EnthoughtBase Documentation

Release 3.0.2

Enthought

March 19, 2009

CONTENTS

Contents:

INTRODUCTION

This is a brief description of how GUI toolkit selection is intended to work across ETS.

The issues are as follows:

- An application, if it is not intended to be portable across toolkits, must be able to explicitly specify what toolkit should be used.

- If an application is portable across toolkits then the user must have a consistent, simple and convenient way to specify what toolkit should be used.

- ETS components (eg. Pyface, TraitsUI) that support multiple toolkits must agree on which toolkit to use.

- If a toolkit has not been explicitly specified then an ETS component must decide on a toolkit to use - usually based on what supporting modules or eggs have been installed.

Ideally an ETS component should be designed so that support for a new toolkit can be added without changing the component itself - but this is out of scope for this document.

ENTHOUGHT.ETSCONFIG.API.ETSCONFIG.TC

The `ETSConfig` is a singleton that has a `toolkit` string property. Its value is the name of the toolkit or an empty string if the toolkit has not yet been selected.

When selecting a toolkit, an ETS component should look at the value of `toolkit`. If it is not an empty string then it should configure itself to use that toolkit. If it is unable to, perhaps because a backend egg hasn't been installed, then it should raise an exception.

If `toolkit` is an empty string then the component must determine what toolkit to use and configure itself accordingly. It must then set `toolkit` to the name of the selected toolkit so that other components follow suit.

If an application wants to explicitly set the toolkit to use then it must set the `toolkit` trait to an appropriate value. It must do this before importing from any other ETS component.

THE USER'S PERSPECTIVE

If the user only ever uses one toolkit then they simply don't install the ETS component eggs for any other toolkit and applications should automatically configure themselves correctly.

If the user needs to override the automatically chosen toolkit then the `-toolkit` command line flag can be used to explicitly specify the toolkit to be used. It will have no effect if the application itself has explicitly specified the toolkit.

Alternatively the `ETS_TOOLKIT` environment variable can be used to define the toolkit to be used by default.

In summary, the toolkit is selected according to the following (in decreasing order of precedence):

1. Explicitly set by the application.

2. Set using the `-toolkit` command line flag.

3. Set using the `ETS_TOOLKIT` environment variable.

4. Determined dynamically by individual ETS components. (Note that as components don't cooperate to determine a selection that satisfies each of them, it is quite possible that an invalid selection is made depending on which component gets to make the choice.)