

# Deployment System #2

scriptable

# Use Cases

- Neues Frontend hinzufügen
- Update auf neue FESA Version (ohne Subsets)
- Update auf neue FESA Version (mit Subsets)
- Update auf neue FESA Version (inkl. Composition, Association)
- Anpassung FESA Klassendesign
- Konfigurationsmanagement

# Use Case: Neues Front-End

- asl cluster:
  - Benötigte NFS-Verzeichnisse und Initialisierungsskripte anlegen mit Skript 'prepfesaenv'
- FESA:
  - Neuen FEC für deploy-unit anlegen (fesa3 -nf ...)
  - Deploy-unit code Synchronisierung (fesa3 -s ...)
  - Neue Instanziierungsdatei anlegen (fesa3 -ni ...)
  - Konfiguration anpassen (human)
  - FESA Software ausrollen (fesa3 -cp ... o. fesa3 -r ...)

# Use Case: Update auf neue FESA Version (ohne Subsets)

- Auschecken/Updaten von Quellen aus SVN (FESA Klasse/DU, ggfs. Treiber wie FESL)
- FESA
  - Klasse (n):
    - FESA FWK Version erhöhen (fesa3 -u ...)
    - Design Version erhöhen (fesa3 -pv ...)
    - Design validieren (fesa3 -v ...)
    - Source Code synchronisieren (fesa3 -s ...)
    - Optional: source code anpassen (human)
    - Bibliothek erstellen (make)
    - Design in DB einfügen (fesa3 -ex ...)

# Use Case: Update auf neue FESA Version (ohne Subsets)

- Deploy-Unit:
  - FESA FWK Version erhöhen (fesa3 -u ...)
  - Design Version erhöhen (fesa3 -pv ...)
  - Versionen inkludierter Klassen erhöhen (fesa3 -in ...)
  - Design validieren (fesa3 -v ...)
  - Source Code synchronisieren (fesa3 -s ...)
  - Optional: source code anpassen (human)
  - Binary erstellen (make)
  - Design in DB einfügen (fesa3 -ex ...)

# Use Case: Update auf neue FESA Version (ohne Subsets)

- Instanziierungsdatei:
  - FESA FWK Version erhöhen (fesa3 -u ...)
  - Design Version der deploy-unit erhöhen
  - Design validieren (fesa3 -v ...)
  - optional: neue Mockup Instanziierungsdatei erstellen (fesa3 -m ...)
  - Instanziierungsdatei in DB exportieren (fesa3 -ex ...)
- Deploy-Unit:
  - Binary und ihre Konfigurationsdateien ausrollen (fesa3 -cp ...oder fesa3 -r ...)
  - optional: Binary mit Mockup Instanziierungsdatei ausrollen

# Use Case: Update auf neue FESA Version (ohne Subsets)

- Änderungen in SVN repository zurück schreiben 'commit'
- Ideal: SVN Tag erstellen

# Use Case: Anpassung FESA Klassendesign

- Auschecken/Updaten von Quellen aus SVN (FESA Klasse/DU, ggfs. Treiber wie FESL)
- FESA
  - Klasse:
    - Design anpassen (z.B. neue Property, neue/andere Parameter, etc) (human)
    - 
    - Design validieren (fesa3 -v ...)
    - Source Code synchronisieren (fesa3 -s ...)
    - Optional: source code anpassen (human)
    - ... analog use case 'update FESA FWK version'

# Use Case: Anpassung FESA Klassendesign

- Instanziierungsdatei:
  - Design Version der deploy-unit erhöhen
  - Promote: neue Konfigurationsparameter mit default-Werten in Instanziierungsdatei automatisch übertragen (fesa3 -p ...)
  - Konfigurationsparameter bearbeiten (human)
  - Design validieren (fesa3 -v ...)
  - optional: neue Mockup Instanziierungsdatei erstellen (fesa3 -m ...)
  - Instanziierungsdatei in DB exportieren (fesa3 -ex ...)
- Änderungen in SVN repository zurück schreiben 'commit'
- Ideal: SVN Tag erstellen

# FESA CLI Tool

- Verfügbar seit Juli 2015
- Erlaubt Anwendung von typischen FESA Operationen in Shell-Skripten
- Aktuellste Informationen im Wiki:  
<https://www-acc.gsi.de/wiki/FESA/FESA3CLITool310>
- Verwendet diesselben Bibliotheken wie FESA Eclipse Plug-In
- Kann im asl cluster von allen Anwendern in der Gruppe 'fesa' verwendet werden

# Beispiele

- **Validierung**

- fesa3 -v ~/workspace/PowerSupply/src/PowerSupply.design
- fesa3 -v  
~/workspace/PowerSupplyDU/src/PowerSupplyDU.deploy
- fesa3 -v  
~/workspace/PowerSupplyDU/src/test/asl733/DeviceData.instance

- **Code Synchronisierung**

- fesa3 -s ~/workspace/PowerSupply/src/PowerSupply.design

- **Update FESA FWK Version**

- fesa3 -u 3.1.0 ~/workspace/PowerSupply/src/PowerSupply.design

# Beispiele

- **Update FESA Design Version**
  - `fesa3 -pv t++ ~workspace/.../*.design | *.deploy`
- **Build ( Kompillieren inkl. FESA Abhängigkeiten)**
  - `fesa3 -fb x86_64 ~workspace/.../*.design | *.deploy`
- **Neues Frontend anlegen**
  - `fesa3 -nf <FEC hostname> ~workspace/.../*.deploy`
- **Neue Instantiierungsdatei anlegen**
  - `fesa3 -ni ~workspace/.../*.deploy`
- **NEU: DB Export von FESA Design Dokumenten**
  - `fesa3 -ex ~workspace/.../*.design | *.deploy | *.instance`

# Beispiele

- **UPDATED: Erzeugung von Subsets**
  - `fesa3 -sub ~workspace/.../* .design | * .deploy | * .instance`
- **NEU: Erzeugung von Mockup Instanzen**
  - `fesa3 -m ~workspace/.../* .instance`
- **NEU: Injizieren von Klasseninformationen in Deploy-Units**
  - `fesa3 -in ~workspace/.../* .design ~workspace/.../* .deploy`
- **NEU: "GSI Release" von Deploy-Units** (reiner Kopiervorgang aller relevanten Dateien für eine DU auf einem FEC, überspringt SVN repository check, Source Code Synchronisierung, Kompillieren, DB Export)
  - `fesa3 -cp ~workspace/.../* .deploy scuxl...`

# Beispiele

- **Anzeige aller verfügbaren Kommandos:**
  - fesa3 -h

# Vorteile

- Viele FESA Arbeitsschritte sind skriptbar
- Für jede FESA Software lässt sich eine zugeschnittene Lösung zur Automatisierung finden
- Skripte sind auch für Kollegen hilfreich (-> z.B. als Anleitung, in der Rufbereitschaft, ...)

# Nachteile

- Isb. FESA CLI Tool produziert zuviel unwesentlichen Spring log output
- Unübersichtlichkeit, wenig Visualisierungsmöglichkeiten
- Keine automatisch zu verarbeitende Rückmeldung über Ergebnis der Aktion!

# Zu klären

- Wie lassen sich Informationen von DB Abfrage Skripten mit Build and Deployment Skripten sinnvoll verknüpfen?
- Sollen FESA Software Binaries an zentraler Stelle gesammelt werden um sie von dort zu verteilen? Oder soll jeder selbst dafür sorgen dass die betreffenden Frontends den passenden Softwarestand haben?
- FEC's einrichten via Skript 'prepfesaenv' – ausreichend?
- Wie Aktualisierung der Datenbank sicherstellen?
- Wie verändert sich der Workflow für die FESA Entwickler?
- Wo werden die Build-Skripte im SVN repository abgelegt?
  - z.B. im jeweiligen deploy-unit Projekt?
- Wo werden die Build-/Datenbank-Skripte für Anwender verfügbar gemacht?