

Finding Database Changes Between Releases

Starting with Platform 2.0, the changesets are split out into individual files, so for later versions you need to look somewhere like <https://github.com/openmrs/openmrs-core/blob/2.1.x/api/src/main/resources/liquibase-update-to-2.1.xml> and use a script like this:

```
version = '2.1.0-beta'
file = 'liquibase-update-to-2.1.xml'

def url = "https://raw.githubusercontent.com/openmrs/openmrs-core/${version}/api/src/main/resources/${file}"
println "fetching ${url}"

def xml = new URL(url).text
println "fetched ${xml.substring(0,10)}"

def trimWhitespace = { s -> s.toString().replaceAll(/\s+/, ' ').trim() }
def parser=new XmlSlurper()
parser.setFeature("http://apache.org/xml/features/disallow-doctype-decl", false)
parser.setFeature("http://apache.org/xml/features/nonvalidating/load-external-dtd", false);

parser.parseText(xml).children().findAll({it.name() == 'changeSet'}).each {
    println trimWhitespace(it.comment);
}
```

Burke Mamlin wrote this Groovy script to determine the sql and changesets that were put into openmrs between any two revisions (prior to version 2.0.0).

```
from = '1.9.1'
to = '1.9.2'

def getUrl = { version ->
    MAJOR = 0; MINOR = 1; v = { v,part -> Integer.valueOf(v.split(/\./)[part]) }
    if (v(version,MAJOR) > 1 || v(version,MINOR) >= 8) // post-maven
        url = "https://raw.githubusercontent.com/openmrs/openmrs-core/@VERSION@/api/src/main/resources/liquibase-update-to-latest.xml"
    else // pre-maven
        url = "https://raw.githubusercontent.com/openmrs/openmrs-core/@VERSION@/metadata/model/liquibase-update-to-latest.xml"
    return url.replace('@VERSION@', version)
}

def trimWhitespace = { s -> s.toString().replaceAll(/\s+/, ' ').trim() }

priorChanges = new XmlSlurper().parseText(new URL(getUrl(from)).text).changeSet*.*@id
newChanges = new XmlSlurper().parseText(new URL(getUrl(to)).text)
    .changeSet.findAll{ !priorChanges.contains(it.@id) }
println newChanges*.comment.collect{ "*" + trimWhitespace(it) }.join("\n")
```

The output of this should be publicized on the release notes page so users know what to expect.