

Unique ID for any object

Status of this Memo

This memo describes a historic protocol for the Xaraya community. It does not specify an Xaraya standard of any kind. Distribution of this memo is unlimited.

Copyright Notice

Copyright © The Digital Development Foundation (2003). All Rights Reserved.

Abstract

This RFC presents a proposal for a Unique Identification System for Abstract Objects inside the Xaraya Architecture.

Table of Contents

1 Introduction	3
2 Pros	4
3 Cons	5
4 Implementation Idea	6
5 Alternative Approach	7
6 Alternative Approach Comments	8
7 Relevant References	9
Author's Address	10
A Example appendix	11
Intellectual Property and Copyright Statements	12

1. Introduction

Across many of our database tables is perceptible the many cases where module name, and some other information like module item, or module variable is being used for uniquely identify a given set of informations. There are other cases using as much as 3 different inputs to generate a unique ID. The idea is to abstract this

2. Pros

The possibility of having the same unique identifier for an object across all of stored information in the Xaraya Architecture

3. Cons

There is a performance hit from bring this unique lookup from the particular table to a general system.

This will double the number of SQL queries of the use cases, unless there is a query abstraction system present.

4. Implementation Idea

The input is an associative array. For example: `array('module' => 'articles', 'variable' => 'pubtype1')`; or: `array('module' => 'articles', 'object' => 'pubtype1', 'action' => 'display')`; So there would be a table for $2*n$ number of inputs. (the key is one string to be looked for, and the values are other) All which will be assigned id numbers. And then select on a specialized table for the given number of inputs the unique ID looked for.

5. Alternative Approach

An alternative approach consists of using the (module + item type + item id) combination as unique identifier, and to rely on the original module to provide a relevant title and link to that module item via a standard user API function : <module>_userapi_getitemlinks().

This allows hook modules (or any other module) to present titles and links of other module items when they want to, without paying the price mentioned above for all queries.

See also RFC 42 about the module hooks system.

6. Alternative Approach Comments

The alternative approach is much better than the original one, which was unnecessarily complex...

Although i think in the future, a more 'core' approach, which lends less flexibility to the modules would make xaraya able to use this feature more extensively. (For example in a Short-Urls to Intuitive-Urls transition)

7 Relevant References

Author's Address

Flavio Borges Botelho

Xaraya Development Group

E-Mail: nuncanada@xaraya.com

URI: <http://www.xaraya.com>

A. Example appendix

Any section which is present after the references will become an appendix

Intellectual Property Statement

The DDF takes no position regarding the validity or scope of any Intellectual Property Rights or other rights that might be claimed to pertain to the implementation or use of the technology described in this document or the extent to which any license under such rights might or might not be available; nor does it represent that it has made any independent effort to identify any such rights. Information on the DDF's procedures with respect to rights in standards-track and standards-related documentation can be found in RFC-0.

The DDF invites any interested party to bring to its attention any copyrights, patents or patent applications, or other proprietary rights which may cover technology that may be required to practice this standard. Please address the information to the DDF Board of Directors.

Acknowledgement

Funding for the RFC Editor function is provided by the DDF