BROADCOM

DevTest Best Practices Series

Episode 2

Naming Conventions

Naming Conventions

- Use a standard naming convention that meets your organization's structure and needs
- Use abbreviations to keep the names from getting too long
- Registries, Coordinators and VSEs should have names that distinctly identify and are descriptive.
- A good naming convention allows infrastructure teams to determine the team that is responsible for a specific Registry/Coordinator/VSE.
- Component Best Practices will be on a future episode on Naming Conventions for Registries and VSEs

BROADCOM Recommendation

- Broadcom highly recommends developing and using a naming convention for Virtual Services and Tests.
- Broadcom also recommends NOT using spaces or special characters (non-alphanumeric) in any Virtual Service, Application Test, external test asset, file system path or SV/Application Test property.

Virtual Service Names

- Use Java class camelcase naming convention.
- A naming structure should be descriptive and meet your needs. Make sure the name of the service is something sensible.
- Don’t include the word service in the service name.
- Don’t include protocol information in the service name.
- Project specific Services MUST be distinguishable from other services.
- Service names MUST be auto-descriptive and provide enough information regarding the behavior of the service.
- **There two different approaches we recommend.** You may choose the one which suits your adoption usage better or you can use both depending on how your teams are structured

BROADCOM

Virtual Service Names: Approach 1

- The first approach uses the Application/Service, the version that the Virtual Service takes the place of, the team using it and the virtual service version. This allows the Service Name to be descriptive and helps support reutilization of services.
- Service names will have **at least four parts**. Each part is described below.
  1. Abbreviation of the Service being Virtualized The Account Lookup Service would be AcctLookup
  2. Version of the Application being Virtualized. Add version details by prefixing with AVXX
  3. Abbreviation of Team that created the Virtual Service. Development Team 2 would be DevTm02
  4. Version of the Virtual Service. Add version details by prefixing with SVXX
- Example usage would be:
  1. AcctLookup_AV04.02_DevTm02_SV02.03
  2. AcctBalanceLookup_AV05.03_QaTm04_SV03.13

BROADCOM

Virtual Service Names: Approach 2

- **A second approach** would be to utilize a relationship between **Release or Project, Application groups and Containers** while also versioning your Service.
  1. Track Community Release or Project Services by assigning REL_ or PRJXXX_
  2. Track each Application group and assign a prefix to be use for every VSM/VSI
  3. Track each container name that shares a VSE by name and assign a prefix to be used for every VSM/VSI.
  4. Add version details at position 4 by prefixing with VXX. If the Project branch is not a released version prefix by Source Control Management (SCM) build number BXXX
  5. Optional: Adding environment details to the virtual
- Example usage on Releases would be:
- REL_(2)_(3)_(4)
- REL_SIBS_(3)_(4)
- REL_SIBS_EBS_(4)
- REL_SIBS_EBS_V101
- REL_SIBS_EBS_V101_SIT
- **Complete Service Name would be: REL_SIBS_EBS_V101_SIT**

BROADCOM

Virtual Service Names: Approach 2

- Example usage on Projects would be:
- RJ512_(2)_(3)_(4)
- PRJXXX_SIBS_(3)_(4)
- PRJXXX_SIBS_EBS_(4)
- PRJXXX_SIBS_EBS_V200 or PRJXXX_SIBS_EBS_B2388
- PRJXXX_SIBS_EBS_V200_LNP
- **Complete Service Name would be: PRJXXX_SIBS_EBS_V200_LNP**

BROADCOM

Virtual Service Descriptions

- Use a naming convention with custom delineators.
- You should include information in the Service description that can be parsed for automated storage and retrieval from Source Code and Artifact Repositories.
- This also allows you to leverage metadata information for better Virtual Service management.
- This information could include the following pieces of information or more:
  1. The Application/Service Version
  2. The team that developed the Virtual Service
  3. The Virtual Service Version
  4. Comments
- Example usage would be:
  1. AcctLookup ~04.02~DevTeam02~02.03
  2. AcctBalanceLookup~05.03~QATeam04~03.13~Endurance-Test

Test Suite Names

- Use Java class camelcase naming convention for test suite names: WebUI, WebService, Regressions

BROADCOM

Test Case Names

- Use Java class camelcase naming convention or underscore notation for test case names. Use either lower or upper case, but upper case is recommended for the first letter of a word: Setup.tst, Install_Clean.tst

Test Data Names

- Test data will have the name of the test, the number of the corresponding test requirement in the test case spreadsheet, its own internal number to distinguish it from other datasets created for the same test, and the _Data suffix: PROD_NEG_ FleetStatus_311_1_Data.txt , PROD_NEG_ FleetStatus_311_2_Data.txt

BROADCOM

Config File Names

- Use the name of the underlying server environment to name the configuration files: TEST.config

BROADCOM

Property Names

- Use all caps for properties coming from the system environment or configuration files: PROJECT_ROOT, REPORTS_DIR
- All SV/Application Test properties created by you on your own test project should begin with a common prefix. Usually this refers to your organization such as ACMEINC_
- This will prevent any name clashing with existing SV/Application Test properties.
- Prefix property names in DataSets with ds: dsName, dsAccountNumber, dsAccountBalance
- Prefix filtered run-time properties with fl: flNewAccountBalance, flUpdatedAccountBalance

BROADCOM


BROADCOM®
connecting everything®