Data protection in practice - Best practices for designing Roles in ActivityInfo

Starting shortly, Please wait!
Presented by the ActivityInfo Team

- Track activities, outcomes
- Beneficiary management
- Surveys
- Work offline/online
Meet your instructors

Jeric Kison
Customer Success Director
BeDataDriven

Victoria Manya
Customer Education Specialist
BeDataDriven
Data Security Webinar Series
Cybersecurity Awareness Month

SESSION 1
OCT 5
Top 5 data security risks for M&E professionals and what you can do about them

SESSION 2
OCT 12
Data protection in practice - Best practices for designing Roles in ActivityInfo

SESSION 3
OCT 19
Office Hours - Designing Roles in ActivityInfo
Poll

What challenges have you faced when managing Roles and Permissions in your organization's database?

- **Complexity**: Dealing with intricate permission setups.
- **Role Overload**: Managing too many roles and tailoring roles to specific needs.
- **Security Balance**: Balancing data protection and access.
- **Onboarding Time**: Time-consuming user setup and Clarifying roles with users.
- **Testing Burden**: Rigorous testing before implementation.
- **Audit Tracking**: Keeping records of changes.
- **Resolving overlapping permissions**.
- **Scalability** Challenges: Adapting to expansion.
- **Documentation gaps**: Keeping role documentation current and Lack of user training.
1

Data security principles
What is data security?

Confidentiality: Confidential data protected from exposure to unauthorized parties

Integrity: Prevention of unauthorized changes to your data

Availability: Ensuring data is available when needed to the parties who need it
Principle of least privilege

“a security architecture should be designed so that each entity is granted the minimum system resources and authorizations that the entity needs to perform its function”

- US Committee on National Security Systems
What role does the M&E professional play in data security?

- Planning M&E systems
- Planning data collection
- Sharing data with internal and external stakeholders
- Communicating results
2

Understanding roles in ActivityInfo
Understanding Roles in ActivityInfo

Determining User Permissions and Actions

- Roles determine user permissions and actions within your database
- Create Roles to control data access and user actions

Roles

- Administrator
- Case worker
- Supervisor
- Case Observer

Operations and conditions

- Permitted operations:
  - View all records
  - Display in the list of forms
  - Add any record
  - Edit all records
  - Edit 'reviewer only' fields
  - Delete any record
  - Bulk record delete

Grants and parameters
Understanding Roles in ActivityInfo

Roles=Combinations of Grants and Parameters.
Understanding Roles in ActivityInfo

Key Concepts

- **Resources**: Forms, Folders, Reports, and Databases.
- **Operations**: Actions performed on resources and users.
- **Grants**: Identify resource-specific operations.
  - Grants are inherited and can be overridden.
- **Optional Grants**: Enable flexibility in permission assignment.
- **Conditions**: Define rules for user operations.
- **Parameters**: Assign attributes to users for conditions.
- **Roles**: Combinations of Grants and Parameters.
Understanding Roles in ActivityInfo

Key Concepts

Resources

- Databases
  - Database

- Folders
  - Folder

- Forms
  - Form

- Reports
Understanding Roles in ActivityInfo

Key Concepts

Operations = Actions performed on resources and users

**Edit permissions**
- View all records
- Display in the list of forms
- Add any record
- Edit all records
- Edit 'Reviewer only' fields
- Delete any record
- Bulk record delete
- Export records

**Management**
- Manage translations
- Manage record locks
- Manage collection links
- Audit user actions

**Sharing/ publishing**
- Publish reports

**Design**
- Add forms, folders, and reports
- Edit forms, folders, and reports
- Delete forms, folders, and reports
Understanding Roles in ActivityInfo

Key Concepts

Grants: Identify resource-specific operations.

- A Grant identifies the specific resource on which the operations a user is allowed to perform are applied.
- Grants can be applied to any type of resource.
- Grants are inherited and can be overridden.
- Grants are inherited by all contained resources.
- A grant that is applied to a folder applies the settings to all the forms/folders within that folder, while a grant that is applied to an entire database applies the settings to all the resources within that database.
- Inherited grants can also be “overridden” on a contained resource by applying a new grant on that resource, if desired.
# Understanding Roles in ActivityInfo

## Cards Analogy

<table>
<thead>
<tr>
<th>Resources</th>
<th>Operations</th>
<th>Grants</th>
</tr>
</thead>
<tbody>
<tr>
<td>● Take</td>
<td>● Tear</td>
<td></td>
</tr>
<tr>
<td>● Colour</td>
<td>● Remove</td>
<td>Take the blue cards</td>
</tr>
</tbody>
</table>

By ActivityInfo
Understanding Roles in ActivityInfo

Key Concepts

Grants, operations and reference forms.

To ensure that the visibility of reference forms is restricted, you must go to your database design, select the reference form and untick the box below in your list of operations:
Understanding Roles in ActivityInfo

Key Concepts

Resources, Grants, and operations tied together:

- Here is a form identifying the resource
- Project supervisors can edit, view, and delete in forms identifying operations (actions) that project supervisors can do
- Project supervisors can edit, view and delete records in the WASH cluster or in the entire database=Grants
Understanding Roles in ActivityInfo

Key Concepts - Inheriting grants

Say you have a database setup like so...

On my role, I apply an explicit grant to the database.
In doing so, all the contained resources (folders, forms) also inherit this grant and the operations are applied to these resources as well. We can think of this as an implicit grant.

If I apply my explicit grant to the folder instead, then the contained resources inherit the grant in the same way. Note that I do not receive permissions on the root level forms as a result, as there is no grant on the database.

If I apply my explicit grant directly to a form instead, then the grant is only applied to the form, note that the other resources in the database do not receive permissions as a result, as there are no other grants.
Understanding Roles in ActivityInfo

Key Concepts - Overriding grants

Say you have a database setup like so:

On my role, I apply an explicit grant to the database to allow the user to view records. In doing so, the combined resources (roles, forms) also inherit this grant and the permissions are applied to those resources as well. The combination of the is an explicit grant.

In addition to the database grant, I also apply an explicit grant to the folder to view, edit, add, and delete records. The inherited grant from the database is overridden on the folder, and the folder (and its contained resources) has the explicit grant applied instead.

We can also use overrides to restrict permission in a subset of the database. In addition to the database grant, I also apply an explicit grant to the folder with no permissions permitted - i.e. I am forbidding access.
Understanding Roles in ActivityInfo

Key Concepts

- Optional Grants: Enable flexibility in permission assignment.
  - A grant can be set as optional, which means that you can choose whether to enable the grant for each user that you invite to your database.
Understanding Roles in ActivityInfo

Key Concepts: Conditions

- Conditions enable you to define rules that determine which records a user can perform operations on.
- Rules are always expressed as a formula that evaluates to TRUE or FALSE.
- The rule builder helps you write the most common kinds of formulas including:
Understanding Roles in ActivityInfo

Key Concepts

- The rule builder helps you write the most common kinds of formulas including:
  - determining whether a record is related to a parameter
  - determining whether a specific field matches a specified value or set of values
  - determining whether a record is assigned to the user

You can always use the formula editor to write your own formulas, including those that make reference to related fields or subrecords.
Understanding Roles in ActivityInfo

Key Concepts-Parameters

- Parameters are attributes that are assigned to a user which can be used in conditions to control the record-level operations they are allowed to perform.

- Parameters are linked to a reference form, such as "Regions", which provides the possible values that can be assigned to a user.
3

How to create roles in ActivityInfo
How to create roles in ActivityInfo

Steps involved in adding a Role

1. Add the role
2. Add resources to the role
3. Grant a resource
4. Assign operations
5. Save and invite a user

Before you start
- Make sure you have already added a database and
- You have been assigned to a role with the “Manage roles” and “Manage users” operation permitted.
- If not, for practice purposes, you can use the training and monitoring template to add a new database.
How to create roles in ActivityInfo

Example 1: Limiting access to records based on a Parameter

- Scenario: Program staff serving beneficiaries by region.
  - Ensure staff can only access records in their assigned region.
- Create a Role for "Programme Officer" with parameters and conditions.
  - Create a single role
  - create a parameter
  - Assign Grants
- Benefits of parameters in this scenario
How to create roles in ActivityInfo

Example: Limiting access to records based on assigned user

- **Scenario: Case management**
  - Case records contain highly sensitive personal information
  - Each case assigned to a single case worker
  - Each case worker has a supervisor for oversight

- **Role requirements**
  - Case workers to access only records of the cases they are assigned to
  - Supervisors can access their own cases as well as those that belong to their case workers
  - Case workers should be able to view reference records in data entry but not be able to edit them

- **Role configuration**
  - Add *condition* based on user assignment
  - Add *parameter* for Supervisor assignment
  - *Grant* to Reference folder with *View records* permission only
4

Best practices for designing roles
Best practices for designing roles

Design roles with narrow user permissions

- **Employ the Principle of Least Privilege as a rule**
- To maintain integrity: consider whether users really need the ability to **Edit** or **Delete**
- To protect sensitive data: be careful with **Export** and **Publish** permissions
- **Grant User Management** operations selectively

[Diagram showing permitted operations with checkboxes for various actions like View, Add, Edit, Delete, View reviewer-only fields, Export, Add forms, folders, and reports, Manage reference data, Manage translations, Manage record locks, Manage users, Manage roles, Manage collection links, Audit user actions, Share reports, Publish reports.

ActivityInfo
Best practices for designing roles

Design roles with the *broad* context in mind

- **Understand** roles and responsibilities within your organization
- **Educate** end users on data security
- Regularly **review** and **update** roles when the context changes
Best practices for designing roles

How many roles do I need?

You will likely need to create fewer roles than you think.

- Design roles according to the tasks that a group of users needs to do, then
  - use **optional grants** to enable access to different resources while maintaining a common set of universal permissions
  - use **parameters** to differentiate which records a user can work with
What we’ll do:

- Discuss questions and best practices for designing roles in the ActivityInfo based on real examples and your own use cases.