

abolish/1

Description

```
abolish(Predicate)
abolish(Functor/Arity)
```

Removes a runtime declared dynamic predicate from an object database.

Template and modes

```
abolish(+predicate_indicator)
```

Errors

Predicate is a variable:

```
instantiation_error
```

Functor is a variable:

```
instantiation_error
```

Arity is a variable:

```
instantiation_error
```

Predicate is neither a variable nor a valid predicate indicator:

```
type_error(predicate_indicator, Predicate)
```

Functor is neither a variable nor an atom:

```
type_error(atom, Functor)
```

Arity is neither a variable nor an integer:

```
type_error(integer, Arity)
```

Predicate is statically declared:

```
permission_error(modify, predicate_declaration, Functor/Arity)
```

Predicate is a private predicate:

```
permission_error(modify, private_predicate, Functor/Arity)
```

Predicate is a protected predicate:

```
permission_error(modify, protected_predicate, Functor/Arity)
```

Predicate is a static predicate:

```
permission_error(modify, static_predicate, Functor/Arity)
```

Predicate is not declared for the object receiving the message:

```
existence_error(predicate_declaration, Functor/Arity)
```

Examples

To abolish any dynamic predicate in this:

```
abolish(Predicate)
```

To abolish a public or protected dynamic predicate in self:

```
::abolish(Predicate)
```

To abolish a public dynamic predicate in an object:

```
Object::abolish(Predicate)
```

Copyright © Paulo Moura — Logtalk.org
XHTML + CSS Last updated on: January 21, 2006