Access Level

About

Access Level is a security feature for ServiceNow that restricts access to records based on account's access level and the access level of the record. This is helpful when your organization needs to restrict access to records based on account's access level.

This guide is to help ServiceNow admins setup Access Level to limit who can see designated records. The goal for Access Level is to work in conjunction with the default out-of-the-box (OOTB) ServiceNow role base security model to insure future upgradeability across ServiceNow releases.

How Access Level works

The *Access level* is a custom developed feature in ServiceNow and not an out-of-box feature of ServiceNow. The purpose of the feature is to limit who can see tickets. There are three areas where a *Access level* is defined; users, tickets, and assignment groups (see below for further information on *Access level* areas). A *Access level* is an integer value from 0 to 2,147,483,647; with 0 (zero) being the lowest *Access level* and any number higher being of a higher access level.

User Access Level

Every account in ServiceNow has a *Access level* field. Every account defaults its *Access level* to 0 (zero); the lowest level possible *Access level*. A *Access level* on an account designates the highest level of clearance the account can see or assign to a ticket. For example, if an account has a *Access level* of 0, then the account can only see tickets designated as *Access level* 0. Whereas, if the account has a *Access level* of 10. then the account can see tickets designated as *Access level* 0 through 10.

Ticket Access Level

Every ticket has a *Access level* field. Every ticket defaults its *Access level* to 0 (zero); the lowest possible *Access level*. A *Access level* on a ticket designates the lowest level of clearance an account must have to see the ticket. For example, if a ticket has a *Access level* of 0 (zero), then any account can see the ticket. Whereas, if the ticket has a *Access level* of 10, then accounts with *Access level* is 10 or greater can see the ticket; accounts with *Access level* is 9 or less cannot see the ticket.

Group Access Level

Every assignment group has a *Access level* field. Every assignment group defaults its *Access level* to 0 (zero); the lowest possible *Access level*. A *Access level* on a group designates the highest level of clearance a ticket can be to be (re)assigned to the assignment group. This is to prevent too high of a *Access level* ticket being assigned to an assignment group where the members of the group cannot see the ticket. For example, if a ticket has a *Access level* of 0, then the ticket can be (re)assigned to any assignment group. Whereas, if a ticket has a *Access level* of 10, then the ticket can only be (re)assigned to assignment groups whose *Access level* is 10 or greater; the ticket cannot be (re)assigned if the assignment groups whose *Access level* is 9 or less.

Setting Access Levels - Tickets

Only fulfillers can set a ticket's *Access level*. See the following sections on further restrictions.

Increasing Ticket Access Level

A ticket's *Access level* can be increased up to the current account's defined *Access level*. Refer to the following scenarios: Scenario A - Aligned Clearance

A ticket has a *Access level* of 0. The current account assigned to the ticket has a *Access level* of 10. If that account chooses, they can increase the *Access level* up to 10 and no more. Scenario B - Misaligned Clearance

A ticket has a *Access level* of 0. The current account (Bob) assigned to the ticket has a *Access level* of 10. Another fulfiller account (Sam) has a *Access level* of 20. If Sam increases the ticket's *Access level* above 10, then Bob can no longer see the ticket, even thou Bob is assigned the ticket.

Decreasing Ticket Access Level

A ticket's *Access level* can be decreased by the assignment group's manager or co-managers. This is to prevent the accidental de-classification of ticket and information. Setting Access Levels - Users & Groups

Only accounts with the u_ent_access_level_manager role can define clearance roles for accounts and groups. This is a restricted role only to be used by designated IT Risk Management personal (ITRMs).

Follow along

ServiceNow offers many ways to solve a problem or configure an operational business model. These instructions are *a* way and does not represent *the* way on creating an Access Level feature. The best way to use these instructions is to read through them and see how the feature can be adopted and modified to fit your organization requirements.

> **Update sets**
>
> There is an update set that implement this guide; Access Level. The update set can be found at https://github.com/ChristopherCarver/AccessLevel.

> **Modifying OOTB Tables**
>
> This guide focuses on creating a robust feature tied closely to the out of the box (OOTB) tables already existing in ServiceNow. There are alternative implementation solutions within ServiceNow to accomplish the same result. Your mileage may vary depending on scope defined by and practices set by your organization and/or development team. This feature is meant to be as open and flexible to meet varying modifications to suit present and future requirements.

Access Level Foundation

The foundation for the Access Level feature is based on a custom field being added to the *User [sys_user]*, *Group [sys_user_group]*, and to the *Task [task]* table called *Access level [u_access_level]*. From this custom field, the functions to restrict or grant Access Level is permissible with only two business rules and two ACLs.

Access Level Override Role

The *access_level_override* role allows accounts to view records regardless of the *Access level* on the record.

Instructions:

1. Navigate to **User Administration > Roles**.
2. Click **New**.
3. Under the **Role New record** section, fill in the following fields:

- Name: access_level_override
- Description: Accounts can view any record regardless of access level.

4. Click **Submit**.

Access Level Manager Role

The *access_level_manager* role allows accounts to set the *Access level* field values on the *User [sys_user]* and *Group [sys_user_group]* tables.

Instructions:

1. Navigate to **User Administration > Roles**.
2. Click **New**.
3. Under the **Role New record** section, fill in the following fields:
   - Name: access_level_manager
   - Description: Accounts can set the access level fields on user and group records.
4. Click **Submit**.

Access Level Field - User

The custom *Access level [u_access_level] field will be applied to the OOTB *User [sys_user]* table. This field signifies the highest level of access the account can gain when it comes to records derived from the *Task [task]* table.

Instructions:

1. Navigate to **System Definition > Tables**.
2. Filter the listing where **Name** is **sys_user**.
3. Open the **User** record.
4. Under the **Columns** tab, click **New**.
5. Under the **Dictionary Entry New record** section, fill in the following fields:
   - Type: Integer
   - Column label: Access level
   - Column name: u_access_level
6. Under the **Default Value** tab, fill in the following fields:
   - Default value: 0
7. Click **Submit**.

Access Level - User ACL

The Access Control List (ACL) on the *Access level [u_access_level]* field for the *User [sys_user]* table is to make sure only accounts with the admin or access_level_manager role can set the *Access level [u_access_level]* value.

Instructions:

1. **Elevate role** to **security_admin**.
2. Navigate to **System Security > Access Control (ACL)**.
3. Click **New**.
4. Under the **Access Control New record** section, fill in the following fields:
    - Type: Record
    - Operation: write
    - Admin overrides: true
    - Name: User [sys_user]
    - Field: Access level
5. Under the **Requires role** section, add the **access_level_manager** role.
6. Click **Submit**.

Access Level Field - Group

The custom *Access level [u_access_level] field will be applied to the OOTB *Group [sys_user_group]* table. This field signifies the highest level a record can be when assigning the group a record. This is to prevent records from being re-assigned to a assignment group whose members cannot access the record.

Instructions:

1. Navigate to **System Definition > Tables**.
2. Filter the listing where **Name** is **sys_user_group**.
3. Open the **Group** record.
4. Under the **Columns** tab, click **New**.
5. Under the **Dictionary Entry New record** section, fill in the following fields:
    - Type: Integer
    - Column label: Access level
    - Column name: u_access_level
6. Under the **Default Value** tab, fill in the following fields:
    - Default value: 0
7. Click **Submit**.

Access Level - Group ACL

The Access Control List (ACL) on the *Access level [u_access_level]* field for the *Group [sys_user_group]* table is to make sure only accounts with the admin or access_level_manager role can set the *Access level [u_access_level]* value.

Instructions:

1. **Elevate role** to **security_admin**.
2. Navigate to **System Security > Access Control (ACL)**.
3. Click **New**.
4. Under the **Access Control New record** section, fill in the following fields:

- Type: Record
- Operation: write
- Admin overrides: true
- Name: Group [sys_user_group]
- Field: Access level

5. Under the **Requires role** section, add the **access_level_manager** role.
6. Click **Submit**.

Access Level Field - Task

The custom *Access level [u_access_level] field will be applied to the OOTB *Task [task]* table. This field signifies the access level an account must be at or higher to have access to the record.

Instructions:

1. Navigate to **System Definition > Tables**.
2. Filter the listing where **Name** is **task**.
3. Open the **Task** record.
4. Under the **Columns** tab, click **New**.
5. Under the **Dictionary Entry New record** section, fill in the following fields:
   - Type: Integer
   - Column label: Access level
   - Column name: u_access_level
6. Under the **Default Value** tab, fill in the following fields:
   - Default value: 0
7. Click **Submit**.

Access Level Before Query Business Rule

The Access Level before query business rule should allow only the following accounts access to a record:

- accounts with an equal to or grater than access level to the record
- accounts with the eyes_only_override role
- accounts with the admin role

Instructions:

1. Navigate to **System Definition > Business Rules**.
2. Click **New**.
3. Under the **Business Rule New record** section, fill in the following fields:
   - Name: Access Level - Query
   - Table: Task [task]
   - Advanced: true
4. Under the **When to run** tab, fill in the following fields:
   - When: before
   - Insert: false

- Update: false
- Delete: false
- Query: true

5. Under the **Advanced** tab, fill in the following fields:
    - Script:

```
(function executeRule(current, previous /*null when async*/ ) {

    if (gs.getUser().hasRole('admin') || gs.getUser().hasRole('access_level_override'
        return;
    }

    current.addQuery('u_access_level', '<=', gs.getUser().getRecord().getValue('u_acc

})(current, previous);
```

1. Click **Submit**.

Access Level Update Business Rule

The Access Level update business rule enforces the following rules:

- on ticket reassignment, the ticket's access level cannot be greater than the assignment group's access level
- one can only change the access level if a member of the assignment group
- only managers of the assignment group can lower access levels
- a access level cannot be increased above ones own access level

Instructions:

1. Navigate to **System Definition > Business Rules**.
2. Click **New**.
3. Under the **Business Rule New record** section, fill in the following fields:
    - Name: Access Level - Update
    - Table: Task [task]
    - Advanced: true
4. Under the **When to run** tab, fill in the following fields:
    - When: before
    - Insert: false
    - Update: true
    - Delete: false
    - Query: false
    - Filter Conditions:
        - Option: Access level
        - Operation: changes
        - Condition: OR
        - Option: Assignment group

- Operation: changes

5. Under the **Advanced** tab, fill in the following fields:
   - Script:

```
(function executeRule(current, previous /*null when async*/ ) {

    // admins can set any access level
    if (gs.getUser().hasRole('admin')) {
        return;
    }

    // retrieve helpful values into variables
    var current_access_level = parseInt(current.getDisplayValue('u_access_level'));
    var previous_access_level = parseInt(previous.getDisplayValue('u_access_level'));
    var user = gs.getUser().getRecord();
    var user_access_level = parseInt(user.getDisplayValue('u_access_level'));
    var current_group = current.assignment_group;
    var current_manager = current_group.manager;
    var current_group_access_level = parseInt(current_group.u_access_level);
    var previous_group = previous.assignment_group;
    var previous_manager = previous_group.manager;
/*
    // uncomment as needed
    gs.addInfoMessage('DEBUG:' +
        ' Current Access Level: ' + current_access_level +
        ' Previous Group: ' + previous_group.getDisplayValue('name') +
        ' Previous Access Level: ' + previous_access_level +
        ' User Access Level: ' + user_access_level +
        ' Current Group: ' + current_group.getDisplayValue('name') +
        ' Current Group Access Level: ' + current_group_access_level +
        ' Current Group Co-managers: ' + current_group.u_co_managers +
        ' Current User: ' + user.getValue('sys_id') +
        ' Index of: ' + current_group.u_co_managers.indexOf(user.getValue('sys_id')))
*/
    // on ticket reassignment, the ticket's access level cannot be greater than the a
    if (current_group.getValue('sys_id') != previous_group.getValue('sys_id') &&
        current_group_access_level < current_access_level) {
        gs.addErrorMessage('You must lower the access level to ' + current_group_acce
        current.setAbortAction(true);
    }

    // one can only change the access level if a member of the assignment group
    if (!gs.getUser().isMemberOf(current.assignment_group.getDisplayValue()) &&
        !gs.getUser().isMemberOf(previous.assignment_group.getDisplayValue()) &&
        user.getValue('sys_id') != current_manager &&
        user.getValue('sys_id') != previous_manager &&
        current_group.u_co_managers.indexOf(user.getValue('sys_id')) < 0 &&
        previous_group.u_co_managers.indexOf(user.getValue('sys_id')) < 0) {
        gs.addErrorMessage('You can only change access levels if you are a member of
        current.setAbortAction(true);
    }

    // only managers of the assignment group can lower access levels
```

```
    if (current_access_level < previous_access_level &&
        user.getValue('sys_id') != current_manager &&
        user.getValue('sys_id') != previous_manager &&
        current_group.u_co_managers.indexOf(user.getValue('sys_id')) < 0 &&
        previous_group.u_co_managers.indexOf(user.getValue('sys_id')) < 0) {
        gs.addErrorMessage('Only the manager (' + previous_manager.getDisplayValue()
        current.setAbortAction(true);
    }

    // a access level cannot be increased above ones own access level
    if (user_access_level < current_access_level) {
        gs.addErrorMessage('You can only increase the access level to ' + user_access
        current.setAbortAction(true);
    }

})(current, previous);
```

1. Click **Submit**.

Final Steps

The components for Access Level security restrictions are now in place. The next steps is to place governance around Access Level management and defining what Access Level values means to your organization.

Outro

You have now successfully setup the Access Level feature. Congratulations.