Advanced Item Process Security

General

The security is “centralized” and will be applied to the entire system, both forms and search grids. The base security is set at basetype. Subtypes may overrule the basetype security to adapt different security schemes for different subtypes.

In types with the stage-gate process as Action, the *ActiveResource* vector role does not exist, instead, **Resource** vector role can be used, it is equal to ActiveResource and available for all types with resource column(s), also the types with no stage-gate process.

Elements defining security

The security is defined by using vector-roles. Vector-roles are high-performance bit-vectors defined by schema and ActiveDirectory. A list of vector roles is available in the menu SYSTEM > DIAGNOSTICS > VECTORROLES.

New vector roles may be defined by the schema, the following line in `custom.schema.xml` will add the Customer vector role, and map it to Active Directory user group Customers:

```xml
<vector-role name="customer" usergroup="customers" />
```

Type element

Default write security is set on type element:

```
<type name="action" write="superuser;manager;teammember;resources" />
```

The above write attribute on type element gives superuser, manager, teammember and resources write permission to all columns, however not columns that have explicit write attribute, only columns with no write attribute will follow any other write attributes including type write attribute. The manager role is the manager on workspace defined by *entity.manager* column or the manager on item defined by *item.manager* column.

The type element contains a ref-write attribute that controls which vector-roles is permitted to edit references. By default, this attribute follows the type write attribute:

```
<type name="part" ref-write="activeresource1" />
```

Process element

The process element specifies vector roles with permission to change step. Process element inherits write permissions from type element:

```
<process name="default" step-write="+resources" />
```

The above write attribute on process element adds write permission to all resources. This enables all resources to be able to change the process step.

Step element

Specific step write security is set on the process/step element, the write vector is only applied for fields in the step, and only when the step is active:

```
<step name="step1" write="+resource3" />
```

The above write attribute on step element adds write permission to resource3 so that resource3 also can edit columns contained in step1, but this just applies when process is in step1. All columns contained in this step section will be editable by resource3 when process is in step1. However, only columns that do not have an explicit write attribute. For example, the *item. item* column will still have write="0" even if it is contained in step1 and process is in step1.

The resource for the active step is added automatically to write vector, so the above line is equivalent to the line below:

```
<step name="step1" write="resource3;activeresource1" />
```

Field element

Specific field write security may be set on process/step/field element:

```
<step name="step3" >
  <field name="reviewby" write="resource1;manager;teammember" />
</step>
```

The above write attribute on field element gives write permission to column ReviewBy but just to Author (resource1), Manager and TeamMember, and only when process is in step 3 (review).

The line below with ‘+’ prefix will add the same resources, this will also give reviewers (resource3) write permission to ReviewBy column:

```
<step name="step3" >
  <field name="reviewby" write="+resource1;manager;teammember" />
</step>
```

Folder security

The virtual column *FolderSecurity* defines folder security. The column also lists current folder security and log output from the security apply process. The following is default action folder security, granting all resources and manager folder write permissions:

```
<type name="action">
    <column name="FolderSecurity">
      <folder name="default" relativepath="." write="resources;manager" />
    </column>
</type>
```

Action security samples

Standard security *(Default)*

The TS standard action security setup is below. *Superuser*, *Manager*, and *TeamMembers* have unrestricted write access to all columns and will be able to quickly correct data errors and push things forward. *DefaultTrustees* will be able to edit resources and thus add themselves as resources to speed things up:

```
<type name="action" write="superuser;manager;teammember;resources">
  <process name="default" step-write="+resources">
  <column name="resource1" write="DefaultTrustee" />
  <column name="resource2" write="DefaultTrustee" />
  <column name="resource3" write="DefaultTrustee" />
  <column name="resource4" write="DefaultTrustee" />
  <column name="resource5" write="DefaultTrustee" />
  <column name="resource6" write="DefaultTrustee" />
  <column name="resource7" write="DefaultTrustee" />
  <column name="resource8" write="DefaultTrustee" />
  <column name="resource9" write="DefaultTrustee" />
  <column name="resource10" write="DefaultTrustee" />
  <column name="resource11" write="DefaultTrustee" />
  <column name="resource12" write="DefaultTrustee" />
  <column name="resource13" write="DefaultTrustee" />
  <column name="resource14" write="DefaultTrustee" />
</type>
```

Loose security

Out-of-the-box the action module has been set up to support smaller organizations with not so formal requirements, their focus is on speed and improving the situation rather than having an absolute formal requirement.

A loose security scheme used in some organizations for some processes will be to give all resources write permissions to all columns in all steps, this could be set on the type element and then there probably will be no need to set write attributes at any steps. Any user will be able to add themselves as a resource and thus get write permission to all columns in process:

```
<type name="action" write=" superuser;manager;teammember;resources">
  <process name="default" step-write="+resources" />
</type>
```

Strict security

In some organizations with strict compliance to regulations, tighter security is required. The following will give Superuser, Manager and the resource in the active step (*activeresource*) write permission to process. The *resources* value has been removed from the above [loose out-of-the-box setup](#).

```
<type name="action" write="superuser;manager;teammember">
  <process name="default" step-write="" />
</type>
```

As mentioned earlier, the resource for the active step will automatically be added. For example, when step3 is active `write="ActiveResource3"` will automatically be set on the step, however only when step is active.

Document security samples

Standard security

Document has a stage-gate process similar to *Action*. The `item.status` column is the process step-index, and the resource columns are defined by the process step elements. This is the skeleton of the document type, which is inherited by the Part and Device types:

```
<type name="doc" write="adminwrite;activeresource1">
  <process name="default" stepcolumn="status">
```

```
    <step name="step1" resourcecolumn="author" write="superuser" />
    <step name="step2" resourcecolumn="author">
      <field name="reviewby" />
      <field name="approveby" />
      <field name="copyto" />
    </step>
    <step name="step3" resourcecolumn="reviewby" />
    <step name="step4" resourcecolumn="approveby" />
    <step name="step5" resourcecolumn="" />
  </process>
</type>
```

The document schema above shows how the fields *reviewby*, *approveby*, and *copyto* are writeable by author in step2 *(Freeze)*. AdminWrite (and ActiveResource1) - defined by write attribute at type element - will still have write access.

Permitting changing reviewers in review state

The sample below when placed in `custom.schema.xml` will give *author*, *Manager* and *TeamMember* write permission to *reviewby* column when document is in review state (step3):

```
<type name="doc">
  <process name="default">
    <step name="step3" >
      <field name="reviewby" write="+resource1;manager;teammember" />
    </step>
  </process>
</type>
```

Permissions for change

The permission for clicking on the CHANGE button can be set to virtually any combination of Highstage vectors roles.
It is only *author* who is invited to make change since only author will have CHANGE button visible at document web page. However other vector roles may have permission to change from Advanced section if user is running as *AdvancedUser*.

By default, *DefaultTrustees* can perform change. *DefaultTrustees* is a parameter which normally would be set to an Active Directory group containing all employees.

Many companies prefer that all users in *DefaultTrustees* can make a document change on documents they have read access to so that the employees can be proactive and help the company in the tough global competition. However, for example FDA regulations and other regulation may have a problem with that.

The Advanced section CHANGE button security can be monitored by selecting Trace security from System button on top of web page *(remember to select stop trace when done)*.

Default settings on *Change write permissions:*

Permission access	#	Roles
user	2000000021	`Everyone`;`AdvancedUser`;`DefaultTrustee`
read	00000000	-
write	20000020D0	`SuperUser`;`Manager`;`TeamMember`;`Resource1`;`DefaultTrustee`

> The vector roles associated with *write* are able to click CHANGE under the advanced section

The roles are:

`SuperUser`, `Manager` and `TeamMember` on workspace, `Resource1` *(work resource in step 1 is the same as authors)*, and `DefaultTrustees`.

The permissions may be restricted by removing some of the vector roles from *write*.

By adding the following XML snippet to custom.schema.xml `DefaultTrustees` will no longer be able to perform change from the Advanced section:

```
<type name="doc ">
    <feature name="config-manager" change-write="resource1;manager;teammember;superuser"/>
</type>
```

Other vector roles could be removed as well, but `ressource1` and `SuperUser` should not be removed.

It will normally be a requirement that `SuperUser` can make a change in case the author is on vacation, or sick leave, or no longer is employed by the company.

Part security

Standard security

The security scheme is inherited from document standard security.

Allow authors to change RoHS status at any time

By giving the RoHS column a write attribute prevents permissions from being modified by process state. Placing the following snippet in `custom.schema.xml` will give `Author`, `Manager` and `SuperUser` *write* access to RoHS status at any time, and even if the part is approved:

```
  <type name="part">
    <column name="rohs" write="resource1;manager;superuser "/>
  </type>
```

User levels

UI is adapted to the user level. Depending on user level the user has more or less UI, and more or less permissions.

In most cases, the user will have to switch to a higher user level to have more features. However, some features will be available as long as the user can switch. For example, the RESET button is always available to a member of `SuperUser` (and above `AdminRead` and `AdminWrite`), even when running as a basic User level. Also, the `ViewAs` function will be available to Admin members even if Admin is running as basic User, but observe that when running as user-level less than `AdminRead` double security will be applied, meaning that only data visible to both users will be displayed.

When running as `AdminRead` or `AdminWrite` the `ViewAs` function will show all data visible by the other user.

The schema controls read/write ability on schema elements. To check the actual abilities, go to System>BASETYPE/XML and search for example for `“SuperUser”` to find the places that grant `SuperUser` permissions, by built-in or custom schema.

User

The basic `User` level strips down the UI to an absolute minimum. This user level is the default user level and it is optimized for first-time users and rare users since it reduces complexity to a minimum.

AdvancedUser

The `AdvancedUser` level inherits all abilities from basic `User` and adds more. The `AdvancedUser` level reveals more UI. All basic users can switch to `AdvancedUser` level and back to basic `User`.

`AdvancedUser` does not have more data access permissions than a basic user unless customization has changed this.

`AdvancedUser` has the following additional UI:

1. Access to advanced sections on items.
2. Access to search grid design section.

SuperUser

The `SuperUser` inherits all abilities from `AdvancedUser` and adds more:

1. Change resources on items on which the `SuperUser` is not a resource. For example, change the author on a document even if `SuperUser` is not an author.
2. See process-index on each step to help develop and debug the action process.
3. Links on the grid column header and form field title for seeing more information about the element, for example, schema and database information.
4. If `ts_mail schema` has been included then `SuperUser` can delete all obsolete mails, to clean up and make more disk space, and thus solve spam issues.
5. Kill file locks. If a user computer crashes while editing a document file, then a file lock may prevent other authors from working. The `SuperUser` can kill file locks.
6. Access to `RESET` button, to restart the web application, to flush all caches, or to load schema after making schema changes.

AdminRead

The `AdminRead` inherits all abilities from `SuperUser` and adds more:

1. Read access to all data in the database, however no additional permissions to file storage. The `AdminRead` will be able to see the existence of all documents, but `AdminRead` will not be able to read more document files compared to the basic User level.
2. Enables `ViewAs` which mimics what other users can access and see in Highstage.
3. Access to the system menu. This can be accessed from the side navigation menu.

AdminWrite

The `AdminWrite` inherits all abilities from `AdminRead` and adds more:

1. Modify system parameters.
2. Modify most columns unless column write attribute is 0.
3. Install new versions.

Highstage