

Base Language - Feature #6420

implement the AUDIT-ENABLED() built-in function

05/26/2022 03:29 PM - Greg Shah

Status:	Review	Start date:	
Priority:	Normal	Due date:	
Assignee:	Theodoros Theodorou	% Done:	100%
Category:		Estimated time:	0.00 hour
Target version:		vendor_id:	GCD
billable:	No		
Description			
Related issues:			
Related to Base Language - Feature #6422: implement auditing support includin...			WIP

History

#1 - 03/06/2023 06:08 PM - Greg Shah

- Assignee set to Theodoros Theodorou

#2 - 04/13/2023 04:52 PM - Theodoros Theodorou

How should we implement this feature? Should we just add a flag into directory/client.xml or should we replicate the exact same features as they do in progress. As I understood so far from reading the documentation, the latter seems very complicated.

#3 - 04/13/2023 05:17 PM - Theodoros Theodorou

- Status changed from New to WIP

#4 - 04/14/2023 08:53 AM - Greg Shah

- Related to Feature #6422: implement auditing support including AUDIT-POLICY and AUDIT-CONTROL added

#5 - 04/14/2023 09:02 AM - Greg Shah

This is meant to report on whether the 4GL-compatible auditing support is activated. That means it is dependent upon the functionality in [#6422](#).

As far as how to activate it, we can either implement a simple directory flag or we can link this back into the SecurityManager auditing support which we already have. Though I like the conceptual simplicity of using the SecurityManager auditing as a common facility, the downside is that it may complicate the 4GL-compatible implementation. I've posted the next steps in [#6422](#). I think it is best to work that first.

#6 - 05/26/2023 07:48 AM - Greg Shah

When this is implemented, please put it in branch 6422a.

#7 - 05/26/2023 03:20 PM - Theodoros Theodorou

How should I implement this feature? Should I add a boolean in server/directory.xml? There is already another boolean for the other audit implementation: security/audit/enabled. Should I use this one or create another one? If I need to create another one, do you have a preference regarding the path?

#8 - 05/26/2023 03:27 PM - Greg Shah

There is already another boolean for the other audit implementation: security/audit/enabled.

No, the two audit facilities have no relation so we can't use the existing feature.

How should I implement this feature? Should I add a boolean in server/directory.xml?

I don't think so. In the 4GL, isn't this just reporting on whether there are any active audit events configured? In other words, if the 4GL runtime is going to add records into the `_aud-audit-data` table then `AUDIT-ENABLED()` will return true. Please test it and see. I think this is just something that will report runtime state from `AUDIT-CONTROL` or `AUDIT-POLICY`.

#9 - 05/29/2023 04:20 AM - Theodoros Theodorou

To enable auditing, the following commands should be executed in proenv:

```
prodb test_db empty // create an empty database named test_db
...
proutil test_db -C enableauditing area AuditData // enable auditing for the test_db
```

If the `proutil ... enableauditing` is not executed, then the auditing is not enabled. This means the `AUDIT-ENABLED()` will always return false and nothing will be inserted into `_aud-audit-data`. Even if we call commands like `LOG-AUDIT-EVENT`, nothing will be added into the db.

I believe the best way to simulate the `proutil ... enableauditing` functionality is by adding a flag in `server/directory.xml`.

#10 - 05/29/2023 10:21 AM - Greg Shah

I believe the best way to simulate the `proutil ... enableauditing` functionality is by adding a flag in `server/directory.xml`.

Agreed. This seems to be best done as a new flag for a given database in the persistence section.

Eric?

#11 - 05/29/2023 11:08 AM - Eric Faulhaber

Greg Shah wrote:

I believe the best way to simulate the proutil ... enableauditing functionality is by adding a flag in server/directory.xml.

Agreed. This seems to be best done as a new flag for a given database in the persistence section.

Eric?

The persistence section is meant for cross-database configuration; database is meant for per-database configuration (persistence and database are at the same level; there is zero or one persistence node, zero or more database nodes).

If we want the flexibility to enable auditing on a per-database instance level, I would suggest putting this flag under `.../database/<db_name>/`. If there are options beyond on/off, I would make an audit container, with the first setting being enabled (defaults to false). If it is just on/off, I would suggest a boolean enable-auditing flag directly beneath `.../database/<db_name>/`.

#12 - 06/08/2023 08:50 AM - Theodoros Theodorou

- *Status changed from WIP to Review*

- *% Done changed from 0 to 100*

#13 - 06/08/2023 08:51 AM - Theodoros Theodorou

The changes are in the p2j branch 6422a

#14 - 08/11/2023 10:55 AM - Greg Shah

Eric: This needs your code review.