Neue m23-Version veröffentlichen

Drucken: /mdk/doc/devguide/m23-Release-Tabelle.odt

ÜBERSETZEN: Integrierte Hilfe

TODO: Auf neuen git-Workflow umstellen.

- 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.

ERSTELLEN: Benutzerhandbuch

- 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/checkForMissingScreenshots.sh* aufrufen. Die fehlenden Bildschirmfotos in **allen Sprachversionen** erstellen.
- Erstellung starten: */mdk/bin/menuStart => doc => manual*
- Sprache auswählen: *lang => all*
- Die **LaTeX**-Dateien erstellen: *tex*
- Neue Seiten in /mdk/doc/manual/tex/??/not-generated/manual.tex eintragen
- **PNGs optimieren** (wenn Bildschirmfotos geändert oder hinzugefügt wurden): *optimisePNGs*
- **PDF**- und **HTML**-Version des Handbuches erstellen: *pdf-html*
- **Hochladen**: *upload*
- **Zurück** zum Dokumentationsmenü

ERSTELLEN: m23-autoTest-Dokumentation

- Ggf. zum Erstellen und Hochladen: cd /mdk/autoTest/doku; *./mkPDFandHTML*

ERSTELLEN: Development guide

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

WECHSELN: Entwickler- => Veröffentlichungsversion

- */mdk/bin/menuStart => fork => makeRelease*
  - *fork =>switchRelease*

ERSTELLEN: m23-Serversoftware-Pakete

- */mdk/bin/menuStart => debs => build*
- Ggf. **m23-ucs-extra-*.deb** neu bauen (liegen anschließend unter */mdk/ucs/debs)*:

  cd */mdk/m23Debs/bin; ./quickBuild.sh m23-ucs-extra*
- Die **Pakete** werden automatisch nach *t5:/var/www/ucs* **hochgeladen**.

VORBEREITEN: Raspberry-Pi-SD-Kartenabbild

- **SD-Karte** mit Debian-ARM64 **beschreiben** (**SSH muß aktiviert sein**, feste IP).
  - Zum Anpassen eines Debian-ARM64-Abbildes siehe in der **m23-autoTest**-Dokumentation im Abschnitt „(Optional) Raspberry Pi“
  - Neues **getestetes Debian-ARM64** (https://raspi.debian.net) herunterladen und xz-Datei entpacken
  - .../RaPi/bin/RasPi-Image-setStaticIP+enable-SSH <*.img>
  - .../RaPi/bin/wiederherstellen-auto-dd-gz <*.img.gz>

VORBEREITEN: m23-autoTest, bei neuer Debian- oder UCS-Version?

- Test-VM erstellen: /mdk/autoTest/doku/m23-autoTest.pdf, */mdk/autoTest/autoTestScriptGenerator.php* **anpassen** und **ausführen**

TESTEN: Lokales Testen: Debian-x86 + Debian-ARM64 (RasPi)

- /mdk/autoTest/aTS-Debian*.sh und */mdk/autoTest/aTS-Pi.sh* ausführen.

TESTEN: Lokales Testen: UCS

- */mdk/autoTest/aTS-UCS*-LokalesTestrepo.sh* ausführen.
  Hierbei werden auch gleichzeitig die "**unmaintained**" Debian-Pakete als Archiv (ucs-x.y-unmaintained-m23-ab.c.tar.gz) auf der jeweiligen VM unter /tmp angelegt.
- Die **Archive** (von **allen UCS-Versionen**) nach *t5:/var/www/ucs* **hochladen**.

ERSTELLEN + TESTEN: m23-Server-Installations-ISO

- Ggf. **Dateien für das initrd** des ISOs herunterladen (wenn noch nicht vorhanden, arch=x86): */mdk/bin/menuStart => clientISO => RFSdownload*
- **ISO erstellen**: */mdk/bin/menuStart => serverISO => iso*
  - ISO wird **automatisch** auf den **Virtualisierungsserver** nach /crypto/iso/ **kopiert**.
  - *„M23SERVER_ISO"* wird **automatisch** in */mdk/autoTest/autoTestScriptGenerator.php* angepaßt.
  - *ggf. autoTestScriptGenerator.php* **ausführen**
- **Testen:** */mdk/autoTest/aTS-atISOm23Server.sh*

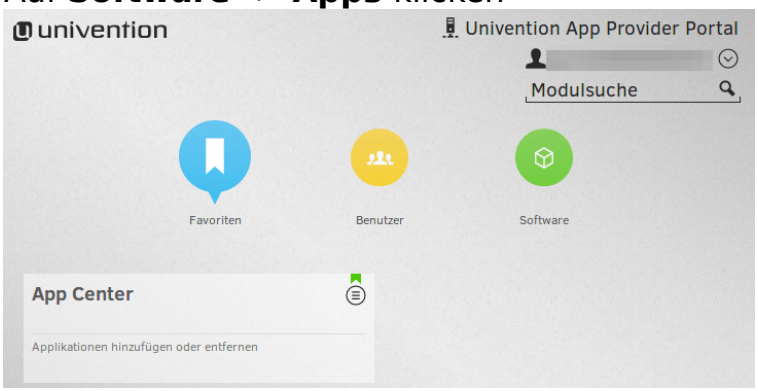
HOCHLADEN: m23-Serversoftware-Pakete

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

TESTEN: Hochladen ins Univention-Testrepo

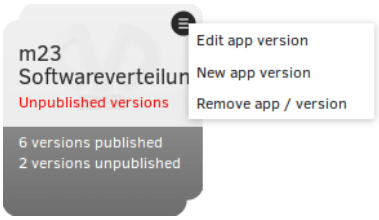
- Einloggen auf:
  https://selfservice.software-univention.de/univention-management-console/

  Auf **Software -> Apps** klicken



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

  "New app version" auswählen

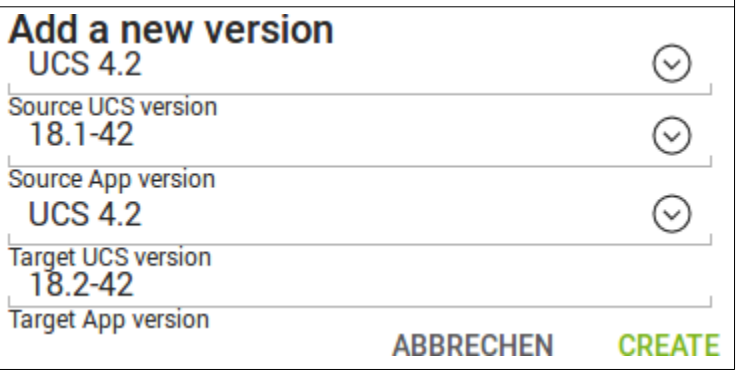


- Diesen Schritt für die neue m23-Version für **alle UCS-Versionen** durchführen.

  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 (unten 18.**2**-4**3**)	**Altes** UCS, neues m23
	

- Ggf. **neue Version des Selfservice-Skriptes** herunterladen: /var/www/ucs/updateSelfServiceScript

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

- Das Skript *t5:/var/www/ucs/ucs-upload* **anpassen** (ggf. neue UCS-Versionen hinzufügen und alte löschen) und **ausführen**. Hierbei werden passende ucs-x.y-unmaintained-m23-ab.c.tar.gz gleich mit hochgeladen

TESTEN: m23-App aus dem Univention-Testrepo installieren

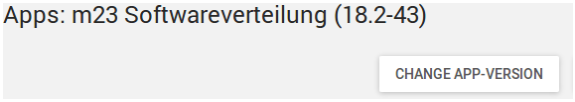
- */mdk/autoTest/aTS-UCS*-UCS-Repo.sh* ausführen.

FREIGEBEN: m23-UCS-App

Wenn die Installations**tests erfolgreich** waren:

- **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 Pfeilmenü (rechts oben) **abmelden**.

ERSTELLEN: Serverinstallations-ISO => VirtualBox-Appliance

- */mdk/autoTest/aTS-OVAfromISO.sh* => Erstellt /crypto/iso/*m23server_xx.y_rock.7z* auf dem Virtualisierungsserver (OVA-Datei im 7Zip-Archiv)
- *m23server_xx.y_rock.7z* auf das **Entwicklersystem** herunterladen
- Hochladen:
  - **OVA**: */matrix23/arbeit/iso/ovf/up-m23-frs m23server_xx.y_rock.7z*
  - **ISO**: */mdk/server/up m23server_xx.y_rock.iso*

    *Unter **Chromium** bei **SourceForge als Standard** setzen (Files, Home/m23, ISO (i) anklicken, Default Download, alle anhaken)*

ERSTELLEN: Raspberry-Pi-SD-Kartenabbild

- **m23** */mdk/autoTest/aTS-PiSDVorbereiten.sh*
  - Hierdurch werden **leere Bereiche** auf der SD-Karte **mit 0 überschrieben** und der Pi **heruntergefahren**.
- **Komprimiertes Abbild** der SD-Karte erstellen: */matrix23/arbeit/RaPi/img/sichern-auto-dd-7z m23server_xx.y_rock-RaPi*
- ***Umbenennen + Verschieben:*** *mv m23server_xx.y_rock-RaPi.dd.7z /matrix23/arbeit/iso/ovf/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*
- **Hochladen:** */matrix23/arbeit/iso/ovf/up-m23-frs m23server_xx.y_rock-RaPi.7z*

ERSTELLEN: m23 Online-Demo

- Die VM mit der **m23-Appliance** starten (IP: 192.168.1.23)
- Ggf. **neue** Demodaten erstellen: 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

ERSTELLEN: CMS

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

  /mdk/doc/CMSLinkCreator.sh <ISO, OVA oder RasPi-Abbild>
- CMS-Dateien **hochladen**:

  cd /matrix23/arbeit/wwwTests;./cms-upload
- **gh-Seite**: Artikel erstellen und veröffentlichen

ERSTELLEN: Newsletter

- Den **Newslettertext** mit

  /mdk/doc/newsletterGenerator3.sh

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