Diligent

A DIgital Library Infrastructure on Grid ENabled Technology

ETICS Command-Line Tools

Basel, 4 October 2006

Paolo Fabriani – Engineering

Marc-Elian Begin - CERN

Information Society Technologies

Summary

- The ETICS Command Line Tools
  - Read commands
  - Edit Commands
- Building with ETICS
- The ETICS packager
- The ETICS publisher
- Remote Build

Diligent

Overview

- Technology
  - Written in Python (tested with version 2.2, 2.3 and 2.4)
  - Uses ZSI library to interact with the Web Service and perform XML (de)serialisation in the workspace

- Naming & Help
  - All commands start with **`etics-*`**
  - All commands support the option **`--help/-h`**

Diligent

Workspace setup

- The command-line client uses a workspace to define an area in which build and test can performed
- Several workspaces can be created on the same machine
- To define a new workspace, run:

  `etics-workspace-setup`
- If the ETICS client is not already installed (as defined by `ETICS_HOME` env var), `etics-workspace-setup` will install the client in the current directory

Workspace structure

- The workspace has the following directories:
  - *dist:* packaged components
  - *reports:* log files (checkout & build)
  - *repository:* project dependencies (binaries)
  - *<module-name>:* project modules as checked out from VCS

Diligent

Configuration files

- The client behaviour can be controlled using configuration files, name:
  - `etics.conf` in the `<etics-home>/etc` or `<workspace>`
  - `.etics.conf` in the user's home directory
  - Highest priority is the workspace, least is the ETICS home directory

Diligent

ETICS Commands

Read Commands 1/4

- `etics-list-project [options] [<project-name>]`
  - Lists existing projects in the database
  - Useful if you do not remember your project name
- `etics-list-platform [options] [<platform-name>]`
  - Lists existing platforms in the data base
  - Useful if you do not remember a platform name
- `etics-list-resource [options] [<resource-name>]`
  - Lists existing resources in the data base
  - Useful if you do not remember a resource name
- Use the option `--details/-d` to shows details of the items managed by each command

ETICS Commands

Read Commands 2/4

- `etics-get-project [options] <project-name>`
  - gets the project definition
  - contacts the ETICS server and gets the metadata describing the layout of the project
  - the metadata is stored in a local xml file (store.xml)

- Many of the following commands expect an optional `<module-name>` to be specified. If missing, the current project name is taken

- `etics-list-configuration [options] [<module-name>]`
  - lists existing configurations for a given module (specified throuh the `<module-name>` parameter)
  - **Note**: `etics-get-project` must have been called before for this command to work

ETICS Commands

Read Commands 3/4

- `etics-checkout [options] [<module-name>]`
  - gets/checkout a configuration for a given module
  - retrieves the `<module-name>.HEAD` configuration, unless a specific configuration is given as an option:
    - `-c/--config <configuration-name>`
  - **Note**: `etics-get-project` must have been called before for this command to work

ETICS Commands

Read Commands 4/4

- `etics-list-property [options] [<module-name>]`
  - lists properties defined for a module configuration
  - **Note**: after checkout, there's only one configuration per module in your workspace
- `etics-show-configuration-structure [options] [<module-name>]`
  - shows the configuration structure for a module configuration:
    - sub-configurations
    - dependencies
- **Note**: `etics-checkout` must have being called before for this command to work

Edit Commands 1/2

- Allow to edit the full project datamodel
- Each command works on a specific etics object type:
  - **`etics-module`**
  - **`etics-configuration`**
  - `etics-user`
  - **`etics-role`**
  - `etics-platform`
  - `etics-resource`
- each command supports **`add`**, **`clone`**, **`remove`**, **`modify`** and **`prepare`** operations
- **Note**: **`etics-checkout`** must have being called before for this command to work

Diligent

ETICS Commands
Edit Commands 2/2

- Parameters for `add` and `modify` operations can be provided by the user in three different ways:
  - via a `.ini` file:
    - if the `--input <ini-file>` option is set, the system creates a new object using the information in the input file
  - by command line options:
    - If the `--param` option is set, the system creates a new object overriding default values with the ones specified via this option
  - interactively
    - If neither `--param` nor `--input` options are set, the system starts an interactive session requesting the user to provide the needed information
- `etics-user` and `etics-role` commands do not support the previous modalities

ETICS Edit Commands
Operations 1/4

`add` - creates new objects (module, configuration, platform, resource, user, role)

- the new object is saved both in the local xml file and on the remote data base

- **`etics-module add`**
  - Requires the module type as an option **`(--component, -s/--subsystem)`**
  - Automatically creates a <module-name>.HEAD configuration
    - e.g. etics-module add -s newsubsystem
  - New components require a parent to be specified
    - e.g. etics-module add --parent subsys --component newcomp

- **`etics-role add`**
  - Assign a role to a user within a context
    - e.g. etics-role add --dn CN=Guest… -c conf developer

ETICS Edit Commands

Operations 2/4

`clone`

- Allows to clone a whole configuration with all its satellite objects (build, vcs and test commands, subconfigurations, properties and environment, dependencies)
- The new object is saved both in the local xml file and on the remote data base.
  - e.g. **`etics-configuration clone -m module oldconf newconf`**

Diligent

ETICS Edit Commands
Operations 3/4

`modify`

- Modifies an existing object (module, configuration, platform, resource or user)
- In particular, for configurations, it allows to:
  - add, modify or delete the satellite objects
  - modify the configuration metadata
  - modify dependencies
  - modify configuration hierarchy (i.e. subconfigurations)

  - Both the .ini and the interactive approaches can be used

Diligent

ETICS Edit Commands
Operations 4/4

prepare

- can be only used with module, configuration, resource and platform commands
- Creates a .ini file in the workspace containing the current information about the element (default values are set if the element does not exist)
- The default file name is **`<element-type>-<element-name>.ini`**

remove

- Deletes an element and all its children
- Could fail if the element itself (or its children) is used by other elements
  - e.g. configuration set as a dependency
  - e.g. the configuration is child of the main configuration

Building Software with ETICS

Diligent

Building Software with ETICS

- ETICS can build your code and create distribution packages
- The system relies on the information you, as a user, set in the ETICS data model
- It consists essentially of three main steps:

BUILDING → PACKAGING → PUBLISHING

The `etics-build` command

- The ETICS command to perform builds is called *etics-build*. The syntax of the command is:

```
etics-build [options] [<module-name>]
```

- It requires that the configurations to build have been previously checked out with the `etics-checkout` command

The `etics-build` command

Command options 1/2

Option	Description
-h, --help	Show the usage instructions
-c, --config <configuration-name>	Define a specific configuration to be used instead of the default one, where default is the first configuration found in the store. This is normally not required unless multiple or non-default configurations of the selected module have been checked out
-p, --property <property=value>	This option allows passing properties to the build process. If the property is already defined, its value is overridden by the value specified in the command-line. To pass multiple properties, use multiple –p options
-e, --env <env=value>	This option allows passing environment variables to the build process. If the variable is already defined, its value is overridden by the value specified in the command-line. To pass multiple variables, use multiple –e options
-t, --target <target-name>	Execute the build stopping at the specified target, If not specified the publish target is executed. Allowed targets are: clean, init, checkstyle, compile, test, doc, packager, publish, install
--platform <platform-name>	Overwrite the local platform (useful for testing or when the local platform is not a valid ETICS platform, but it's compatible with one).

The `etics-build` command

Command options 2/2

Option	Description
--nobuild	Do not perform the build, just print the sequence
--continueonerror	Do not stop building if an error is found
--nodeps	Only build the currently specified module (do not build children and dependencies)
--force	Force the build of unmodified modules
--verbose	Print verbose messages
--version	Return the current client version number.

Build Targets

Target name	Description	Mandatory
clean	Command to clean	No
init	Command to perform initialization operations before compiling	No
checkstyle	Command to perform code checking operations (coding conventions)	No
compile	Command to compile code	No
test	Command to perform static tests like unit tests or coverage tests	No
doc	Command to generate documentation	No
packaging	Command to generate distribution packages	No
publish	Command to publish build artifacts to standard distribution formats	No
install	Command to install the package	No (but no packaging can be done if empty)

init → checkstyle → compile → test → doc → packaging → publish

Packaging

Diligent

The ETICS Packaging System

- The ETICS Client comes with a built-in packaging system that allows to build distribution packages on all supported platforms in different formats (tarballs, RPMS, debs, MSIs, etc)
- In order to use the ETICS Packager *the install* target has to be defined, since it is used internally to identify what the content of the package should be.
- In order to activate the ETICS Packager, leave the *packaging* target undefined

Diligent

How to Specify What to Package

- The packager relies on two elements from the module metadata
  - The install target
  - The installation prefix

- Specify what to be package by installing software under ${prefix} in the install target

- The ETICS Packager will then resolve the ${prefix} property to an internal location of its own

Publishing

The ETICS Publisher

- The ETICS Client comes with pre-defined publishing conventions that allow retrieve build artifacts (packages, logs, etc) from the various modules into a common location. The ETICS Publisher uses the following conventions:
  - All **checkout** and **build logs** are stored in the ***reports*** directory of the workspace. The main log file is called:

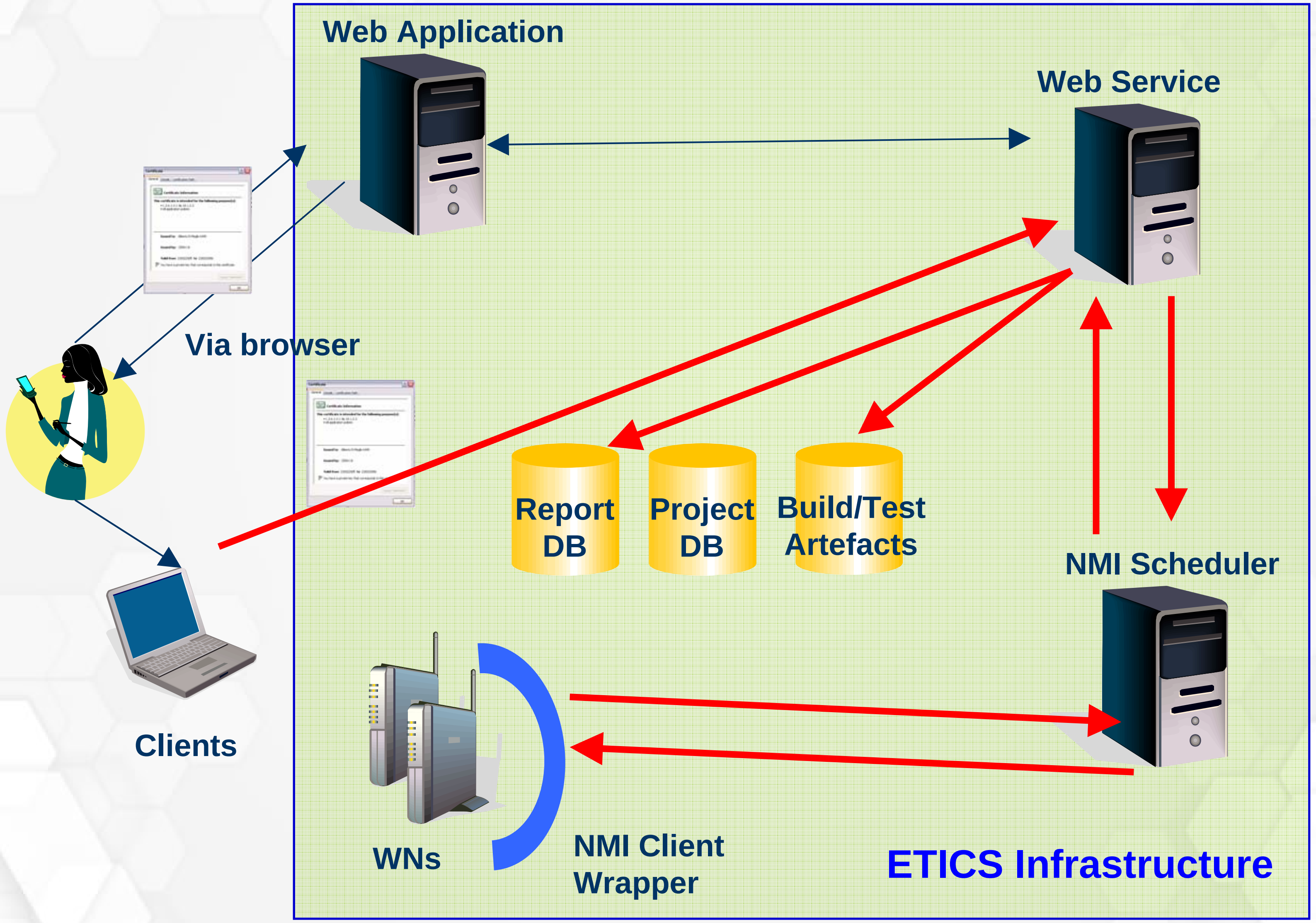
    build-status.xml

    and contains summary checkout and build information for each module in the current build run. Detailed logs of the checkout and build operations of each modules are saved in files of the format:

    [CHECKOUT | BUILD]-<module-name>-<configuration-name>.log
  - All **packages** are stored in the directory ***dist***

Remote Build

The Remote Build Service



The Build Service
Condor and NMI

- ETICS uses Condor and NMI to manage the submission of build/test jobs to resource nodes
- Condor offers mechanisms and policies that support High Throughput Computing (HTC) on large collections of distributed computing resources
- NMI is a multi-platform tool designed to provide (automated) submission of build/test jobs and job monitoring

Diligent

ETICS Production Setup

- **etics.cern.ch**
  - Official ETICS submission node - production host
  - 2250+ jobs (as of 22 Sept 2006)
- **etics-01.cnaf.infn.it**
  - 200+ jobs (as of 22 Sept 2006)
- **“Grand Central” at University of Wisconsin**
  - Hundred Thousands jobs used by several projects

Diligent

Automated Build in DILIGENT

- DILIGENT Current constraints:
  - DILIGENT VCSs requires authenticated access
  - Passwords cannot be published in ETICS
- Current scenario
  - A dedicated server builds DILIGENT configurations daily using the build commands
  - Results are published through a custom UI at http://grids16.eng.it/BuildReports
- Future scenario
  - Some dedicated hosts will join the NMI infrastructure accepting only DILIGENT builds

Diligent

References

- Project Home page:
  - http://www.eu-etics.org
- ETICS Web Application:
  - https://grids17.eng.it:8443/etics
- Bug and requirements tracking (Savannah)
  - http://savannah.cern.ch/projects/etics/
- Twiki
  - https://uimon.cern.ch/twiki/bin/view/ETICS/WebHome
- Mailing-lists
  - etics-discuss@cern.ch
  - etics-bugs@cern.ch