DSpace 4.0 indexing commands

DSpace 3.0, DSpace 4.0rc1		backend
dspace index	org.dspace.browse.IndexBrowse	DB-backed browse
dspace index-init/index-update	org.dspace.browse.IndexBrowse/	DB-backed browse/
	org.dspace.browse.ItemCounter/	DB-backed itemcounter/
	org.dspace.search.DSIndexer	Lucene-backed search
dspace update-discovery-index	org.dspace.discovery.IndexClient	Solr-backed browse and search

Problem A

Background: Default indexing backend changed from Lucene to Solr in 4.0.
Problem A: How to let people know which indexing command to use?

Option 1: Keep the meaning of existing commands. Rely on documentation and messages.

Problem: If user doesn't read documentation and has been using Lucene, he'll be confused why index, index-init/index-update don't do anything.
Solution: Make index and index-init/index-update print a message that the default backend changed; but see Problem C.

Option 2: Introduce a separate command for the Lucene index, remove the index, index-init/index-update commands.

Suggested command would be update-lucene-index because we already have update-discovery-index.

The old commands would be removed to avoid confusion and call attention to the fact that the default indexing backend changed.

Option 3: Introduce a separate command for the Lucene index, change the index, index-init/index-update commands to work with Solr instead.

Suggested command would be called update-lucene-index because we already have update-discovery-index.

This option is based on the assumption that most people are still using the Lucene backend and are ignorant of Solr and its associated command.

Option 4: Introduce a separate command for the Lucene index, change the index, index-init/index-update commands to work with the currently set backend.

Problem: No single option specifies which index is the default. Can there be a single option? See Problem C.

This is a convenience option for those who are still using Lucene to make the transition to Solr seamless. It also declares the "index" (or index-init/index-update) as the recommended indexing command regardless of the configured backend; if we decide to change backend again in the future, this command should seamlessly work with the new default. Using a non-default backend explicitly is still supported by calling the update-discovery-index or update-lucene-index commands.

Problem B

Background: Default item counter backend changed from DB to Solr in 4.0.
Problem B: index-init/index-update now throws the following message:

```
$ /dspace/bin/dspace index-init
Started: 1384372703082
Ended: 1384372747679
Elapsed time: 44 secs (44597 msecs)
Exception: Caching is not supported by the ItemCountDAOSolr as it is not really needed, Solr is faster!
org.dspace.browse.ItemCountException: Caching is not supported by the ItemCountDAOSolr as it is not really
needed, Solr is faster!
 at org.dspace.browse.ItemCountDAOSolr.communityCount(ItemCountDAOSolr.java:82)
 at org.dspace.browse.ItemCounter.count(ItemCounter.java:176)
 at org.dspace.browse.ItemCounter.buildItemCounts(ItemCounter.java:90)
 at org.dspace.browse.ItemCounter.main(ItemCounter.java:57)
 at sun.reflect.NativeMethodAccessorImpl.invoke0(Native Method)
 at sun.reflect.NativeMethodAccessorImpl.invoke(NativeMethodAccessorImpl.java:57)
 at sun.reflect.DelegatingMethodAccessorImpl.invoke(DelegatingMethodAccessorImpl.java:43)
 at java.lang.reflect.Method.invoke(Method.java:616)
 at org.dspace.app.launcher.ScriptLauncher.runOneCommand(ScriptLauncher.java:225)
 at org.dspace.app.launcher.ScriptLauncher.main(ScriptLauncher.java:77)
```

TODO: find out why index-init/index-update need to use ItemCountDAO.

Problem C

Problem C: The search and browse backend is configured using more than one configuration option.

The procedure for changing the default backend to Lucene is documented here: https://wiki.duraspace.org/pages/viewpage.action?pageId=34640921#ReIndexingContentwiththeoldlegacyproviders(DBMSforBrowseorLuceneforSearch)-Re-EnablingthelegacyLuceneSearchand/orDBMSBrowseproviders

```
List of configuration options:
browseDAO.class
browseCreateDAO.class
event.dispatcher.default.consumers
event.consumer.discovery.class
event.consumer.browse.class
<aspect name="Browsing Artifacts" path="resource://aspects/BrowseArtifacts/" />
<aspect name="Discovery" path="resource://aspects/Discovery/" />
```

Note: We can also have a single "index" command to work with the default index, and pass parameters (--lucene, --discovery, --browse-db) to ask for an explicit index to work with. This is just a naming convention, the general ideas are still contained in the options above, just named consistently with what we had in previous DSpace versions.