            </component>
        </settings>
    </unattend>
```

Example of Unattend File for Windows 2012 and 2012R2

The following is an example of an unattend file to be used when creating a golden image for Windows 2012 and 2012R2.

```
<?xml version="1.0" encoding="utf-8"?>
<unattend xmlns="urn:schemas-microsoft-com:unattend">
    <settings pass="oobeSystem">
        <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyT
            <OOBE>
                <HideEULAPage>true</HideEULAPage>
                <NetworkLocation>Work</NetworkLocation>
                <ProtectYourPC>3</ProtectYourPC>
                <SkipUserOOBE>true</SkipUserOOBE>
            </OOBE>
            <UserAccounts>
                <AdministratorPassword>
                    <Value>Welcome@UShareSoft</Value>
                    <PlainText>true</PlainText>
                </AdministratorPassword>
            </UserAccounts>
            <FirstLogonCommands>
                <SynchronousCommand wcm:action="add">
                    <CommandLine>c:\uforge\changepasswd.bat</CommandLine>
                    <Description>ChangeDefaultPassword</Description>
                    <Order>1</Order>
                </SynchronousCommand>
            </FirstLogonCommands>
        </component>
        <component name="Microsoft-Windows-International-Core" processorArchitecture="amd64" pub
            <InputLocale>0411:00000411</InputLocale>
            <SystemLocale>ja-JP</SystemLocale>
            <UILanguage>ja-JP</UILanguage>
            <UILanguageFallback>ja-JP</UILanguageFallback>
            <UserLocale>ja-JP</UserLocale>
        </component>
    </settings>
    <settings pass="specialize">
        <component name="Microsoft-Windows-Shell-Setup" processorArchitecture="amd64" publicKeyT
            <ProductKey>XXXXX-XXXXX-XXXXX-XXXXX-XXXXX</ProductKey>
            <ComputerName />
        </component>
        <component name="Microsoft-Windows-DNS-Client" processorArchitecture="amd64" publicKeyTo
```

```
                <DNSDomain />
                <UseDomainNameDevolution>true</UseDomainNameDevolution>
            </component>
        </settings>
        <settings pass="generalize">
            <component name="Microsoft-Windows-PnpSysprep" processorArchitecture="amd64" publicKeyTo
                <PersistAllDeviceInstalls>false</PersistAllDeviceInstalls>
                <DoNotCleanUpNonPresentDevices>false</DoNotCleanUpNonPresentDevices>
            </component>
        </settings>
    </unattend>
```

Storing Golden Images on the NAS

Each time you have a new Golden Image, you need to store them in the right NAS location.

Note: To store the golden images (all profiles in one language) you will need about 40Gb of disk space on the UForge NAS.

The golden images should be stored in:

```
Base dir = Windows/releases/Server2008R2/x86_64/
```

The path is:

```
{Language}/{Edition}/{Type}/{generation date}(YYYY-MM-DD)/goldenImagePathCompressedInGz
```

So for example:

```
Windows/releases/Server2008R2/x86_64/English/Standard/Core/2012-10-19/Windows_2008R2_Standa
```

Adding a Golden Image to UForge AppCenter

Once you have your Golden Image, you need to add it to your UForge AppCenter in order to be able to use the Windows version to create appliance templates. Your golden image must be in one of the following formats:

- raw.gz
- raw.zip
- raw.bz2
- raw.lrz
- vdi
- vhd
- vmdk

To add your Golden Image to UForge:

1. Copy the image to:

```
$ /tmp/DISTROS/Windows/releases/<windows os version>/x86_64/<language>/<my custom profile na
```

For example: /tmp/DISTROS/Windows/releases/Server2008R2/x86_64/English/MyProfile/Core/2014-04-28/Windows_2008R2_English_Datacenter_Core_2014-04-28.raw.gz

Note:

- File and directory ownership should be `tomcat:tomcat`.

- Permissions should be readable for all users

- Disk name must be unique in the `/tmp/DISTORS/Windows` file tree

2. You must ensure that the Windows distribution exists on your UForge AppCenter. If it does not, run:

```
$ uforge org os add --name Windows --arch x86_64 --version Server2008R2
```

3. In order to add the new golden image to the distribution, run:

```
$ uforge org golden create --name Windows --arch x86_64 --version Server2008R2 --edition Sta
```

Note: The parameters set when running `org golden create` should correspond to the path on the NAS, that is: {Language}/{Edition}/{Type}/{generation date}(YYYY-MM-DD)/goldenImagePathCompressedInGz

---

For example to install the golden image saved to the following path: `Windows/releases/Server2008R2/x86_64/English/Standard/Full/2012-10-19/Windows_2008R` you need to run:

```
$ org golden create --name Windows --arch x86_64 --version Server2008R2 --language English -
```

Managing the Project Catalog

A Project Catalog is a collection of software components that are available for UForge users to use when building their server templates. A project is one software component. A project includes files, binaries or native packages and may have an EULA (End User License Agreement).

Each organization within the UForge platform has a project catalog, and each member of the organization has access to the project catalog, allowing them to add any of the projects to their server templates. The contents of the project catalog, however, can only be updated and managed by the administrators of the organization where the project catalog belongs. An organization administrator can:

- add new projects to the catalog

- update existing projects to the catalog

- mark a project obsolete.

- delete a project.

You can also create custom OS profiles to include specific packages to an existing (standard) OS profile. See *Creating Custom OS Profiles*.

Adding a Project

To add a new project:

1. Under the Administration tab, click `Projects`.

2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.

3. Projects are associated with a specific version of operating system. Click on the operating system you want to create a project for. This will provide the current list of projects this operating system has.

Note: If a project supports more than one version of operating system, then you must re-create a new project for each operating system version. To automate the way projects are added and maintained, use the UForge APIs (for example to add the same project to multiple operating systems).

---

4. Click the `create` button in the `Projects` section of the page.

   Fill in the mandatory information including:

   - Name of the project

   - Version of the project

   - Project tag

   - Category

   - License type. Note if the software has a proprietary license, then choose `Custom`. This allows you to upload a custom license

   - Company information. All previous custom companies registered for projects are provided in the list. If the company of the project does not yet exist, choose any existing company and edit it afterwards.

5. Click `create`.

6. Enter a description.

7. Optionally, you can add the following meta-data:

   - a logo for the project

   - website information of where the project came from

   - in the case of a Custom license, upload the license file (HTML or text files)

   - set the default install path

   You can upload the software files that are required by the project. Files can be binaries, text files, jar files etc. The administrator can also choose native packages from the operating system itself to be part of the project.

8. Click `Save`.

Updating a Project

UForge provides the default projects for the OSes provided.

To modify the projects:

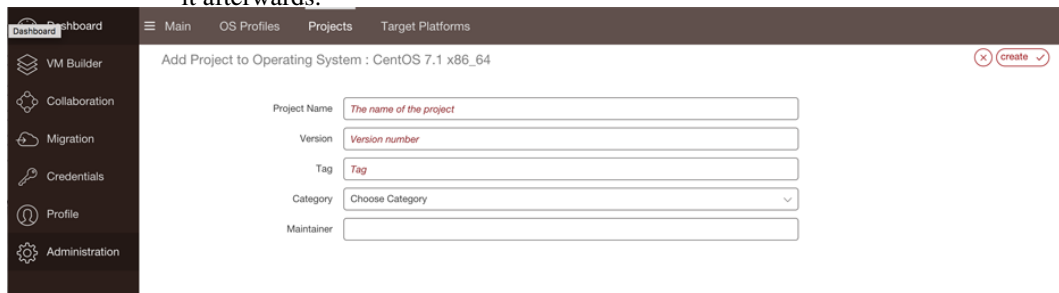1. Under the `Administration` tab, click `Projects`.

2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.

3. Projects are associated with a specific version of operating system. Click on the operating system you want to modify. You will see a list of all the projects that are part of the OS.

4. Click on the project you wish to edit. At this stage you can change any of the meta-data and files uploaded.

Deleting a Project

When deleting projects from the catalog, it is important to understand that you should check that no appliance templates are using the project you wish to delete (otherwise this may break generating a machine image for the appliance template).

Note: If an appliance template is still using a project, you can remove the project from the catalog by marking it as `obsolete`. This does not delete the project contents, however users cannot use the project for new appliance templates. See *Marking a Project as Obsolete* for more information.

To delete the project:

1. Under the `Administration` tab, click `Projects`.

2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.

3. Projects are associated with a specific version of operating system. Click on the operating system you want to modify. You will see a list of all the projects that are part of the OS.

4. To delete the project can either:

   click on the `delete` icon at the extreme right-hand side in the table for the project item.

   or

   click on the project item to edit it, then click on the `delete` icon at the top right-hand side of the project edit page.
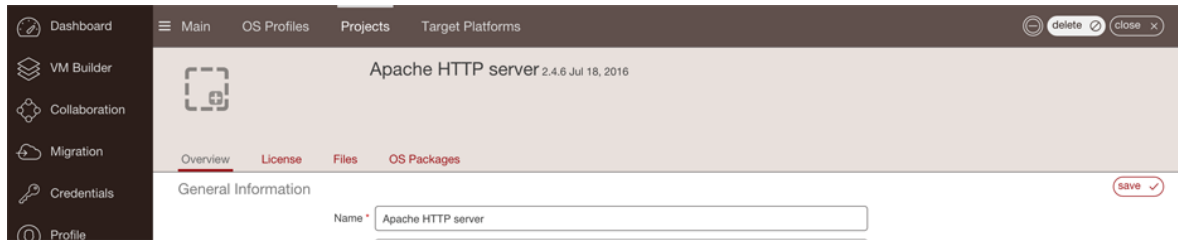
Marking a Project as Obsolete

Another way to remove a project from the project catalog is to mark the project as `obsolete`. This ensures that any existing templates that use the project can still generate images correctly, however the project is no longer accessible in the catalog for new appliance templates.

To mark a project as `obsolete`:

1. Under the `Administration` tab, click `Projects`.

2. If you are an administrator to more than one organization, then you can choose the organization to administer from the drop-down menu.

3. Projects are associated with a specific version of operating system. Click on the operating system you want to modify. You will see a list of all the projects that are part of the OS.

4. Click on the project to edit it.

5. Click on the `obsolete` icon at the top right-hand side of the project edit page.
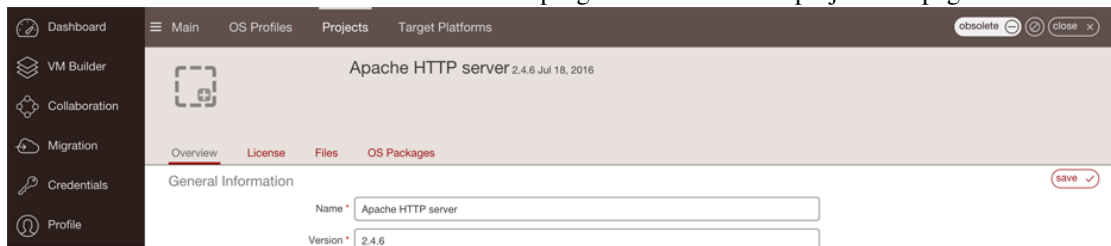
Creating and Managing Categories

Categories are used to organize projects. Users can use categories in order to filter projects in the appliance library.

Managing Categories with CLI

To create or manage categories with the CLI, use the command `uforge org`.

The available commands are:

- `category create`
- `category delete`
- `category list`

Creating and Managing Milestones

Milestones are used as a marker for a specific event, such as beta or GA for example. They appear on the GUI under OS package updates.
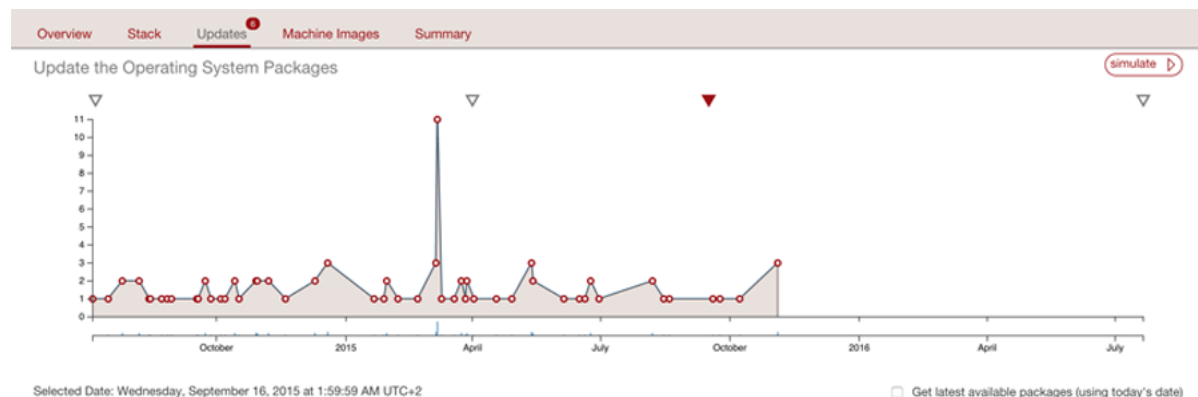
In the figure above, the triangle shape indicates a Milestone. For CentOS, this is the versions (6.1, 6.2 etc). For CentOS, Debian and RedHat, the major versions are automatically marked as Milestones when the distribution is added to the platform.

Milestones can be created by the UForge administrator using the command line interface, in order to mark a specific date or version, for example for a beta version or for a security fix.

To create or manage milestones using the CLI, use the command `uforge os`.

The available commands are:

- milestone add
- milestone list
- milestone modify
- milestone remove

In order to create a milestone called "beta" for Debian, run:

```
# uforge os milestone add --dname Debian --darch i386 --dversion 6 --date 2014-01-01 12:00:00 --name
```

Manage Users

The following sections cover information regarding managing the UForge users. Managing users is intrinsically linked to role based access control (RBAC). Please make sure to read the information on role-based access control in *RBAC Overview*.

Managing User Accounts

When managing user accounts, the administrator must have administration privileges for the organization where the user account is part of or where the member will be created.

Creating User Accounts with CLI

When a user account is created, the account is added to the specified organization. If no organization is specified, the UForge default organization is used. When creating a user, the administrator must specify a subscription code. This subscription code defines the roles, OSes, formats the user has access to, in addition to any quotas.

Note: Since a subscription code is required to create a user, the UForge administrator must first create these subscription profiles, and add the organization administrator (or another user) to the admin list.

To create a user account:

```
# uforge user create --account kermit --code <subscriptionProfileCode> --email kermit@usharesoft.com
```

As no default password is specified, UForge creates a temporary random password for the new user account. An email is automatically sent to the email address assigned to the user account when the account has been successfully created. The user account is automatically active, allowing the user to sign in and begin using UForge.

The `create` command provides a set of optional arguments allowing the administrator to control certain aspects of the account being created.

Disabling a User Account

Once disabled, the user no longer has access to the UForge platform.

To disable a user account:

```
# uforge user disable --account kermit --url https://uforge.usharesoft.com:443 -u $ADMIN -p $PASS
Getting user [kermit] ...
Success: User kermit has been de-activated
```

RBAC Overview

Access to the various services of the UForge platform is enforced by roles. Each role contains one or more entitlements that define the access to one or more services or permissions for various actions.

Viewing Default Roles

A set of default roles are provided by UForge. To view these default roles, use the command `uforge role list`:

```
$ uforge role list -u $ADMIN -p $PASS
Getting all the roles for the default organization ...
+------------------------+--------------------------------------------------------------------
| Name                   | Description
+------------------------+--------------------------------------------------------------------
| admin                  | Provides organization administration rights.  Includes managing the orga
|                        | user accounts.
+ Entitlements ----------------+--------------------------------------------------------------------
| marketplace_access     | used to determine whether to allow a user to interact with the marketpla
|                        | templates + template info, but voting, adding a comment, import, follow
| org_administrate       | Access to manage the entitlements list available for an organization.
| org_formats_administrate | Access to manage the generation formats available for an organization.
| org_members_administrate | Access to manage user accounts for an organization.
| org_os_administrate    | Access to manage the operating systems available for an organization.
| org_os_profiles_administrate| Access to manage the os profiles available for an organization.
| org_projects_administrate| Access to manage the project catalog for an organization.
| user_create            | Access to create user accounts in UForge.
+------------------------+--------------------------------------------------------------------
Found 15 role(s)
| apiuser                | Allows the user to communicate with UForge via the APIs.
+ Entitlements ------------+--------------------------------------------------------------------
| api_key_access         | Access to UForge APIs.
```

Organization administrators can create and manage roles as well as assign these roles to the members of the organization. For a list of all the commands possible to manage RBAC, use `uforge role --help`.

To view the entitlements associated to a role run the command `uforge role info --name <roleName>`.

Note: Adding a specific role to a user (such as Administrator) is often only the first step in granting specific rights. You must then specify the entity (for an Administrator this would be the organization) for which the user has the newly assigned rights. See the examples later in this section.

Listing Entitlements

An entitlement describes the right to perform an action on the UForge platform. These are pre-defined by UForge. Entitlements are added to roles to describe the access rights to the various UForge features.

To view all the entitlements provided by UForge, use the command:

```
$ uforge entitlement list -u $ADMIN -p $PASS
```

Listing Roles for an Organization

To view the available roles for an organization, run the command:

```
$ uforge role list --org usharesoft -u $ADMIN -p $PASS
```

If no organization is provided, then the default organization is used.

Creating and Updating User Roles

If the pre-defined roles delivered with UForge do not match with the permissions you want to allow users, you can either create a new role or modify an existing one.

To create a new role in the default organization:

```
$ uforge role create --name newrole --description "description of new role" -u $ADMIN -p $PASS
```

By default, this role will be empty (containing no entitlements). You can then add entitlements to a role by using the command:

```
$ uforge role entitlement add --name newrole --entitlements appliance_create studio_access -u $ADMIN
```

You can also remove an entitlement from a role by using the command:

```
$ uforge role entitlement remove --name newrole --entitlements studio_access -u $ADMIN -p $PASS
```

You can add entitlements as part of the creation by using the `--entitlements` option in the create command:

```
$ uforge role create --name newrole --description "description of new role" --entitlements appliance_
```

Note: You cannot modify the "root" role.

Listing Roles Assigned to a User

To view the roles already assigned to a specific user, run the command:

```
$ uforge user role list --account <username> -u $ADMIN -p $PASS
```

Adding Roles to a Subscription Profile

If you want a group of users to have the same role, then you can add it to the subscription profile (refer to *Managing Subscription Profiles*). This means that all the users that are created with this subscription profile will have this role. However, this will apply only to the new users created, unless you use the option `--allusers`.

```
$ uforge subscription role add --name sub --roles newrole --allusers -u $ADMIN -p $PASS
```

In the example above `sub` is the name of the subscription profile.

Adding a Role to a User

Note: Adding a specific role to a user (such as Administrator) is often only the first step in granting specific rights. You must then specify the entity (for an Administrator this would be the organization) for which the user has the newly assigned rights. See the examples later in this section.

To add a role to a user:

```
$ uforge user role add --account <username> --roles newrole -u $ADMIN -p $PASS
```

To add several roles to the same user:

```
$ uforge user role add --account <username> --roles role1 role2 -u $ADMIN -p $PASS
```

Deleting a Role from a User

To delete a role from a user:

```
$ uforge role delete --account <username> --roles roleA -u $ADMIN -p $PASS
```

Note: If this is the only role assigned to a user, once deleted, the user will no longer have any roles. With no roles the user will only be able to view the UForge account and dashboard.

Managing Subscription Profiles

A subscription profile defines the roles, OSes, formats the user has access to, in addition to any quotas. Since a subscription code is required to create a user, the UForge administrator must:

1. Create subscription profiles
2. Add the organization administrator to the admin list.

To create a subscription profile you must be the UForge administrator.

The subscription command allows you to create, manage and delete user accounts in UForge. Each user command can have mandatory and optional arguments to complete the request.

The subscription target has the following commands:

```
===========================================================================================================
Subscription_Cmd help
===========================================================================================================
admin                    | Manage subscription profile admins
create                   | Create a new subscription profile within an organization.
delete                   | Delete a subscription profile from an organization.
disable                  | Activates or enables a subscription profile within an organization.
enable                   | Activates or enables a subscription profile within an organization.
help                     | List available commands with "help" or detailed help with "help cmd"
info                     | Get detailed information on a subscription profile within an organiza
list                     | List all the subscription profiles for a given organization. If no
os                       | Manage subscription profile formats
quota                    | Manage subscription profile quotas
role                     | Manage subscription profile roles
targetformat             | Manage subscription profile target formats
```

```
targetplatform                  | Manage subscription profile target platforms
update                          | Updates an existing subscription profile.
```

Creating and Enabling Subscription Profiles

To create a subscription profile you must be a subscription profile administrator. In order to become a subscription profile administrator see *Allowing Administrators to Create Users*.

There are two steps to adding subscription profiles:

1. creating the profile

2. enabling the profile

By default, the subscription profiles you create are not active. You can activate the subscription profile when you create it by using the argument `--active`. Otherwise, you will need to activate it using the command `subscription enable`.

You can add the list of administrators that will be allowed to create users by using the argument `--admin`. If you do not want to define the administrators that will create the users at the same time as you create the subscription profile, you can do so later. See *Allowing Administrators to Create Users*.

When creating a subscription profile, you can also specify the following:

- formats (using targetformat and targetplatform)

- OS

- quotas

- roles

If you specify any of the above when you create a subscription profile, then all users created using this subscription profile will have the formats, OSes, quotas and roles defined in the subscription profile.

Note: Once you create a user with a specific subscription profile, even if you modify the subscription profile, the rights of the users already created will not be modified. For example, if profileA used to create UserA has quota set to unlimited. Once UserA is created, you modify the profileA to set quota to 3 generations. UserA will still have quota set to unlimited, but UserB created with the updated profileA will have quota set to 3 generations.

In order to force the changes to apply to all users (even those already created), use the option `--allusers`. This option can be used for roles, OS and format management in subscription profile. It cannot be used for quota.

1. To create a subscription profile for an organization, run the command:

```
$ uforge subscription create --code <string> --name <string> --org usharesoft --active -u $A
```

The code can be any alphanumeric string, excluding spaces and special characters.

2. To enable a subscription profile for an organization, run the command:

```
$ uforge subscription enable --name <string> --org usharesoft -u $ADMIN -p $PASS
```

Allowing Administrators to Create Users

In order to create a user, the following are required:

- The organization Administrator needs to be part of the list of subscription profile administrator. Only users that are part of this list can create user accounts.

- Subscription profile code. This code must be part of the request to create a user. Only the UForge administrator can create these codes.

When creating subscription profiles, the UForge administrator can add subscription profile administrators. However, they can also be added after the fact, as follows:

```
$ uforge subscription admin add --admin kermit --name profileA --org usharesoft -u $ADMIN -p $PASS
```

The argument `--admin` is the login of the user you want to add as an administrator. This administrator will be able to create users with the subscription profile specified by the argument `--name`.

Disabling a Subscription Profile

If you no longer want a subscription profile to be used when creating new users, you can either delete or disable the subscription profile. However, we recommend that you simply disable the subscription profile, in order to keep a history of the profile. Regardless of whether you delete or disable the subscription profile, the users created with the associated subscription code will not be deleted or deactivated.

To disable a subscription profile:

```
$ uforge subscription disable --name profileA -u $ADMIN -p $PASS
```

If no org is specified, the default organization is used.

Granting a User Administrator Rights

Note: The user you set as Administrator will have access to the Admin tab on UForge Portal and have rights to manage the organization. However, an Admin user does not automatically have the right to create users. In order to create new users the user must be part of the Admin list. See *Allowing Administrators to Create Users*.

There are two steps in granting a user Administrator privileges; you must:

1. Give the user the administrator role. This only indicates that the user CAN be an administrator, but does not specify for which organizations:

   ```
   $ uforge user role add --account <username> --roles admin -u $ADMIN -p $PASS
   ```

2. Assign the user as administrator to a specific organization. If no organization is provided, then the default organization is used:

   ```
   $ uforge user admin promote --account <username> --org <org name> -u $ADMIN -p $PASS
   ```

Setting Quotas

You can set a number of quotas for a user account using the command: `uforge user quota modify`.

Setting quotas allows you to limit the user's access to UForge based on one (or several) of the following criteria:

- number of appliances (includes imports and scans)
- number of images generated (includes all image generations)

- diskusage in bytes (includes storage of mysoftware uploads, bootscripts, image generations, scans)

- number of scans for migration (includes initial scan and incremental scans)

Note: You can set the size of the Scan Overlay. This is done not through the CLI but using the uforge.conf file. This is described in *Setting the Overlay Limit*.

You can set the quotas to refresh once a month using the argument `--frequency`. You can set the frequency to:

- `monthly`: the quota counter will be reset every month. The day of the reset is based on the date of the user creation (and not the date when the limit is set).

- `none`: once the quota is reached it will not be reset automatically (it can however be increased manually).

Viewing the Quotas for a User

You can see the quotas set for a given user as follows:

```
$ uforge user quota list --account <user> -u $ADMIN -p $PASS
```

In the example above, the argument `--user` is the account of the administrator. The argument `--account` is the user name of the account you want to view the quotas for.

Typically, when no limits are set, the results should be:

```
$ uforge user quota list --account kermit -u $ADMIN -p $PASS
Getting quotas for [kermit] ...
List of quotas available for [kermit] :
Scan (0)        -------------------UNLIMITED--------------------
Template (0)    -------------------UNLIMITED--------------------
Generation (0)  -------------------UNLIMITED--------------------
Disk usage (0B) -------------------UNLIMITED--------------------
```

Setting a Quota for Appliance

You can set a limit to the number of appliances a given user can have. This limit can be reset monthly. Note that if the user deletes an appliance, the count will go down. For example, if the user has reached the set limit of 10 appliances, the user can delete an appliance in order to create a new one and stay within the quota limit.

- The option `--type` must be set to `appliance`.

- The option `--limit` determines the quota.

For example to set the quota of appliances to 10 per month:

```
$ uforge user quota modify --user $ADMIN --password $PASS --account kermit --type appliance --limit 1
```

In the example above, the argument `--user` is the account of the administrator. The argument `--account` is the user name of the account you want to view the quotas for.

Setting a Quota for Image Generations

You can set a limit to the number of images a given user can generate. This limit can be reset monthly.

- The option `--type` must be set to `generation`.

- The option `--limit` determines the quota.

For example to set the quota of images a user can generate to 10 per month:

```
$ uforge user quota modify --user $ADMIN --password $PASS --account kermit --type generation --limit
```

Setting a Quota for Migration

You can set a limit to number of scans a given user can run. This quota includes both scan generation and scan appliance generation. This limit can be reset monthly.

- The option `--type` must be set to `scan`.

- The option `--limit` determines the quota.

For example to set the number of scans the user can run to 5 per month:

```
$ uforge user quota modify --user $ADMIN --password $PASS --account kermit --type scan --limit 5 --f
```

Setting the Overlay Limit

You can set the maximum size of the scan overlay. The overlay includes all the files in MySoftware and all other non-native files. This limit is set in the uforge.conf file. You must add the parameter:

```
UFORGE_SCAN_OVERLAY_MAX_SIZE = maximum size in octets
```

For example, to set the limit de 10G (10 x 1024 x 1024 x 1024):

```
UFORGE_SCAN_OVERLAY_MAX_SIZE=10737418240
```

Setting a Quota for Disk Usage

You can set a limit to the disk space a user can use. Disk space usage includes: mysoftware uploads, bootscripts, images generations, scans etc.

- The option `--type` must be set to `diskusage`

- The option `--limit` determines the quota in bytes. For disk usage, the quota is expressed in bytes.

For example to set the disk space quota a user can use to 10Gb per month:

```
$ uforge user quota modify --user $ADMIN --account user --type diskusage --limit 10737418240 --passwo
```

The results should be:

```
$ uforge user quota list --user $ADMIN --account <username> --password $PASS
Getting user [user] ...


+-----------------+-------------+-----------+-----------+---------------------------+
| Type            |    Consumed |     Limit | Frequency | Renewal date              |
+-----------------+-------------+-----------+-----------+---------------------------+
| appliance       |           1 | unlimited |           | -                         |
| diskusage       |         0.0 |     10 GB |           | -                         |
| generation      |           0 | unlimited |           | -                         |
| scan            |           0 | unlimited |           | -                         |
+-----------------+-------------+-----------+-----------+---------------------------+
Found 4 formats
```

Resetting Quotas

If you want to remove a quota set on a user, you can do this using the `--unlimited` flag. For example, to remove a quota limit you might have set on the number of scans for a user, run:

```
$ uforge user quota modify --user $ADMIN --account user --type scan --unlimited --password $PASS
```

Managing User Access to Operating Systems

Each user account can be configured to have access to certain operating systems and image formats that are enabled in the organization. When a user account is created, the organization's default operating systems and image formats are automatically added, thanks to the subscription profile (refer to *Managing Subscription Profiles*). The administrator can then add or remove operating systems and image formats for a specific user account using the command-line interface.

Listing Available OSes

To list all the operating systems the user has access to:

```
$ uforge user os list --account <username> -u $ADMIN -p $PASS
```

Allowing Access to OSes Using Subscription Profile

An administrator can modify access to OSes for a group of users using the subscription profile option `--allusers`. For example if a group of users with a specific subscription profile is created but you decide you want to modify the operating systems available, then you can modify the subscription profile. In order to force the changes to apply to all users (even those already created), use the option `--allusers`.

```
$ uforge subscription os add --name sub --os CentOS --version 6 --arch x86_64 --allusers -u $ADMIN -p
```

In the example above `sub` is the name of the subscription profile.

Adding or Removing Access to OSes for a Specific User

An administrator can add or remove access to operating systems for a specific user.

  • Adding/removing all CentOS versions:

```
    $ uforge user os enable --account <username> --name CentOS
    $ uforge user os disable --account <username> --name CentOS
```

  • Adding/removing CentOS version 5.8 all architectures:

```
    $ uforge user os enable --account <username> --name CentOS --version 5.8
    $ uforge user os disable --account <username> --name CentOS --version 5.8
```

  • Adding/removing CentOS version 5.8 i386:

```
    $ uforge user os enable --account <username> --name CentOS --version 5.8 --arch i386
    $ uforge user os disable --account <username> --name CentOS --version 5.8 --arch i386
```

Note: For "Scientific Linux" and "Red Hat Enterprise Linux", use the following syntax : Scientific.* RedHat.* for the distribution name.

Managing User Access to Formats

Image formats can be managed at the level of the organization or of the user.

An administrator who wants to add format rights to several users within an organization should create the formats at the level of the organization (refer to *Adding Formats to an Organization Using the CLI*) and then add them to a subscription profile (refer to *Adding Formats Using Subscription Profile*). In this case, all new users created with the given subscription profile will have access to the formats.

Adding Formats Using Subscription Profile

To add access to a format to a group of users, you can add it as part of a subscription profile. This means that all the users that are created with this subscription profile will have access to the format. You cannot add access to a format that is not included in the organization. In order to force the changes to apply to all users (even those already created), use the option `--allusers`. For a list of formats that are part of the organization, use the command:

- `org targetformat list`
- `org targetplatform list`

Therefore, in order to add formats using the subscription profile, run the following command. For example:

```
$ uforge subscription targetformat add --targetformat ovf qcow2 vbox --allusers --name sub --url http
```

In the example above, the argument `--name` is the name of the subscription profile.

Adding Formats to a Specific User

To add access to a format for a specific user, you must follow these steps:

1. Ensure the target format exists for the organization. For a list of formats that are part of the organization, use the command `org targetformat list`.

2. Enable the format for the user using `user targetformat enable`.

Enabling Access to a Format

Note: When enabling or disabling a format, you must use the "name" of the format and not the "format".

For example, if you are trying to enable the following format:

```
+----+--------------+--------------+----------+-------+----------------+--------+
| Id |     Name     |    Format    | Category | Type  | CredAccountType | Access |
+====+==============+==============+==========+=======+================+========+
| 34 | OpenStack VHD | openstackvhd | Cloud    | cloud | openstack       |        |
+----+--------------+--------------+----------+-------+----------------+--------+
```

You would use the following command:

```
$ user targetformat disable --targetFormats "OpenStack VHD" --account kermit --url https://uforge.us
```

Disabling User Access to Formats

To disable access to one or more formats for a specific user (in this example "kermit"), you must specify the format name with option --targetformats, as follows:

```
$ user targetformat disable --targetformats "OVF or OVA" QCOW2 VirtualBox --account kermit --url http
```

Granting a User API Access

Each user account can be configured to have access to the UForge APIs. This allows the user to automate the creation of appliances.

To use UForge APIs, an API key pair (public and secret keys) must be used as part of the communication. When a user account is enabled, an API key pair is created automatically. The administrator may also grant the user to have more than one set of API key pairs. This allows the user to create and manage multiple API key pairs.

To grant API access to a user account you must create a role to which you will assign the entitlement api_key_access.

1. Create the new role:

```
$ uforge role create --name newrole --entitlement api_key_access -u $ADMIN -p $PASS
```

2. Assign this new role to the user:

```
$ uforge user role add --name new role --account kermit -u $ADMIN -p $PASS
```

3. Optionally you can set the number of API keys:

```
$ uforge user api quota --apimax 5 --account kermit -u $ADMIN -p $PASS
Getting user [kermit] ...
Success: User kermit now has an API Key Quota of [5]
```

To disable API access simply remove the entitlement api_key_access:

```
$ uforge user role remove --name newrole --account kermit -u $ADMIN -p $PASS
```

Granting a User Supervisor Rights

If you want to allow a user to login to UForge as another user, you must grant the user supervisor rights. To do so, use the entitlement supervisor_access. You can either:

- add the entitlement to an existing role (note this will give supervisor rights to ALL the users with that role)
- create a new role with the supervisor access.

> **Warning:** Users with Supervisor Access will be able to log in as ANY of the users in the organization without entering a password. This right should be limited to support or managed services. Users with Supervisor Role needs to respect the privacy of the user data, according to current legislation.

It is probably safer to create a new "supervisor" role and add the supervisor rights, to limit the number of users that can access all the user accounts in the organization.

1. To create a new role, refer to *Creating and Updating User Roles*.

2. Add this new supervisor role to the user who will be acting as "supervisor"; see *Adding a Role to a User*.

3. If the user is not already an administrator, you will need to promote the user as an administrator of the organization for which supervisor rights are assigned:

```
$ uforge user admin promote --account kermit --org MyOrg -u $ADMIN -p $PASS
```

If no organization is provided, then the default organization is used. If you prefer to add supervisor access to an existing role (in this case "admin") run the following command:

```
$ uforge role entitlement add --name admin --entitlements supervisor_access -u $ADMIN -p $PASS
```

Monitoring Overview

The following sections cover some elements to monitor and troubleshoot your UForge Platform.

Viewing the Web Service Logs

All the web service logs can be found in the domain directory of the Tomcat application server. The web service uses the log4j logger. You can change the log level of the web service resources for debugging purposes.

To view the logs:

Log in to the web service node as root:

```
$ ssh root@<ip address of the node>
$ cd /var/log/tomcat/
$ tail -f catalina-daemon.out
```

To change the log level of the web service:

1. Log in to the web service node as root

```
$ ssh root@<ip address of the node>
$ cd /opt/Tomcat/webapps/ufws/WEB-INF/classes
$ vi log4j.properties
```

2. Update the logging level for each resource you wish by using the following keywords: `info|warn|debug`

3. Force Tomcat to reload

```
$ touch /opt/Tomcat/webapps/ufws/.reload
```

4. Restart the Web Service

```
$ service tomcat restart
```

Viewing Current Jobs in the Scheduler

To view the scheduler's current queue, log in to the oar scheduler node as root and run the command oarstat:

```
$ oarstat
Job id     Name          User           Submission Date     S Queue
---------- ------------- -------------- ------------------- - ----------
5725       4825          glassfish      2012-05-01 18:53:32 R default
```

This provides information on the jobs currently being executed as well as the jobs that are scheduled to be executed once a resource is free.

To view more details of a specific job, log in to the oar scheduler node as root and run the command oarstat and specify the JOB_ID:

```
$ oarstat --job 5725 --full
Job_Id: 5725
    job_array_id = 5725
    job_array_index = 1
    name = 4825
    project = [% 62, Installing distribution]
    owner = glassfish
    state = Running
    wanted_resources = -l "{type = 'default'}/resource_id=1,walltime=2:0:0"
    types =
    dependencies =
    assigned_resources = 51
    assigned_hostnames = vm
    queue = default
    command = /tmp/USER_DATA/FactoryContainer/images/4825/oar/ISO_4825.sh
    launchingDirectory = /tmp/USER_DATA/FactoryContainer/images/4825/oar
    jobType = PASSIVE
    properties = (nature=6) AND desktop_computing = 'NO'
    reservation = None
    walltime = 2:0:0
    submissionTime = 2012-05-01 18:53:32
    startTime = 2012-05-01 18:53:33
    cpuset_name = glassfish_5725
    initial_request = oarsub -d /tmp/USER_DATA/FactoryContainer/images/4825/oar -E oar_image_job
    message = FIFO scheduling OK
    scheduledStart = 2012-05-01 18:53:33
    resubmit_job_id = 0
    events =
```

Viewing the Logs of a Job

The main logs of OAR are stored in: /var/log/oar.log Each job launched on the OAR cluster, whether it be an image generation or publish to a cloud, logs are stored for the job. These include all the traces and error information during the execution of the job. Each job has a unique ID provided to it, which can be recuperated using the "oarstat" command as shown in *Viewing Current Jobs in the Scheduler*.

For jobs that generate an image, the log files are stored under: cd <user data mount point>/FactoryContainer/images/<job_name>

For example:

```
$ oarstat
Job id     Name          User           Submission Date     S Queue
---------- ------------- -------------- ------------------- - ----------
5725       4825          glassfish      2012-05-01 18:53:32 R default
```

The directory will be:

```
$ cd <user data mount point>/FactoryContainer/images/4825
```

Logs for jobs that publish an image to a specific cloud are stored in a sub-directory of the generated image directory. So for example if a user generates an Amazon image, then publishes the machine image to Amazon, the directory structure created is:

```
/<user data mount point>/FactoryContainer/images
                 |
                 |
                 |-- generated image logs dir --> 4825
                                          |
                                          |
                                          |-- published image logs dir --> publish_<job_name>
```

To view the logs of a job, log in to the oar scheduler node as root:

```
$ cd /<user data mount point>/FactoryContainer/images/<job_name>/oar
```

so for example

```
$ cd /tmp/USER_DATA/FactoryContainer/images/4825/oar
$ ls -al
total 376
drwxr-xr-x 2 glassfish glassfish   4096 Apr 30 18:21 .
drwxr-xr-x 6 glassfish glassfish   4096 Apr 30 18:22 ..
-rwxr-xr-x 1 glassfish glassfish    980 Apr 30 18:15 ISO_4825.sh
-rwxr-xr-x 1 glassfish glassfish   1088 Apr 30 18:15 cmd_4825.sh
-rwxrwxrwx 1 glassfish glassfish    300 Apr 30 18:18 oar_image_job4825.stderr
-rwxrwxrwx 1 glassfish glassfish 360500 Apr 30 18:21 oar_image_job4825.stdout
```

To check for suspicious jobs you can run:

```
$ oarnodes | grep -i suspected
```

Using Supervisor Mode

UForge allows you to login as another user if you have supervisor access. Supervisor access rights are assigned by the UForge administrator. Once you have supervisor rights, you will see glasses icon in the top right of the UForge GUI banner.

> **Warning:** Users with Supervisor Access will be able to log in as ANY of the users in the organization without entering a password. This right should be limited to support or managed services. Users with Supervisor Role needs to respect the privacy of the user data, according to current legislation.

To log in as another user with supervisor mode:

1. Click on the eye icon (`supervision`) in the top right of the UForge banner.

2. In the login screen, enter the name of the user you want to log in as. You will note that you do not need to enter a password.

Supervision

Enter the name of the user to supervise

User

(×) (supervise ◎)

UForge REST API in Supervisor Mode

The syntax for using basic authentication in supervisor mode is:

```
Authorization:Basic supervisorusername\supervisedusername:supervisorpassword
```

Getting Support

If you require support for your UForge platform, please contact:

> support [@] usharesoft [dot] com

In order to help us understand your issue. Please include any logs pertinent to the issue you are having.

To help gather information of the UForge platform, we have included a script you can run on the UForge platform instances.

To run the script as superuser:

```
$ /opt/UShareSoft/uforge/tools/support/machine_infos.sh
```

The script recuperates the logs from the main services that are running on the instance where the script is launched and creates an archive in `/tmp`, with the name starting `uforge-support-`. This archive can be sent to our support team as part of your support request.

UForge Tooling

To manage the UForge AppCenter, there are a number of tools available:

- The graphical user interface

- The command line interface

- Hammr

- The APIs

This section covers:

Using the CLI Tool

The UForge platform has a command-line tool called `uforge` to administer the platform. The CLI has the following usage:

```
uforge -u <name> -p <password> -U <URL> <target> <command> [options][args]
```

where the flags:

- `-u`: provides the username of the administrator

- `-p`: provides the password of the administrator

- `-U`: provides the UForge URL endpoint to communicate with the platform

If the `target`, `command` and arguments are ommitted, then you enter into an interactive mode e.g.:

```
$ uforge -u <user> -p <password> -U https://uforge.usharesoft.com/api
```

The following targets are available for the `uforge` command-line tool:

- `entitlement` : Administration of all the entitlements in UForge for RBAC

- `images` : Administer generated images for a user

- `org` : Organization administration (list/info/create/update/delete etc)

- `os` : Administer operating systems/distributions (list/info etc)

- `pimages` : Administer published images for a user

- `role` : Manage platform roles

- `subscription` : Manage subscription profiles : list profile, create profiles, update profiles

- `template` : Administer templates for a user (list/info/create/delete/generate/share etc)
- `user` : User's administration (list/info/create/update/delete etc)

To list the command for each target, use the `-h` or `--help` option. For example:

```
$ uforge -u root -p mypassword -U https://uforge.usharesoft.com/api user --help
```

The arguments are:

- `-U URL, --url URL` : the server URL endpoint to use
- `-u USER, --user USER` : the user name used to authenticate to the server
- `-p PASSWORD, --password PASSWORD` : the password used to authenticate to the server
- `-v` : displays the current version of the uforge-cli tool
- `-h, --help` : show this help message and exit
- `-k PUBLICKEY, --publickey PUBLICKEY` : public API key to use for this request. Default: no default
- `-s SECRETKEY, --secretkey SECRETKEY` : secret API key to use for this request. Default: no default
- `-c, --no-check-certificate` : Don't check the server certificate against the available certificate authorities

Rebranding Your UForge GUI

The following sections cover information regarding rebranding elements of the UForge GUI. Most customizations are done using the config.xml file located in: /var/opt/UShareSoft/uforge-client/gwt/uforge/templates. This is a relative path. Once you have completed your changes, you will need to run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Note: Other than the sections indicated in this document, we recommend that you do not modify other elements of the configuration files, otherwise this may cause issues when using UForge Portal.

Creating Dedicated Image Directory

If you plan to include your own logos and images when customizing Portal, you should create a sub-directory under: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/images`

For example, place all your logos and images under: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/i`

In all of the following sections, if you update logos and images, use your new path. For example:

```
<c:signInLogoUrl>images/signInLogo.png</c:signInLogoUrl>
```

Must be changed to:

```
<c:signInLogoUrl>images/myCompany/signInLogo.png</c:signInLogoUrl>
```

Modifying the Sign-In and Sign-Up Page

You can modify a few elements on the sign-in and sign-up pages.

The following figure shows the default Sign-in page.

Modifying the Logo

You can modify the logo that appears on the sign-in and sign-up page in section `<c:client>`:

```
<c:signInLogoUrl>images/common/fujitsu.svg</c:signInLogoUrl>
```

You can also modify the text that appears when you scroll over the logo, as well as the link the user is directed to when clicking on the link in section `<c:client>`:

```
<c:signInLogoTitle>Go to Fujitsu.com</c:signInLogoTitle>
<c:signInLogoLink>http://fujitsu.com</c:signInLogoLink>
```

Modifying the Title

You can modify the title that appears in the Sign In and Sign Up pages by updating the following attribute in section `<c:client>`:

```
<c:signInProductName>FUJITSU Software&lt;br/&gt;UForge</c:signInProductName>
```

Modifying the Sign Up Landing Page

You can modify the page you are directed to when you click on "Need an account? Sign up", in section `<c:client>`:

```
<c:externalSignUpUrl>http://www.usharesoft.com/products/signup</c:externalSignUpUrl>
```

This allows you to create a dedicated external sign up page outside this user interface so you can customize the way users are created on the platform.

Removing the Sign Up Page

If you wish to hide the text `Need an account?  Sign up` then change the following attribute to `false`:

```
<c:showSignUp>false</c:showSignUp>
```

Removing User and Password Caching

By default, the Sign In page allows the user to click a check box to remember previous sign in credentials. This is the check box labelled `Remember Me`. if you wish to remove this checkbox from the Sign In page, then update the following attribute to `false`:

```
<c:showRememberMe>false</c:showRememberMe>
```

Hiding Option to Reset Password

You can choose to hide the option that allows the user to reset the password from the sign in page. To do so set the option to "false" in section `<c:client>`:

```
<c:showChangePassword>false</c:showChangePassword>
```

This is useful if you have your own authentication mechanism and do not want UForge users to be able to change the password in UForge, since their credentials are managed somewhere else.

Note: When setting this value to `false`, this also hides password management for the user on the `User Profile` section of the user interface.

Modifying the Signed In Header

You can modify several elements of the header that is displayed when the user is signed in. This includes:

- The logo displayed on the right-hand side
- An external URL link and tooltip (title) when the logo is clicked
- The title displayed on the left-hand side

The following figure displays the default signed in header.

For colours and layout modifications, please refer to *Customizing the CSS*.

Modifying the Title

You can modify the title displayed in the top left-hand corner of the header by updating the following attribute under the `<c:client>` section:

```
<c:headerProductName>UForge AppCenter</c:headerProductName>
```

Modifying the Logo

You can modify the displayed signed in logo by updating the following attribute under the `<c:client>` section:

```
<c:headerRightLogoUrl>images/common/fujitsu.svg</c:headerRightLogoUrl>
```

You can also modify the text that appears when you scroll over the logo, as well as the link the user is directed to when clicking on the link by updating the following attribute respectively:

```
    <c:headerRightLogoTitle>Go to Fujitsu.com</c:headerRightLogoTitle>
<c:headerRightLogoLink>http://fujitsu.com</c:headerRightLogoLink>
```

Adding Links to the Header

You can add links that will be displayed in the header to external resources.

The following example shows how to add a link to a blog under the `<c:header>` section:

```
    <c:header>
    <c:linkItem>
            <c:title>Blog</c:title>
            <c:link>https://blog.usharesoft.com/</c:link>
    </c:linkItem>
</c:header>
```

Customizing the Footer

UForge Portal allows you to customize the information that appears in the footer. The config.xml file provides two sections to modify this info:

- `<c:client>`
- `<c:footer>`

The version number is automatically provided as part of the initial configuration and any update of the platform.

For colours and layout modifications, please refer to *Customizing the CSS*.

Modifying the Copyright

To modify the copyright under `<c:client>`:

```
<c:copyright>Copyright © 2007-2014</c:copyright>
```

Adding Links to the Footer

You can add links to external resources under the `<c:footer>` section.

The following example shows how to add a link to twitter:

```
<c:footer>
        <c:linkItem>
            <c:title>twitter</c:title>
            <c:icon>images/common/icons/twitter.svg</c:icon>
            <c:link>https:twitter.com/usharesoft</c:link>
        </c:linkItem>
</c:footer>
```

You can add as many links as you like by adding a `linkItem` for each link.

Adding Terms of Use or Privacy Policy

If you wish to add a "Terms of Use" or "Privacy Policy", you MUST include the UShareSoft Terms of Use and Privacy Policy as part of your own.

To modify the Terms of Use or Privacy Policy, go to the sections under <c:footer> and enter the path to the new terms of use and/or privacy policy. You can also modify the link text if you wish:

```
<c:footer>
        <c:linkItem>
            <c:title>terms of use</c:title>
            <c:link>https//www.usharesoft.com/about/terms-of-use.html</c:link>
        </c:linkItem>
        <c:linkItem>
            <c:title>privacy policy</c:title>
            <c:link>https//www.usharesoft.com/about/privacy-policy.html</c:link>
        </c:linkItem>
</c:footer>
```

Restricting Change Password

If you wish to remove the ability for a user to change their password in the `User Profile` section of the user interface, then update the following attribute in the `<c:client>` section to `false`:

```
<c:showUserProfile>false</c:showUserProfile>
```

Restricting User Profile Usage

If you wish to manage user information (email, etc) outside the user interface, then you can restrict the user profile to `read only`.

```
<c:editUserProfile>false</c:editUserProfile>
```

To hide completely the `User Profile` section of the user interface, update the following attribute to `false`:

```
<c:showUserProfile>false</c:showUserProfile>
```

Restricting Formats

UForge Portal allows users to generate the templates provided to all the formats the user has access to. In certain circumstances, you may wish to restrict the formats shown to the user. To restrict the available machine image formats in the UI you rquire to update the `config.xml` file.

To disable a format in the UI configuration, find the `<t:target>` section of the machine image format you wish to disable, then either add or change the value of the `<t:enabled>` tag. For example if you wish to deactivate OpenStack, then the following changes should be made to the configuration file:

```
<t:target>
    <t:id>openstack</t:id>
    <t:name>OpenStack</t:name>
    <t:enabled>false</t:enabled>
    <t:visible>true</t:visible>
</t:target>
```

If you wish to hide a format completely, then update the `<t:visible>` tag to `false`.

Note: Restricting formats for specific users can also been done through RBAC in the platform.

Restricting the Cloud Accounts

When using the UForge Portal, all the cloud account types are displayed. You can restrict the cloud accounts that are visible by updating the config.xml file of the UI. To hide a specific cloud format, find the `<t:account` section of the account type you wish to hide, then update the `<t:visible>` tag to `false`. For example, if you wish to remove the CloudStack account type, then the following changes should be made to the configuration file:

```
<t:account>
    <t:id>cloudstack</t:id>
    <t:type>cloudstack</t:type>
    <t:name>Cloudstack</t:name>
    <t:visible>false</t:visible>
</t:account>
```

Customizing the CSS

You can customize all the CSS elements of Portal. You can modify a number of elements like text color, background color, font type and size using the custo.css file located in: `/var/opt/UShareSoft/uforge-client/gwt/uforge/templates/css`.

Customizing the Platform

You can customize some elements of your UForge Portal using the config.xml file located in: `/etc/UShareSoft/uforge/uforge.conf`. This is a relative path.

Configuring UForge Sign Up Information

To modify the UForge sign up information, you can modify the following variables in the uforge.conf file.

- `UFORGE_REGISTRATION_USER`
- `UFORGE_REGISTRATION_CODE`
- `UFORGE_REGISTRATION_PASSWORD`

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Configuring Email Notification Sender

To modify the sender of automatic emails generated from UForge, you can modify:

- `UFORGE_REGISTRATIONS_EMAIL`

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge/tools/update_scripts/uforge_update.sh
```

Modifying UForge WebService URL

To modify the UForge webservice URL, you can modify the following variable in the uforge.conf file:

    UFORGE_URL

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Modifying UForge Portal Application Root Context

To modify the UForge root context of the UForge Portal, you can modify the following variable in the uforge.conf file:

    UFORGE_UI_ROOT_CONTEXT

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Modifying UForge API Root Context

By default the root context for the UForge API is set to `/api`. To change the UForge API root context, modify the following variable in the uforge.conf file:

    UFORGE_API_ROOT_CONTEXT

To apply the changes you made, run the following command. This will stop Tomcat, integrate the changes and restart Tomcat:

```
$ /opt/UShareSoft/uforge-client/bin/uforge_ui_update.sh
```

Troubleshooting

If you have errors with your Portal once you have applied the changes, please do the following:

1. Check the syntax of the customization files.

2. Check `/var/log/tomcat/catalina-daemon.out` for error reports.

Changelog

3.6-fp2

Release Date: 2016-12-05

New Features

- Fujitsu K5 support. Can now register machine images generated on the platform to Fujitsu K5.

  > **Note:** The following operating systems are supported for the moment (others will be supported soon):
  >
  > - CentOS 7.0
  > - Ubuntu 14.04

- SELinux support when creating appliance templates and during migration
- Docker machine image generation support. This allows users to build docker base images.
- When scanning Windows machines, the scan report now includes the services detected.

  > **Note:** The platform does not support the comparison of windows-based scans at this time.

RFEs

- Better progress status when scanning Windows machines
- Less restrictive validation of website information in the MySoftware/Project Overview
- New icons for 'pull' and 'upload' for software/project files management
- Added directory icon when displaying all the files for software/project files view
- When deleting a folder, the confirm message should be more explicit (that all sub folders and files will also be deleted)
- Better explanation of the "cached" option for software/project files in the UI

- Managing licenses for software/project components; there is now an explicit delet button to remove an uploaded license file

Bug Fixes

- 6123 Publishing a generation from a scan results in 500 error in UI

- 6089 Member's role on workspace couldn't be changed if language is set as French or Japanese

- 6017 Canceling from Appliance Create no longer returns to previous page

- 5946 Publishing to CloudStack fails with the next error: vhd.gz: No such file or directory

- 5942 RHEL is added despite launching *org os add* for Oracle Linux or Scientific Linux with cli

- 5909 User ID and group ID of the install profile can be set 0

- 5906 UserResourcesAccessRights database mapping not proxied

- 5896 Deployment fails due to NIC settings

- 5892 Deployment fails when using eth1

- 5843 "org category delete" raises an error

- 5777 Launching uforge-scan.exe from command prompt fails with an error if the file path to the binary includes Japanese characters.

- 5762 Cannot register the third disk with a VirtualBox image

- 5756 New users, the defaukt country is: Abkhasia

- 5754 opening the Dashboard > Generations page first shows progress bar for all publications

- 5752 Number of MySoftware components not properly refreshed in the UI

- 5750 Number of Appliance not properly refreshed in the UI

- 5748 The diskusage of "uforge user quota list" is displayed by byte

- 5684 Invite the same user in the collaboration members list does not show error message

- 5676 Duplicated variable in /etc/default/grub if distribution provides default values.

- 5647 Keyboard and kernel parameters are not taken into the scan report on CentOS 7 scan.

- 5635 Broken incremental scan for windows 2012R2

- 5627 Cancelling scan via ctrl+c is not correctly displayed in the UI

- 5625 uforge-scan does not respect bandwidth limit

- 5623 When the image of CentOS7 is generated, RPM-GPG-KEY-CentOS import read fails

- 5621 rpmgen fails to build package if file path in %file includes space.

- 5570 Impossible to delete an incremental scan

- 5562 UForge CLI accesses to interactive mode even if the user or password are wrong

- 5560 The input value of the activation key is not saved in a Windows appliance

- 5342 Scan incremental with Ubuntu does not appear in UI

- 5265 No dialog box displayed after running an instance on Azure

3.6-fp1

Release Date: 2016-10-31

New Features

- Import/Export of appliance templates in the user interface
- **Software (MySoftware) and Project bundles now consolidated. New features added including:**
    - pulling files from remote locations (HTTP, FTP endpoints) so the user no longer requires to upload software components to the platform
    - pulling files can be cached for future generations or pulled on each generation
    - file permissions added for files and directories
    - can create directory structures in a software bundle
    - can add tagging information to a software bundle
    - can add native packages from OS respositories to a software bundle
    - can add boot scripts to a software bundle
    - identify the software bundle only being supported on a subset of operating systems
- API keys can be used for authentication when running a scan for migration.
- Scan messages and error messages cleaned up and more understandable
- Japanese language localization for the UI

Bug Fixes

- 5293 Image generation error: Windows image must have at least 512 MB of memory
- 5729 Issues with migration from 3.5.1. to 3.6
- 5465 Build fails due to unreachable rpm-4.11.2.tar.bz2
- 5740 Fix DB schema checks
- 5331 AWS publish no longer works
- 5637 Windows generation from scan fails at boot
- 5427 Unable to generate a virtual machine with LVM inside a MSDOS disk
- 5291 All combo boxes are empty when a value has been selected
- 5876 Logo broken on Dashboard
- 5444 Unable to populate Fedora/RHEL distributions
- 5420 When a template is remobed from a workspace, a DELETE error appears in the log file
- 5527 Subscription info does not list the frequency of quotas
- 5494 Scan fails because of files of type c (character device file)
- 5483 The service command not found in a machine generated by UForge
- 5442 The file deletion of Project fails

- 5429 Root can disable root account in UForge CLI

- 5746 Timeout of 10 seconds for the UForge CLI is not usable

- 5563 Internal error in Migration tab

- 5558 500 Call Fail Error when generating an image from scan

- 5556 The targetformat of Amazon is not displayed when generating an image

- 5553 If a scan is deleted, the image generated from the same scan is not deleted

- 5551 Spelling mistake in UI when publishing to Flexiant

- 5549 The error of Keystone version is displayed in Keystone Server URL

- 5403 Scan fails when trying to rebuild a non repo package

Trademarks

UForge is a registered trademark of UShareSoft, a Fujitsu company.

LINUX is a registered trademark of Linus Torvalds.

Microsoft and Windows are either registered trademarks or trademarks of Microsoft Corporation in the United States and/or other countries.

Oracle, GlassFish, Java, and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle Corporation and/or its affiliates.

Apache Ant, Ant, and Apache are trademarks of The Apache Software Foundation.

Red Hat Enterprise Linux is a trademark of Red Hat.

MySQL and the MySQL logo are the servicemarks, trademarks, or registered trademarks owned by Oracle Corporation Inc.

UNIX is a registered trademark of the Open Group in the United States and in other countries.

Other company names and product names are trademarks or registered trademarks of their respective owners.

Copyright FUJITSU LIMITED 2017

High Risk Activity

The Customer acknowledges and agrees that the Product is designed, developed and manufactured as contemplated for general use, including without limitation, general office use, personal use, household use, and ordinary industrial use, but is not designed, developed and manufactured as contemplated for use accompanying fatal risks or dangers that, unless extremely high safety is secured, could lead directly to death, personal injury, severe physical damage or other loss (hereinafter "High Safety Required Use"), including without limitation, nuclear reaction control in nuclear facility, aircraft flight control, air traffic control, mass transport control, medical life support system, missile launch control in weapon system. The Customer shall not use the Product without securing the sufficient safety required for the High Safety Required Use. In addition, FUJITSU (or other affiliate's name) shall not be liable against the Customer and/or any third party for any claims or damages arising in connection with the High Safety Required Use of the Product.

Export Restrictions

Exportation/release of this document may require necessary procedures in accordance with the regulations of your resident country and/or US export control laws.