Working with Structured Data and Local Version Control



Nick Russo NETWORK ENGINEER

@nickrusso42518 www.njrusmc.net



Agenda



Structured data types

- JSON
- YAML
- XML

Introducing Git version control

Basic Git operations

Easily comparing two files



Why Structure Data?

Text is hard for machines to read

Easy to load into memory

Decouple data from code



```
"customer_list": [
    "name": "john smith",
    "balance": 19,
    "is_happy": false
    "name": "jane doe",
    "balance": 0,
    "is_happy": true
```

- Wrapped in { }
- Lists use []
- **▲ List of dictionaries**
- string, int, and boolean attribs

◄ Comma-separated dicts

■ Another customer

```
# YAML is fun!
customer_list:
  - name: "john smith"
   balance: 19
   is_happy: false
  - name: jane doe
   balance: 0
   is_happy: true
```

◄ Supports comments

■ List of dictionaries again

■ Dash indicates list element

■ Quotes only needed if ":" exists

■ Optional ending of three periods



```
<?xml version="1.0"</pre>
encoding="UTF-8" ?>
<root>
  <customer_list>
    <name>john smith</name>
    <balance>19</balance>
    <is_happy>false</is_happy>
  </customer_list>
  <customer_list>
    <name>jane doe</name>
    <balance>0</balance>
    <is_happy>true</is_happy>
  </customer_list>
</root>
```

- Mandatory XML header
- Mandatory top-most element
- List elements
- Standard data in "elements"

◆ Closing element prepended with forward slash "/"

```
<?xml version="1.0"</pre>
encoding="UTF-8" ?>
<root>
  <customer type="business">
    <name>john smith</name>
    <balance>19</balance>
    <is_happy>false</is_happy>
  </customer>
  <customer type="personal">
    <name>jane doe</name>
    <balance>0</balance>
    <is_happy>true</is_happy>
  </customer>
</root>
```

■ Attribute: business customer

■ Attribute: personal customer

Useful but often hard to translate





Using JSON for our CRM app





Using YAML for our CRM app





Using XML for our CRM app



My Thoughts on Git

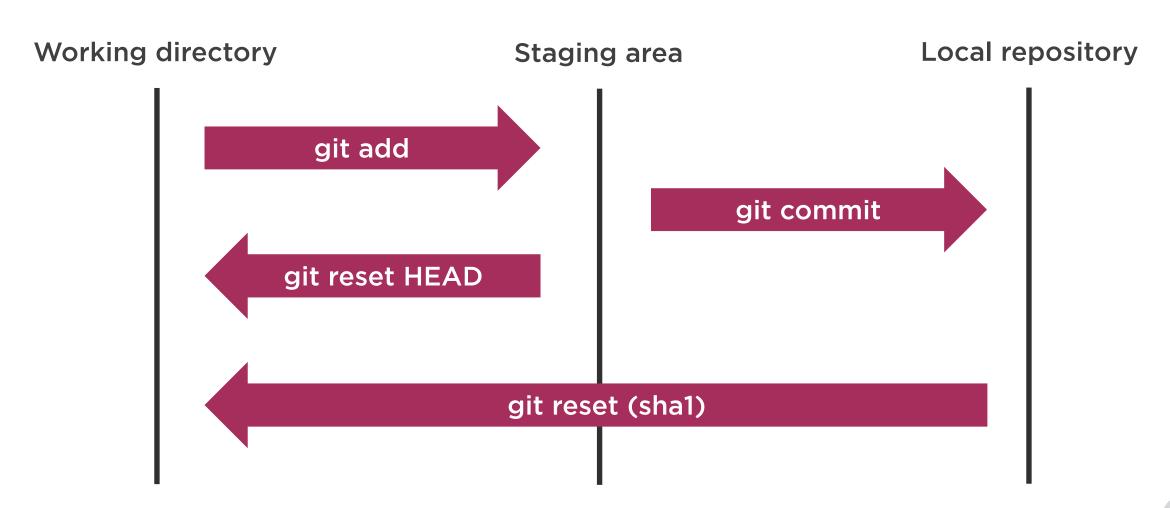
Time travel

Concurrent development

Complex to learn



Local Git Operations Visualized







Gittin' (sic) familiar with Git





Concurrent development with branches





The almighty "unified diff"



Module in Review

Use structured data

Version control all source files

Practice!

