Site Tree and Permission Tree Requirements

13 May 2020

Context

Sites facilitate the distribution of an organisation’s Assets and other items at different locations, and they restrict a User’s access to items that are associated with the Sites for which the User has access rights. This is an extremely powerful feature as it allows Assets and other items to be managed in different locations using a single database

A Site is associated with every item in the On Key database (virtual or not virtual)

**Site Tree:** The site tree allows the administrator to define a hierarchy of sites. The administrator can link a site node to a right and state if the child sites should be included or not

Site Tree Requirements

The ability to:

* Configure “virtual sites” for different divisions so that items that are common to each of the different processes can be configured once.
  + In OK these are known as Virtual-Sites
* Configure “actual sites” for items that only occur at that *Site* and they do not occur at any other *Site*.
  + In OK5 these are known as Non-Virtual Sites
* visually see Virtual and Non-Virtual Sites
  + in OK5 a globe icon is used for Non-virtual sites and a folder icon is used for Virtual Sites
* define a hierarchy of sites
  + Add parent sites to the tree structure
  + add child sites, to the tree structure business rules apply (i.e. A Virtual Site may not be inserted below an Actual Site)
  + Expand and collapse parent record to view child records
* Cut records from one place in on tree and paste them somewhere else, business rules apply (i.e. A Virtual Site may not be inserted below an Actual Site)
  + Cut a child site and paste as a parent site
  + A site cannot be moved to one of its child sites
  + Maybe consider drag and drop for OK+
* search for site codes
  + consider having more search options in OK+
* Change site status to active/inactive
* Batch Update: Multiselect nodes and make them inactive/active
* Quick navigation feature
* Mimic Site: There should be a feature whereby the user can mimic the site tree. The whole site tree should then be copied to the Permissions tree

**Business Rules:**

* A Site must be specified for all items in the system.
* A virtual Site cannot be changed to a non-virtual Site, and vice versa.
* Assets, In-Place Assets and Asset Placeholders must be linked to a non-virtual Site.
* Work and Stock transactions must be posted to a non-virtual Site.
* The Site Lookup only lists Sites to which the User has access Rights.
* The number of non-virtual or actual Sites may not exceed the number of actual Sites specified in the On Key licence number.
* The number of actual Sites specified in the On Key licence number is updated every time a non-virtual Site is added.
* If the Currency for the Site is changed then the labour rates for the Site and Trades linked to the Site must be changed for the new Currency.

Lacking features

The site tree should not be a permission tree. It could be a convenient way to structure your permission tree but should not be the only mechanism. There will be scenarios where the site tree cannot enforce the correct level of control. Artificial methods would then need to be applied to extend it.

Actual sites should only be used on records where it makes sense. The introduction of Virtual Sites to the site tree was required to fill the limitation that the physical site tree had. It is used for “non-site” related records such as lookup data.

There are essentially 2 limitations to the current feature:

* **Additional filter criteria**: Workers that work across all sites but should be restricted based on responsibility. The site filtering does not cater for this and users are restricted to specific applications where the user interface applies the additional filter.

This is a big security risk. If a user gains access to web services or On Key, he/she will have access to records that they should not see. The filtering criteria should be applied on an API level. This will apply the filtering across all applications and interfacing solutions.

* **Virtual Site concept**: The concept of a virtual site is foreign for many users. Certain non-site related records should not be forced to have a site value just for the sake of data filtering. A Permission field should be introduced to manage data filtering.
* **Specific shortcomings**:
  + Work Order sites may not be changed. So essentially you cannot control the filtering.
  + Site Validation rules: On the site tree and the asset tree there are validations that restrict the allocation of site fields. This restricts the permission filtering feature.

Currently in On Key 5 the Site tree is used as the filter criteria. Each record in On Key contains a site field. The site value on these records determine if the user may see them or not.

The proposed solution is to rather use a **Permission Tree** as the filtering criteria.

This is required to:

* Facilitate the shortcomings of the Site Tree being used as filtering criteria:
  + User working across sites
  + Site fields on objects that do not require them
* Provide more flexibility and power to the filtering process
* Provide a mechanism to simplify the user rights process
* Strengthen the security of the system on an API level

To filter a list of data, a “filter field” needs to be linked to the records and to the user. This will then dictate if the user may see a record or not. These “filter fields” are called Permission Fields. They contain a Permission Value. These Permission Values will be maintained in a Permission Tree.

* Permission Field: The field on a record that contributes to the security data filter feature.
* Permission Value: The value within the Permission Field. This is based on a Permission Record which is defined within the Permission Tree
* Permission Record: These are records that can be created that are then used to set Permission Field values on records.
* Permission Tree: This is the area where Permission Records are maintained. They can be structured within a tree hierarchy.

**Permission Tree**

The permission tree should facilitate the Permission Records required for the data filtering feature.

Permission Tree Requirements:

* Provide a mechanism to create new Permission records
* Organise these records within a tree
* Be able to remove fields
* Be able to make fields inactive
* Be able to mimic the site tree
* Nice to have: Be able to select a field and view which users have access to it

**Use Cases**

* Visually display the security structure
* Assist in the management of user rights: When the permission record is linked to the user role, it can now have the feature – “from this point down”.
* The user should be able to create a simple Permission Tree.