

LexEVS 6.0 Design Document - Detailed Design - Registry

Contents of this Page

- [Registry](#)
- [Backwards Compatibility](#)
 - [Registry](#)
 - [Database](#)
 - [Running LexEVS in a Mixed Environment](#)

Document Information

Author: Craig Stancl
Email: Stancl.craig@mayo.edu
Team: LexEVS
Contract: CBITT BOA Subcontract# 29XS223
Client: NCI CBIIT
National Institutes of Health
US Department of Health and Human Services

Revision History

Version	Date	Description of Changes	Author
1.0	5/14/10	Initial Version Approved via Design Review	Team

Registry

LexEVS 6.0 marks the move to a Database-based Registry. All previous releases of LexEVS relied on a file-system based XML Registry:

XML Based Registry

```
<?xml version="1.0" encoding="UTF-8"?>
<LexBIG_Registry>
  <variables>
    <lastUpdateTime value="1265818643217" />
    <lastUsedDBIdentifier value="a5" />
    <lastUsedHistoryIdentifier value="a0" />
  </variables>
  <codingSchemes>
    <codingScheme urn="urn:oid:1.1.1.0.1" dbURL="jdbc:mysql://bmidev3:3307/junit" dbName="" prefix="lba5_"
status="active" tag="" version="1.0" deactivateDate="0" lastUpdateDate="1265818643217" />
  </codingSchemes>
  <histories />
</LexBIG_Registry>
```

In LexEVS 6.0, this will be represented in two database tables:

registry
«column»
*PK registryGuid: VARCHAR2(36)
resourceURI: VARCHAR2(250)
resourceVersion: VARCHAR2(50)
resourceType: VARCHAR2(50)
dbURI: VARCHAR2(250)
dbName: VARCHAR2(50)
prefix: VARCHAR2(20)
status: VARCHAR2(50)
tag: VARCHAR2(50)
lastUpdateDate: TIMESTAMP
activationDate: TIMESTAMP
deactivationDate: TIMESTAMP
baseRevision: VARCHAR2(50)
fixedAtRevision: VARCHAR2(50)
isLocked: CHAR(1)
dbSchemaVersion: VARCHAR2(50)
dbSchemaDescription: VARCHAR2(255)
«PK»
+ PK_registry(VARCHAR2)
«unique»
+ UQ_registry(VARCHAR2, VARCHAR2, VARCHAR2)

registryMetaData
«column»
lastUpdateTime: TIMESTAMP
lastUsedDBIdentifier: VARCHAR2(50)
lastUsedHistoryIdentifier: VARCHAR2(50)

resourceType:

1. CodingScheme
2. ValueDomain
3. PickList
4. NCIHistory

Motivations for a Database-based Registry:

- Allow multiple LexEVS installations to share a Registry without having to access the same filesystem. In previous releases, either each LexEVS installation had to maintain a copy of the Registry XML file, or the installations had to be on the same physical file-system.
- Avoid XML locking problems. Preventing concurrent edits to an XML file is error-prone. In previous versions of LexEVS, this was done by 'lock' files placed on the filesystem to alert other JVMs that an edit to the XML was being done. When using a database, the database's transaction framework will replace the 'lock' files, preventing concurrent updates.

Backwards Compatibility

LexEVS will be Backwards Compatible with N-1 releases of LexEVS. For version 6.0, LexEVS will be able to be a drop-in replacement for LexEVS 5.1.

Registry

LexEVS 6.0 will use a database-based Registry -- whereas all previous releases have used a file-system XML based Registry file.

- LexEVS 6.0 will be able to Read/Write to the XML based registry as before, with some exceptions:
 - No new loaded Coding Schemes may be added to an XML-based Registry.
 - All new loaded Content will be registered to the database Registry.
 - LexEVS 6.0 will be able to perform admin operations on an XML Registry (remove an entry, change a tag, deactivate, etc).

Database

Database Backwards Compatibility will be handled on the DAO level. The DAO Manager will select the appropriate DAO from the pool based on the table structure of the Resource requested.

Running LexEVS in a Mixed Environment

LexEVS 6.0 and 5.1 are able to share the same set of Resources (database, loaded content, etc), with the following stipulations:

- All changes done by a LexEVS 5.1 installation will be available to the LexEVS 6.0 installation using the same resources.
- Content loaded by a LexEVS 5.1 installation will be accessible to the LexEVS 6.0 installation.



Content loaded by a LexEVS 6.0 installation will **NOT** be accessible to the LexEVS 5.1 installation.