DLS Change Management for OPS/MVS

Automating the Automation

Freddy Sonnemans

Managing Director

DLS Systems bvba

Belgium

DLS Change Management for OPS/MVS

- **Personalia :**

- 1970 – 1971 : Sidmar NV (Dev Assember/Sysprog)
- 1972 – 1976 : Continental Bank (Dev Assember/PM)
- 1976 – 1990 : ABN Bank Belgium (IT Mgr/Sysprog/PM)
- 1990 – 1992 : Goal Systems/Legent (Tech Mgr)
- 1992 – 1997 : Sapiens (Tech Mgr)
- 1997 - : DLS Systems bvba
  - » CA Partner on Automation & System Mgt products

DLS Change Management for OPS/MVS

- **Unicenter CA-OPS/MVS Event Management and Automation**

- CA’s Automation tool for zOS
- Based on Rexx Rules/Rexx Procedures
- Rules Stored in RuleSets (= PDS)
- Rexx Procedures stored in PDS
- Rules executing synchroniously (in requesting AS)
- Procedures executing in TSO-like servers

DLS Change Management for OPS/MVS

CMfOPS

- **Reason for Development ?** 1

- Experience/Frustration in the field
- Problems :
  - Do more with less (people)
  - Growing number of lpars
  - Managebility of lpars

DLS Change Management for OPS/MVS

CMfOPS

- **Reason for Development ?** 2

- Results :
  - An extremely powerfull tool becomes dangerous
  - Tools that should help become a burden
  - Automation is abended

DLS Change Management for OPS/MVS

CMfOPS

- **Choice of language for project :**

- Rexx vs Assembler
- Parent product : primarily Rexx
- TSO/ISPF Environment
- OBVIOUS CHOICE

DLS Change Management for OPS/MVS

CMfOPS

- **Concepts :**

- Strict seperation of development and production environments
  - TESTRULE vs RULES libraries
  - TESTSUBF vs SUBF libraries
  - TESTREXX vs REXX libraries
- Enforce standards
  - Only rules should be in Rulesets
  - Subroutine/Functions should be in separate libraries
  - Rexx procedures should be in REXX libraries
- All rules/procedures should work everywhere

DLS Change Management for OPS/MVS

CMfOPS

- **Concepts :**

- Assured delivery to all lpars in scope
  - Delivery is independent of availibilty of lpar(IPL, Shutdown, Lost connection)
  - Delay (Freeze/Unfreeze) mechanism for production lpars

- Control activation of rules across lpars

DLS Change Management for OPS/MVS

CMfOPS

- **Concepts :**

- Deployment of Rules :

  - Disable Rule
  - Backup existing rule before overwrite
  - Replace rule by new version
  - Enable rule
  - Set AutoEnable

DLS Change Management for OPS/MVS

CMfOPS

- **Concepts :**

- Deployment of Rexx procedures :

  - Backup existing procedure before overwrite
  - Compile the procedure (if set)

DLS Change Management for OPS/MVS

CMfOPS

- **Concepts :**

- Deployment of Subroutine/Function :
  - Backup existing subroutine before overwrite
  - Refresh(Disable/Enale) ALL rules containing this subroutine
  - Recompile ALL procedures containing this subroutine

DLS Change Management for OPS/MVS

CMfOPS

- **Concepts :**

- Extensive logging/queries :
  - Detailed logging of all functions executed on every lpar
  - Query/Set for all delayed deployments
  - Integrity checking between central pool of libraries and remote OPS libraries

DLS Change Management for OPS/MVS

Standards

Goals : All OPS/MVS rules/procedures should be applicable everywhere

- All rules/procedures have now one version that :
  - Applies to all **existing** environments
    - ✓ PROD(EB,EG,EN)
    - ✓ CONT(EB,EG,EN)
    - ✓ CLON(EB,EG,EN)
    - ✓ KSYS
    - ✓ DEVL
    - ✓ SYST
    - ✓ TEST
  - Will be activated dynaically to new lpars, according to their type

DLS Change Management for OPS/MVS

Demo

Demo

Questions

- QUESTIONS ??????