

Neue m23-Version veröffentlichen

Integrierte Hilfe übersetzen

- Alle neuen und geänderten Texte auf Deutsch fertigstellen (/m23/inc/help/* and /m23/inc/i18n/*)
- Mit /mdk/doc/manual/bin/checkForMissingHlp.sh überprüfen, ob bei den **Sprachdateien Änderungen** zwischen der stabilen und der (aktuellen) Entwicklerversion vorliegen oder ob es neue Dateien gibt. Erzeugt automatisch /tmp/m23-i18n-help-translate.tar.gz, in der alle neuen und geänderten Sprachdateien enthalten sind. Das **Archiv versenden**.
- Die neuen **Übersetzungen herunterladen** und **einfügen** mit
/mdk/m23helper/gitCloneAndMeldTranslations
In *meld* darauf achten, daß auch **neue Dateien** berücksichtigt werden.
- Zum Überprüfen, ob alle \$I18N_-Variablen vorhanden sind
/mdk/doc/manual/bin/checkForMissingi18n.sh
ausführen.

Handbuch erstellen

- Check, if all HTML entities are in the HTML to LaTeX translation index. Run:
/mdk/doc/manual/bin/checkForMissingHTML2LatexEntities.sh
- Zum Überprüfen, ob **alle Bildschirmfotos** vorhanden sind
/mdk/doc/manual/bin/checkMissingScreenshots.sh
aufrufen. Die fehlenden Bildschirmfotos in **allen Sprachversionen** erstellen.
- Erstellung starten: */mdk/bin/menuStart => doc => manual*
- Sprache auswählen: *lang => all*
- Die **LaTeX**-Dateien erstellen: *text*
- **PNGs optimieren** (wenn Bildschirmfotos geändert oder hinzugefügt wurden):
optimisePNGs
- **PDF**- und **HTML**-Version des Handbuches erstellen: *pdf-html*
- **Hochladen**: *upload*
- **Zurück** zum Dokumentationsmenü

„Development guide“ erstellen

- Ggf. neue LaTeX-Dateien in `/mdk/doc/devguide/devguide.tex` **einbinden**
- API-Dokumentation erstellen: `devguide`
- **PDF-** und **HTML-**Version erstellen: `generate`
- **Hochladen:** `upload`

Von Entwickler- zu Veröffentlichungsversion wechseln

- `/mdk/bin/menuStart => fork => moveDevel2Release`

Pakete bauen

- `/mdk/bin/menuStart => debs => build`
- Ggf. **m23-ucs-extra-*.deb** neu bauen. Kommando:

`/mdk/m23Debs/bin/quickBuild.sh m23-ucs-extra`

Paket liegt anschließend unter `/mdk/ucs/debs`

Lokales Testen: UCS, Debian und Raspbian

- **UCS VM (Lokales Testrepo):** Diesen Schritt für jede UCS-Version mit einer eigenen VM wiederholen. Zum Installieren der m23-Pakete auf der jeweiligen VM **vom m23-Entwicklungssystem**

`/mdk/ucs/installLocalm23DebsOnUCS`

aufrufen. Das Skript **fragt interaktiv** nach dem letzten Teil der **IP**.

Hierbei werden auch gleichzeitig die “unmaintained” Debian-Pakete als Archiv (`ucs-x.y-unmaintained-m23-ab.c.tar.gz`) in das aktuelle Verzeichnis heruntergeladen. Die **Archive** nach

`t5:/var/www/ucs`

hochladen.

- **Debian/RaspBian:** Das Skript

`/mdk/m23helper/addLocalm23ServerRepository`

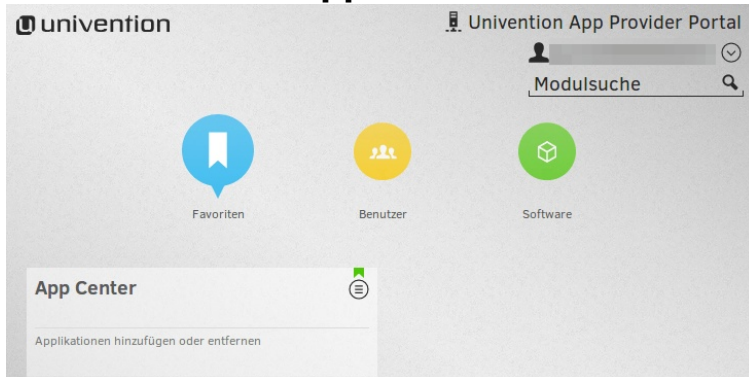
auf das Zielsystem kopieren, ausführen und anschließend m23 per APT installieren.

Hochladen der Debian-Pakete

- Die Debian-Pakete hochladen: `/mdk/bin/menuStart => debs => directuplinst`
- Warten, bis die Mail mit dem Upload-Ergebnis eingetroffen ist

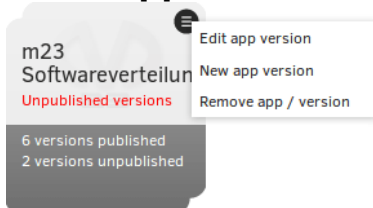
Hochladen ins Univention Testrepo

- Einloggen auf: <https://selfservice.software-univention.de/univention-management-console/>
- Auf **Software** -> **Apps** klicken



- Beim Bewegen des Mauszeigers über das Symbol der eigenen App (im Beispiel "m23 Softwareverteilung") wird ein Icon eingeblendet, über das, nach einem Klick, Werkzeuge zum Editieren, Anlegen und Löschen von (neuen) App-Versionen aufgelistet werden.

“New app version” auswählen



- Diesen Schritt für die neue m23-Version für **alle UCS-Versionen** durchführen.

Bei der **UCS-Version** immer **dieselbe Nummer** wie beim vorherigen m23-Release als **Quelle** (“Source UCS version”) verwenden, es sei denn, es gibt eine neue UCS-Version. Dann die vorherige UCS-Version verwenden.

- Die jeweilige UCS- (“Source UCS version”) bzw. m23-Vorgängerversion (“Source App version”) als Quelle und die nächsthöhere als Ziel (“Target UCS version” bzw. “Target App version”).

Als **Versionsschema** für die App-Version die m23- und die UCS-Version (ohne Punkt) verwenden. Z.B. “18.1-42” für m23 18.1 auf UCS 4.2.

Neues UCS, neues m23	Altes UCS, neues m23
Add a new version UCS 4.2 <input type="button" value="v"/> Source UCS version 18.1-42 <input type="button" value="v"/> Source App version UCS 4.3 <input type="button" value="v"/> Target UCS version 18.1-42 <input type="button" value="v"/> Target App version ABBRECHEN CREATE	Add a new version UCS 4.2 <input type="button" value="v"/> Source UCS version 18.1-42 <input type="button" value="v"/> Source App version UCS 4.2 <input type="button" value="v"/> Target UCS version 18.2-42 <input type="button" value="v"/> Target App version ABBRECHEN CREATE

- Das Skript `t5:/var/www/ucs/ucs-upload` **anpassen** (ggf. neue UCS-Versionen hinzufügen und alte löschen)

- Ggf. **neue Version des Selfservice-Skriptes** herunterladen:

curl <https://provider-portal.software-univention.de/appcenter-selfservice/univention-appcenter-control> > /bin/univention-appcenter-control

m23-App aus dem Univention-Testrepo installieren

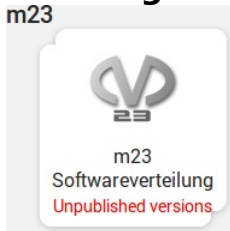
- Die UCS-VMs für das Testen der Univention-Testrepos starten
- Ggf. auf den VMs das **Univention-Testrepo freischalten** mit den Befehlen aus

```
/mdk/ucs/addUCSTestRepository
```

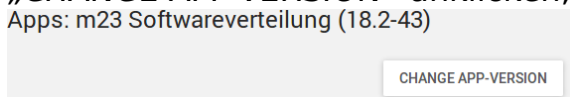
- Anschließend die **m23-App** über die UCS-Oberfläche **installieren**. Dabei darauf achten, daß die **m23-Versionnummer stimmt**.

m23-App freigeben

- **m23-Logo** anklicken



- Überprüfen, ob schon die richtige **m23-** und **UCS-Version** ausgewählt ist, ansonsten „CHANGE APP-VERSION“ anklicken, um die **richtige Version** zu **wählen**.



- „PUBLISH APP“ zum **Freigeben** anklicken.
- Die beiden Punkte für alle freizugebenden Versionen **wiederholen**.
- Anschließend über das Burgermenü (rechts oben) **abmelden**.

Serverinstallations-ISO und VirtualBox-Appliance

- Ggf. **Dateien für das initrd** des ISOs heruntergeladen (wenn noch nicht vorhanden, arch=x86): */mdk/bin/menuStart => clientISO => RFSdownload*
- **ISO erstellen:** */mdk/bin/menuStart => serverISO => iso*
- Neue **VM in VirtualBox** erstellen (Linux/Debian, 1024 MB Ram, wachsende VDI-Festplatte mit 16GB)
- Einstellungen ändern: Netzwerk auf **Netzwerkbrücke** stellen, Audio deaktivieren oder auf „**Null Audiotreiber**“ ändern
- Die VM starten und das ISO verwenden
- Server installieren mit den Einstellungen:
 - Sprache: English
 - Servername: m23s
 - root-Paßwort: test
 - IP: 192.168.1.23
 - Gateway: 192.168.1.5
 - DNS: beibehalten
 - Partionierung und Formatierung: Automatisch
- Anschließend die VM neu von der **Festplatte booten**
- Leere Bereiche von der Festplatte mit 0 überschreiben

/mdk/bin/prepareOSForCompression

- **VM herunterfahren**
- VM **exportieren** (Dateiname m23server_xx.y_rock.ova)
- OVA **komprimieren:** („7zm“ entspricht „7zr -t7z -m0=lzma -mx=9 -mfb=64 -md=32m -ms=on“)

7zm a m23server_xx.y_rock.7z m23server_xx.y_rock.ova)

- **OVA hochladen:**

/matrix23/arbeit/iso/ovf/up-m23-frs m23server_xx.y_rock.7z

- **ISO hochladen:**

cd /mdk/server

./up m23server_xx.y_rock.iso

Raspberry Pi

- **SD-Karte** mit Raspbian **beschreiben** (**SSH muß aktiviert sein**, feste IP).
- Nach dem Booten per **SSH einloggen** (Benutzer: pi, Paßwort: raspberry bzw. test)
- Ggf. **zusätzliche Konfiguration**:

sudo raspi-config

- *Expand Filesystem*
 - *Advanced Options => Memory split: 16 MB*
 - *Update aufrufen*
 - Neu starten
- **m23** mit den Befehlen aus

/mdk/m23helper/copyPasteTestm23ServerInstall

installieren.

- **Leere Bereiche** von der SD-Karte **mit 0 überschreiben**:

/mdk/bin/prepareOSForCompression

- Pi **herunterfahren**
- **Komprimiertes Abbild** der SD-Karte erstellen:

/matrix23/arbeit/RaPi/img/sichern-mmcbk0-dd-7z m23server_xx.y_rock-RaPi.7z

Entspricht: *dd if=/dev/sdX | 7zr a -t7z -m0=lzma -mx=9 -mfb=64 -md=32m -ms=on -si m23server_xx.y_rock-RaPi.7z*

- */matrix23/arbeit/iso/ovf/up-m23-frs m23server_xx.y_rock-RaPi.7z*

m23 Online-Demo aktualisieren

- Die VM mit der **m23-Appliance** starten (IP: 192.168.1.23=)
- Auf dem **Entwicklersystem** die Datenbank und Demodateien **exportieren** und automatisch per scp kopieren:

root: */mdk/m23helper/m23admin-offline-copy/export-dataset.sh*

- Auf der **VM** Datenbank und Demodateien **importieren**:

root: */mdk/m23helper/m23admin-offline-copy/import.sh*

- Auf dem **Entwicklersystem** die **statischen HTML-Seiten** für die Online-Demo von der VM **herunterladen** und automatisch auf SourceForge **hochladen**:

`/mdk/m23helper/m23admin-offline-copy/dlhttrack`

CMS

- **Ankündigungsartikel** schreiben
- **Übersetzen** lassen
- Die **Linkdateien** für ISO, OVA und RasPi-Abbild **erstellen**

`/mdk/doc/CMSLinkCreator.sh <ISO, OVA oder RasPi-Abbild>`

- Änderungen in Bazaar **einchecken**
- CMS-Dateien **hochladen**:

`cd /matrix23/arbeit/wwwTests
./cms-upload`

Newsletter

- Den **Newslettertext** mit

`/mdk/doc/newsletterGenerator3.sh`

erstellen, **anpassen** und **verschicken**.