

ENHANCING AND  
EXTENDING

Deployment  
Packages



# Deployment Packages (DPK)

# WHAT IS THE DPK?

- Automated creation of PeopleSoft environments
- DPK Types:
  - PeopleSoft Images (App DPK)
  - PeopleTools DPK

# DPK TECHNOLOGIES

- Puppet
- Ruby
- PowerShell/Bash
- Automated Configuration Management (ACM)

# DPK SKILLSET

- Puppet, YAML, some Ruby
- Basic PS Admin Knowledge
- Scripting Experience

# DevOps for PeopleSoft

**PeopleSoft Image DPK**

# PEOPLESOFT IMAGE DPK

- Demo Environments
- 1 Environment/machine
- `psft_customizations.yaml` give some control



# TYPICAL DPK BUILD

1. Unzip files and ask configuration questions
2. Deploy middleware/database software
3. Deploy PeopleTools/application homes
4. Create web/app/batch domains
5. Run pre-boot ACM
6. Start domains
7. Run post-boot ACM

**DOES THE DPK DO EVERYTHING?**

**NOPE!**

# DPK LIMITATIONS

- Designed for the Cloud/PI's
- Windows DPK has bugs
- Oracle-only PI databases
- Architecture design limitations
- Not maintenance-focused

**Why use the DPK for  
environment builds?**

# PeopleSoft Image to Custom Builds

# BUILD PLAN

- Bootstrapping your server
- Defining you configuration
- Making the DPK work for you
- Embracing Puppet

# Bootstrapping



# BOOTSTRAP

- `psft-dpk-setup.ps1/sh`
- Deploys PeopleTools/App Homes
- Install middleware
- Changes the DPK Role

# BOOTSTRAP PARAMETERS

- env\_type **dbtier**
- env\_type **midtier**
- env\_type **midtier** -domain\_type appserver
- env\_type **midtier** -domain\_type prcs
- env\_type **midtier** -domain\_type pia
- env\_type **midtier** -domain\_type appbatch
- env\_type **midtier** -deploy\_only -deploy\_type tools\_home
- env\_type **midtier** -deploy\_only -deploy\_type app\_home
- env\_type **midtier** -deploy\_only -deploy\_type app\_and\_tools\_home
- silent
- no\_env\_setup
- patches\_dir path

# BOOTSTRAP TIPS

- Use DFS when bootstrapping your installation
- Install base software only
- Deploy only PS\_APP\_HOME (for catch-up projects)
- Deploy only PS\_HOME (for PeopleTools patches)
- Include new server bootstrap scripts

**psft\_customizations.yaml**

# PSFT\_CUSTOMIZATIONS.YAML

- Define as much configuration as possible
- Think “template” - make your servers as similar as you can
- Define all domains and sites
- Variables and syntax
- TIPS:
  - Keep the structure, but you can add variables/sections
  - Spaces, not tabs (I prefer 2-space indentation)
  - Watch your slashes - Use `/` for paths unless explicitly called out

# .CFG FILE

config\_settings:

Domain Settings/Domain ID:	DOMAIN
PSAPPSRV/Min Instances:	3
PSAPPSRV/Max Instances:	5
PSAPPSRV/Max Fetch Size:	15000
JOLT Listener/Port:	15010
Domain Settings/Allow Dynamic Changes:	Y
Trace/TraceSql:	0
Trace/TraceSqlMask:	31

# DOMAIN FEATURES

feature\_settings:

PUBSUB: “Yes”

QUICKSRV: “No”

QUERYSRV: “Yes”

JOLT: “Yes”

JRAD: “No”

WSL: “No”

DBGSRV: “No”

RENSRV: “No”

# ENVIRONMENT VARIABLES

env\_settings:

PS\_HOME: “{%hiera(‘ps\_base\_norm’)}\\pt-8.55.08”

PS\_APP\_HOME: “{%hiera(‘ps\_base\_norm’)}\\fs-9.2.010”

PS\_CUST\_HOME: “{%hiera(‘ps\_base\_norm’)}\\FS92DEV”

PS\_CFG\_HOME: “{%hiera(‘ps\_base\_norm’)}\\FS92DEV-8.55.08”

PS\_FILEDIR: “{%hiera(‘ps\_base\_norm’)}\\FS92DEV\\files”

PS\_SERVDIR: “{%hiera(‘ps\_base\_norm’)}\\FS92DEV-8.55.08\\files”

TEMP: “{%hiera(‘ps\_base\_norm’)}\\FS92DEV-8.55.08\\temp”



# CONFIG.XML

config\_settings:

Servers/PIA:

ListenAddress: 0.0.0.0

Servers/PIA/WebServer/PIA/WebServerLog/PIA:

LoggingEnabled: true

LogFileFormat: extended

BufferSizeKB: 0

# SITE LIST

site\_list:

SITE1:

appserver\_connections: "%{hiera('pia\_psserver\_list')}}"

domain\_conn\_pwd: "%{hiera('domain\_conn\_pwd')}}"

webprofile\_settings:

profile\_name: "%{hiera('pia\_webprofile\_name')}}"

profile\_user: PTWEBSERVER

profile\_user\_pwd: "%{hiera('appserv\_user\_pwd')}}"

report\_repository\_dir: "%{hiera('report\_repository\_dir')}}"

SITE2:

**psft\_customizations.yaml**

# DPK Customizations

# DPK ROLES

- `site.pp` define the server's role
  - PeopleSoft Image (PUM)
  - DB Server
  - Web/App/Batch Server
  - App/Batch Server
  - Web-only, App-only, Batch-only
  - No Web/App

# DPK ROLES

- Default Roles run ACM
- Change roles to fit your build
- `puppet/etc/modules/pt_role/manifests`

# SITE.PP

```
node default {  
  include ::pt_role::pt_tools_pia  
}
```

# PT\_TOOLS\_PIA.PP

Class['::pt\_profile::pt\_system'] ->

Class['::pt\_profile::pt\_tools\_deployment'] ->

Class['::pt\_profile::pt\_psft\_environment'] ->

Class['::pt\_profile::pt\_pia'] ->

Class['::pt\_profile::pt\_tools\_preboot\_config'] -> (ACM)

Class['::pt\_profile::pt\_domain\_boot'] ->

Class['::pt\_profile::pt\_tools\_postboot\_config'] (ACM)



# MPLS\_TOOLS\_WEBAPP.PP

```
Class['::pt_profile::pt_system'] ->  
Class['::pt_profile::pt_tools_deployment'] ->  
Class['::pt_profile::pt_psft_environment'] ->  
Class['::pt_profile::pt_appserver'] ->  
Class['::pt_profile::pt_pia'] ->  
Class['::pt_profile::pt_samba'] ->  
Class['::pt_profile::pt_domain_boot'] ->  
Class['::pt_profile::pt_source_details']
```

# DPK BUG FIXES

- Enable Tuxedo Domain Features
- Multiple PS\_CFG\_HOME's
- Stop WebLogic installation on App-only and Prcs-only Deployments
- Enable `hiera-eyaml` on Windows

# Puppet

# CUSTOM MANIFESTS

- Learn Puppet syntax/limitations
- Focus on Idempotency
- Use Puppet types and DPK types
  - `file_line{}`
  - `file{}`
  - `pt_webserver_domain{}`

# EMFHUB.PP

- Create web domain
- Customize files
- Run commands
- Create Schedule Tasks

# CUSTOM\_SIGNON.PP

- Deploy HTML/JS/Images to web sites
- Modify `index.html`
- Create load balancer health check
- Set custom Web Profile name
- Customize `text.properties`

# PUPPET MANIFESTS

- Build manifests that eliminate manual changes
- Small, reusable manifests to address a specific configuration
- Look at every change to re-apply when you patch
- **Goal:** Make everything repeatable, in code

# Starting Your Build



# PLANNING YOUR BUILD

- Identify what the DPK can't do for you
  - Sledgehammer approach - all or nothing
  - Little/No additional functionality beyond **psadmin**'s capabilities
  - Delivered DPK Roles don't cover all scenarios

# PLANNING YOUR BUILD

- All (or as much as possible) server/configuration changes by script
  - PowerShell or Bash
  - Puppet
  - Ruby (for custom Puppet Types or Providers)
  - YAML for configuration data

# PLANNING YOUR BUILD

- Consistency between environments
  - Keep `psft_customizations.yaml` simple
  - Consistency == flexibility
  - Template your servers

**[psadmin.io/umrug](https://psadmin.io/umrug)**

**psadmin.io  
Community**

**DPK  
Quickstart  
Course**