Database - Feature #2206

use encoded passwords for _User table data import

12/05/2013 04:07 PM - Eric Faulhaber

Status:	Rejected	**Start date:**	
Priority:	Normal	**Due date:**	
Assignee:		**% Done:**	0%
Category:		**Estimated time:**	0.00 hour
Target version:	Deployment and Management Improvements		
billable:	No	**version:**	
vendor_id:	GCD		

Description

History

#1 - 12/05/2013 04:18 PM - Eric Faulhaber

Currently, we have a special data import requirement for applications which use the _User metadata table: the deployer must provide a pwds.txt file containing the user IDs and clear-text passwords along with the *.d dump files. This is to get around the current limitation in P2J of an incompatible hash algorithm for the 4GL ENCODE feature.

The storage of user passwords in a clear text format, even if temporary, is a security concern.

If we do not ultimately provide a compatible ENCODE implementation, we will need customers to modify their applications to intercept logins (or perhaps provide a special-purpose login), to accept the users' passwords and hash them using the P2J ENCODE implementation. These hashed passwords would then be stored (possibly in a special-purpose database table), and eventually would be saved off into a text format, like in pwds.txt.

Data import will have to be adjusted to skip the encoding step if password data is provided pre-encoded. The decision whether or not to encode during import would be governed either by some special configuration or perhaps something as simple as a different password text file name.

#2 - 03/28/2016 10:40 AM - Eric Faulhaber

- Status changed from New to Rejected

We have since implemented a backward-compatible ENCODE function, so this issue is moot.

#3 - 11/16/2016 01:02 PM - Greg Shah

- Target version changed from Deployment and Management Improvements to Deployment and Management Improvements