1 Overview

ABC's has complex security requirements for the Claims+ Workers Comp Claims. These requirements are met by configuring Access Control List ACL settings, configuration of other configuration files, writing new rules and creating admin data to introduce security specific groups, zones and users. WC claims are assigned to TPA s and managed by them as well as by ABC employees. Each claim is associated to a PEO, PEO client, PEO Client Locations, Partner Agents and Co-broker. These claims can take different confidentiality levels (security level). Access to the claim also changes based on the confidentiality level, permission to edit/view to the claims is based on the groups. For example only TPAs and ABC employees can edit the claims all other group users will have View access for their claims. There are other security requirements based on Exposure type, Subrogation Status, loss type etc. These are discussed in detail in the requirement section.

Claim Center offers two types of security

1. Screen Security: Functionality to control what a user will actually be able to do inside of ClaimCenter
2. Data Security: Functionality to control what a user will actually see inside of ClaimCenter.

Screen Security will be implemented by means of the Administration Data definition. Specific tasks that will be included are the definition of Roles, associating System Privileges to the Roles, and assigning Roles to all of the Users defined in Claim+. Each user defined in Claim+ must have a role associated with their profile.

Data Security will be implemented through the use of Security Zones and Access Profiles. There are 7 different "Groups" where Data Security will come into play for the Claim+ system.
In this document we discuss the configuration details, rules and admin setup and maintenance to meet the Claim level security requirements. Design is considered with some assumptions that are mentioned in the assumptions and issues section. At the end of the design document I have provided use case scenario that highlights the functionality to be achieved by the configuration.

2 Requirements:

The following are the Workers Compensation groups participate in claim level security. All non WC claims groups are maintained separately.

1. PEOs
2. PEO Clients
3. PEO Client Locatopns
4. TPA
5. ABC Employees
6. Partner agents
7. Co-broker

- PEO: At the PEO level, users defined as part of a PEO group should see only the Claims that belong to their Clients. They should be able to only view the claim and will not have edit capability for existing claims
- PEO Clients: At this level, users associated with a specific PEO client group should only be able to see the Claims that belong to their client organization. These users should be able to only view the claim file, and they will not have edit capability for existing claims
- At the PEO and PEO Client level, the system should allow them to create claims, but once a claim is initially created, it is view only and it cannot be edited by any user at the PEO or PEO Client level.
- TPA: The TPA user should view claims belong to their group. Adjuster should be able to edit the claim assigned to him/her.
- Partner Agents: Partner agents need to view claim data for their Insured's only.
- ABC Employees: ABC has employees that will be users of Claim+. It is believed that at the non-manager/executive levels, it will be necessary to have users see only claims within their line of business (WC and Non-Comp). Selected managers and executives may have access that spans lines of business.
- Handling TPAs and ABC employees should have regular access to the following claims. No other user should have access to these claims.
    1. Coverage
    2. Employee claim
    3. SIU
    4. Sensitive" claims The only people that should have access to are handling adjuster, supervisors, managers, ABC
- **"Loss Type"** Based Security.

  Claim access for ABC Employees needs to work as follows.

  The criterion is "Loss Types" for the Claim: WC Employees (Member of the 'Administration' or ' Analysts' Group): can see only WC (standard comp) or WCEX (WC Excess/Buffer) Claims

  Non-WC Employees: They can only see AUTO, PR, or GL Claims.
- **Co-Brokers group** to view claim for their Insured's only.
- **Subrogation if** the claim Subrogation Status is open or Review then only specific users (user part of the subrogation group) should have access to the claims.
- **PEO Client Location:** A PEO Client can have different PEO client locations. Claims belonging to specific locations should be used by those location users. User from different PEO client locations should not have access. Parent PEO client users can see all the claim under their PEO Location group
- **WC Excess claims assigned to** "ABC Claims department" should not be viewed by any other users from other groups other than "ABC Claims department" group.
- **Employer Liability Adjusters,** If a claim has an Exposure of type "Employer Liability" then users in the "Employer Liability Adjusters" group should be able to access and edit that claim. This is applicable for all the claims irrespective of Confidentiality type(unsecured ,coverage, SIU, employee and sensitive)

3 Assumptions

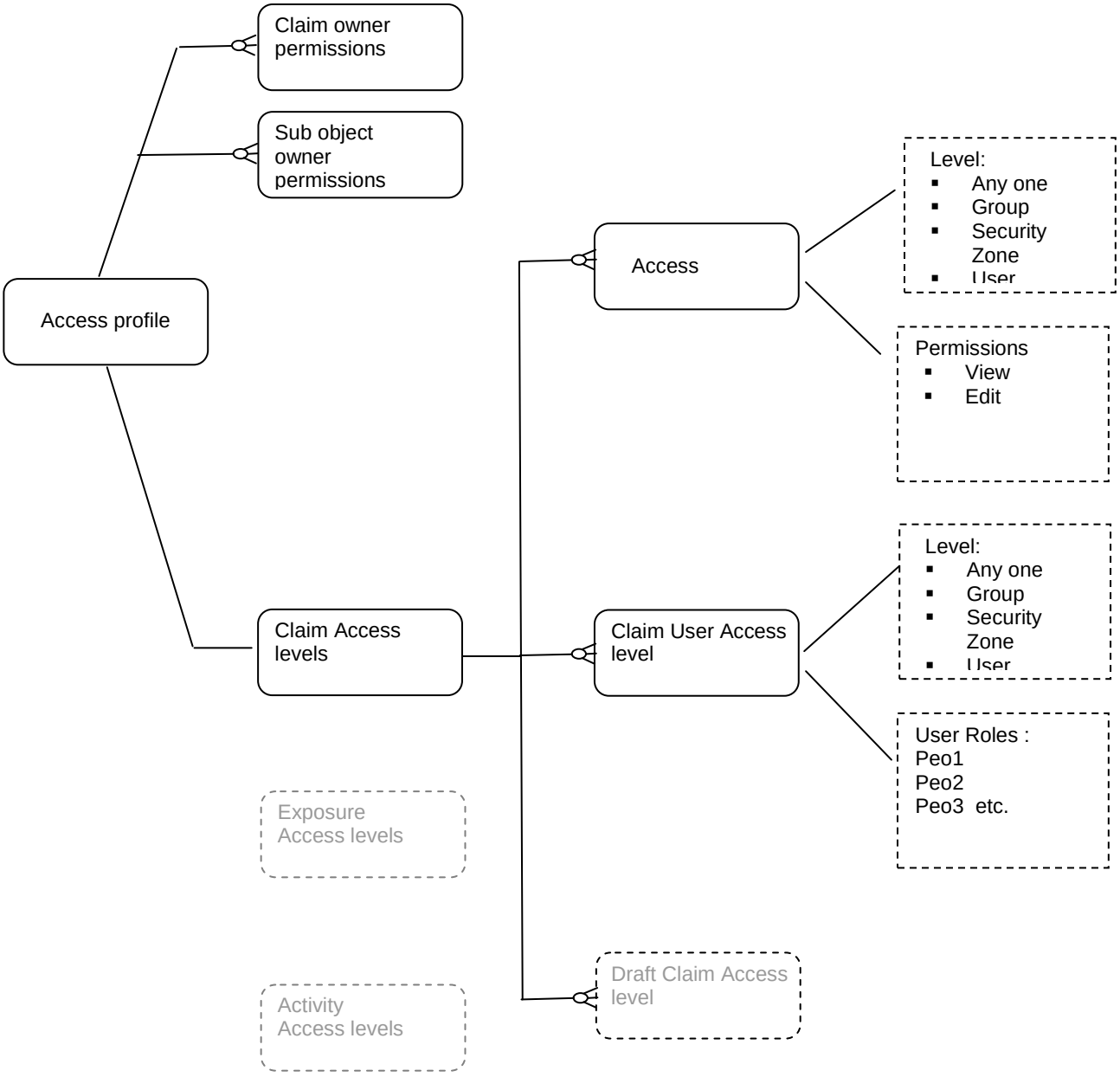
- There are no specific security requirements for any other user actions like payment or assignment.
- There are no security requirements for Exposure and Activities Out of the box configuration will be maintained.

4 Issues

- Exposure and Activity associated to Claims will cause security holes! But for the case of WC Claims exposures are assigned only to the claim owner.

5 Security structure ABC context

Using access control mechanism, we can subcategorize an object type and then restrict object access by these subcategories. We can apply access control to claim, (exposure, and document objects). This diagram explains the ClaimCenter security structure and how permissions interact with access control.



6 Design

- To restrict claim access across the different groups separate security zones are created and associated to each TPA group.
    - Attenta Zone
    - Intercare Zone
    - Unisource Zone

  Following Zone is for the non WC groups
    - Auto_Property_GL Zone

  For the specific case of WC Excess claims being assigned to “ABC Claims department” users; I have created a security zone to maintain the security across “ABC Claims department” and other groups.
  All the groups under “ABC Claims department” should be associated with this security Zone.
    - ABC_Claims_Dept zone

- PEO and PEO Client group users will access the claims through claim user Access level. We associate claims with PEO and PEO Clients by creating a user with PEO/PEO Client specific role in the “Parties involved”. And these roles are view only. This is achieved in Pre update rules.
- To provide access security for the “PEO Client Locations” under PEO Client, (as of now there exists one PEO client and there are 23 locations under that client.) I have created “PEO Client Location Groups” under PEO client Parent group. In the pre update rule I check for the “Location Code” on the “Loss Detail page” to decide on the “PEO Client Group”. Then I add a security user from that group to the “Parties Involved” section of the claim. By doing this any user in the “PEO Client Location group” will have the access to this claim. Users from other client Location groups will not have access.
  When users change the “Location Code” then I will remove the old Security user from the “Parties Involved” section and add new security user from the new “PEO Client Location Group” so access is changed to new location group!
- Partner Agents will access the claims through Claim user Access level. We associate claims with Partner Agents by creating a user with specific role in the Parties involved. This is achieved using Pre update rule.
- Co brokers will access the claims through Claim user Access level. We associate claims with Co brokers group by creating a security user from Co-Broker group with specific role in the Parties involved. This is achieved using Pre update rule when there exists a co broker on the Policy associated with the claim.
- A new “ABC Subrogation Adjusters” group is created with a dummy security user associated. This is to provide the access to the claims with subrogation status Open/view. Any user who needs access to subro claims should be made part of this group. The Dummy security user will be added to the parties involved section of the claim using pre_update rule.
- A new “**Employer Liability Adjusters”** group is created with a dummy security user associated. This is to provide the access to the claims with Exposure of type” Employer Liability”. Users in this group will be able to access the claim. The Dummy security user with Role “ABC_EL_Role_Edit” will be added to the parties involved section of the claim using pre_update rule.
- **Security Groups** To provide the necessary access to ABC users within and across security zones I defined a security group in each security zone, as of now I have 4 security zones defined and three security groups.

  ABC Security Group Attenta Zone
  ABC Security Group Intercare Zone
  ABC Security Group Unisource Zone

- To achieve security across “Loss Type” (refer the requirement) I have created separate Security Zone **“Auto_Property_GL Zone”** for the loss types AUTO, PR, or GL Claims.
  All the claims with above mentioned loss type are assigned to group/user in this security zone.
  There are no specific Security Level requirements for these claims.

- To facilitate the Sensitivity claim requirement of access to only handling Adjuster, I made AccessLevel level="user”. To provide access to ABC employees I have created following security groups users whoever needs access to Sensitive clams need to be added to this group.

  ABC Security Sensitivity Group

- To meet the requirements for other Security levels Coverage, Employee claim, SIU and unsecured claims security_config.xml is configured accordingly. It is discussed further in configuration section under securityconfig.xml.

- I have added a new Group Type “ABCsecurity” in the GroupType.xml type list, to separate the security groups. All security Groups are created of type “ABCsecurity” This may help in future enhancements.

- To have access across the TPA security zones, ABC Employees will be associated with all the 4 Security groups along with the group they belong to.

- The ability to add a User and Role in the Parties Involved screen will affect security access. So the ability to add someone in this screen is restricted. Only the Claim supervisors at ABC will have the “Manage claim users” privilege. We need to create a separate role (in admin) for this permission and grant it to individual users that need to have this permission.

- Administration and maintenance is discussed in separate section.

7 Conversion Claims Security Design

The security for the converted claim is same as the regular claims created through Claim + application. All converted claims should to run through security related rules so that security is effective on them.
After conversion of claims from legacy system and ported in to Claims+, we have to trigger a one time rule that calls claim validation and pre_update rules on each converted claim. This is done through the Claim Exception Rule: “ABC-CLMEXPT-001: Claim Exception rule for converted claims”. We need to trigger this Exception rule manually since it is one time requirement. It can be scheduled also. This rule in turn triggers validation rules. The Claim validation rule “ABC-CLMVAL-0001: Claim Validation rule for converted claims”.
I update claim.FireDeptInfo=claim.FireDeptInfo+" ." This in turn causes claim pre-update rules to fire. These rules are discussed in “Rules” section.
I identify the converted claims with the fact that claim.LoadCommandID is not null in case of converted claims. This field is populated by conversion process during conversion.Claims created via Claim+ application will have null in this field. In the exception Rule# 9 I identify the claims that are not null in claim.LoadCommandID and trigger the validation. To trigger exception rules we need to either schedule or manually run the exception procedure using maintenance_tools utility “claimexception “process

1. Start a command prompt
2. change directory to workspace\ABC_ClaimCenter\toolkit\bin
3. run the following command

maintenance_tools **-password** *password* [**-server** *url*] [**-user** *user*] [**-help**] { **-D** *name*=*value* |
-getdbstatisticsstatements | **-processstatus** *process* | **-startprocess** *process* |
-terminateprocess *process* **| -markclaimforpurge** *claimnumber* **|**
-markclaimsforpurge *filename* **| -whenstats}**

Example : maintenance_tools.bat -password gw -user su -server http://localhost:8081/cc -startprocess claimexception

Once this process is run all the converted claims till that date are now part of the security.

4. Disable “ABC-CLMVAL-0001: Claim Validation rule for converted claims” and “ABC-CLMEXPT-001: Claim Exception rule for converted claims” after completion of conversion claim process.

8 Configuration

This section provides details of the entire required configuration to implement the design discussed above.

- **Parameters in config.xml**

  These parameters affect the security and permissions used to access claims and perform other operations.

Parameter Name	Parameter Description	Default value
UseACLPermission	Specifies whether access control is used	true
EnableDowlinePermissions	Specifies whether a user has access to all claims to which any user or group they supervise or manage also has access.	true
RestrictSearchesToPermittedItems	Specifies whether searches include only items that the user can view.	true

- **Configuring Security Zones**
  We use the INSTALL_DIR\config\import\security-zones.xml file to manage security zones.

```
<?xml version="1.0"?>
<import xmlns:xsi="http://www.w3.org/2001/XMLSchema-
instance"

xsi:noNamespaceSchemaLocation="http://www.guidewire.com/cc/
import/cc_import.xsd">
<SecurityZone public-id="default_data:1">
      <description>Default Security Zone</description>
      <name>Default Security Zone</name>
      </SecurityZone>
      <SecurityZone public-id="default_data:2">
      <description>Attenta Zone</description>
      <name>Attenta Zone</name>
</SecurityZone>
<SecurityZone public-id="default_data:3">
      <description>Intercare Zone</description>
      <name>Intercare Zone</name>
</SecurityZone>
<SecurityZone public-id="default_data:4">
      <description>Unisource Security Zone</description>
      <name>Unisource Zone</name>
```

```
</SecurityZone>
</SecurityZone>
<SecurityZone public-id="default_data:5">
      <description>Auto_Property_GL Zone</description>
      <name>Auto_Property_GL Zone</name>
</SecurityZone>
 </import>>
```

- **UserRole.xml**

  I have created four user roles for Edit and view. These roles will provide the user access to the claims using <ClaimUserAccessLevel> in the ACL. These roles are assigned to specific group like PEO, PEO Client, Partner agent groups by creating a user and role in Parties involved section of the claim. The decision of creating a user specific to that group is made based on the value in “Insured”, “Partner agent” and “PEO Client” fields in the Policy. For sensitive claims user/role will be added in the Parties involved section. This is achieved by the Pre update rules. Please refer rules in rules section.

  1. ABC_Sec_Role_Edit
  2. ABC_Sec_Role_View
  3. ABC_Sens_Role_Vw
  4. ABC_Sens_Role_Ed

  These roles provide access to claims that cross security zones.

```
<?xml version="1.0" encoding="UTF-8"?>
<typelist name="UserRole"
desc="Roles users can have on an assignable object"
final="false">
<typecode code="primaryadjuster" name="Primary Adjuster"
      desc="Primary adjuster for the claim" retired="true">
      <category typelist="UserRoleConstraint" code="objectowner"
/>
      <category typelist="UserRoleConstraint"
code="claimexclusive" />
</typecode>
<typecode code="subrogationowner" name="Subrogation Owner"
      desc="User in charge of subrogation">
<category typelist="UserRoleConstraint" code="expexclusive" />
</typecode>
<typecode code="relateduser" name="Related User"
      desc="Other user associated with the claim" />
<typecode code="siuinvestigator" name="SIU Investigator"
      desc="SIU Investigator associated with the claim">
      <category typelist="UserRoleConstraint"
code="claimexclusive" />
</typecode>
<typecode code="siu" name="SIU"
      desc="SIU associated with the claim" />
<typecode code="police" name="Police" desc="Police" />
      :
      :
      :
      <typecode code=" ABC_Sec_Role_Edit " name="ABC
Security Edit Role " desc="Role to edit claims" />
<typecode code=" ABC_Sec_Role_View " name=" ABC Security
View Role " desc="Role to view Claims" />
<typecode code="ABC_Sens_Role_Vw"
name="ABC_Sensitive_Role_View" desc="Role to View
Sensitivity claims" />
<typecode code="ABC_Sens_Role_Ed"
name="ABC_Sensitive_Role_Edit" desc="Role to edit
Sensitivity claims" />
```

```
    <typecode code="ABC_EL_Role_Edit"
    name="ABC_EL_Role_Edit" desc="Role to edit claims with
    Employer Liability Exposure"/>
          :
          :
          :
    </typelist>
```

- **Access Control Security-config.xml changes**

  Claim Center gives great flexibility to control the access that users have to view or modify claims. While roles and permissions determine what a user can do, access control determines on which claims they are allowed do it. Roles and access control work together to decide whether a user can perform a task; to be successful, the user requires both the appropriate role and also proper access to the claim data.
  The accessProfile portion of the security-config file which defines claim access is configured as follows:

```
  <AccessProfile securitylevel="unsecuredclaim">
    <ClaimAccessLevels>
      <AccessLevel level="securityzone" permission="view"/>
      <AccessLevel level="securityzone" permission="edit"/>
      <DraftClaimAccessLevel level="securityzone"/>
      <ClaimUserAccessLevel role="subrogationowner" level="user"
permission="view"/>
      <ClaimUserAccessLevel role="subrogationowner" level="user"
permission="edit"/>
      <ClaimUserAccessLevel role="ABC_Sec_Role_View" level="group"
permission="view"/>
      <ClaimUserAccessLevel role="ABC_Sec_Role_Edit" level="group"
permission="edit"/>
      <ClaimUserAccessLevel role="ABC_Sec_Role_Edit" level="group"
permission="view"/>     <ClaimUserAccessLevel role="ABC_EL_Role_Edit"
level="group" permission="view"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="edit"/>
      <ClaimUserAccessLevel role="relateduser" level="user"
permission="view"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>


  <AccessProfile securitylevel="coverage">
    <ClaimOwnPermission permission="ownsensclaim"/>
    <SubObjectOwnPermission permission="ownsensclaimsub"/>
    <ClaimAccessLevels><!--
      <AccessLevel level="group" permission="view"/>
      <AccessLevel level="group" permission="edit"/>
      -->

      <AccessLevel level="securityzone" permission="view"/>
      <AccessLevel level="securityzone" permission="edit"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="view"/>
```

```
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="edit"/>

      <DraftClaimAccessLevel level="group"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>

  <AccessProfile securitylevel="employeeclaim">
    <ClaimOwnPermission permission="ownsensclaim"/>
    <SubObjectOwnPermission permission="ownsensclaimsub"/>
    <ClaimAccessLevels><!--
      <AccessLevel level="group" permission="view"/>
      <AccessLevel level="group" permission="edit"/>
      -->

      <AccessLevel level="securityzone" permission="view"/>
      <AccessLevel level="securityzone" permission="edit"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="view"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="edit"/>

      <DraftClaimAccessLevel level="group"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>

    <AccessProfile securitylevel="SIU">
    <ClaimOwnPermission permission="ownsensclaim"/>
    <SubObjectOwnPermission permission="ownsensclaimsub"/>
    <ClaimAccessLevels><!--
      <AccessLevel level="group" permission="view"/>
      <AccessLevel level="group" permission="edit"/>
      -->

      <AccessLevel level="securityzone" permission="view"/>
      <AccessLevel level="securityzone" permission="edit"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="view"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="edit"/>
```

```
      <DraftClaimAccessLevel level="group"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>
    <AccessProfile securitylevel="sensitiveclaim">
    <ClaimOwnPermission permission="ownsensclaim"/>
    <SubObjectOwnPermission permission="ownsensclaimsub"/>
    <ClaimAccessLevels>
      <!--<AccessLevel level="group" permission="view"/>
      <AccessLevel level="group" permission="edit"/>
      -->

      <!--<AccessLevel level="securityzone" permission="view"/>
      <AccessLevel level="securityzone" permission="edit"/>

      -->
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
      <ClaimUserAccessLevel role="subrogationowner" level="user"
permission="view"/>
      <ClaimUserAccessLevel role="subrogationowner" level="user"
permission="edit"/>
      <ClaimUserAccessLevel role="ABC_Sens_Role_Vw" level="group"
permission="view"/>
        <ClaimUserAccessLevel role="ABC_Sens_Role_Ed" level="group"
permission="view"/>
      <ClaimUserAccessLevel role="ABC_Sens_Role_Ed" level="group"
permission="edit"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="view"/>
      <ClaimUserAccessLevel role="ABC_EL_Role_Edit" level="group"
permission="edit"/>

      <DraftClaimAccessLevel level="group"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>

  <!--

  <AccessProfile securitylevel="fraudriskclaim">
    <ClaimOwnPermission permission="ownsensclaim"/>
    <SubObjectOwnPermission permission="ownsensclaimsub"/>
```

```
    <ClaimAccessLevels>
      <AccessLevel level="group" permission="view"/>
      <AccessLevel level="group" permission="edit"/>
      <DraftClaimAccessLevel level="group"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>

  -->

  <!--

  <AccessProfile securitylevel="underlitclaim">
    <ClaimOwnPermission permission="ownsensclaim"/>
    <SubObjectOwnPermission permission="ownsensclaimsub"/>
    <ClaimAccessLevels>
      <AccessLevel level="group" permission="view"/>
      <AccessLevel level="group" permission="edit"/>
      <DraftClaimAccessLevel level="group"/>
    </ClaimAccessLevels>
    <ActivityAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ActivityAccessLevels>
    <ExposureAccessLevels>
      <AccessLevel level="user" permission="view"/>
      <AccessLevel level="user" permission="edit"/>
    </ExposureAccessLevels>
  </AccessProfile>

  -->
  <DocumentPermissions>
    <DocumentAccessProfile securitylevel="unrestricted"/>
    <DocumentAccessProfile securitylevel="sensitive">
      <DocumentViewPermission permission="viewsensdoc"/>
      <DocumentEditPermission permission="editsensdoc"/>
      <DocumentDeletePermission permission="delsensdoc"/>
    </DocumentAccessProfile>
  </DocumentPermissions>

</SecurityConfig>
```

I maintained exposure and activity access levels OOB, however as completed activities would provide access to users who had completed activities on the claim (product default).That results (user access via completed activities) would violate ABC security zone access requirements. When Exposure and activity Security requirements are defined this should be confirmed. If exposure and activity security requirements are not decided then we can disable the activity/exposure access level by commenting out or deleting the section.

- **SystemPermissionsType.xml**

OOTB

- **DocumentSecurityType.xml**

OOTB

- **ClaimAccessType.xml**

OOTB

- **ClaimSecurityType.xml**

Commented out following for ABC

```
  <!--<typecode code="fraudriskclaim" name="Fraud risk" retired="true"
      desc="Permission to see a claim with a high fraud risk">
  </typecode>
     -->
  <!--<typecode code="underlitclaim" name="Under litigation"
retired="true"
      desc="Permission to see a claim that is currently under
litigation">
  </typecode>
     -->
```

- **ResourceContext.xml**

  Add the following type code in this file.

```
  <typecode code="ABCsecurity" name="ABC Claim Level Security Rules"
desc="ABC Claim Level Security Rules"/>
```

Logon to Studio and select this ResourceContext to all security rules. So that when we import or export we can use this context.

- **GroupType.xml**

```
  <!-- ABC CUSTOM VALUES BELOW THIS LINE -->
  <typecode code="TPA" name="TPA" desc="All TPA Master Group"/>
  <typecode code="tpa_attenta" name="Attenta TPA" desc="Attenta TPA
Master Group"/>
  <typecode code="tpa_intercare" name="Intercare TPA" desc="Intercare
TPA Master Group"/>
  <typecode code="tpa_unisource" name="Unisource TPA" desc="Unisource
TPA Master Group"/>
  <typecode code="intercare_roseville" name="Intercare Roseville"
desc="Intercare Roseville"/>
  <typecode code="intercare_irvine" name="Intercare Irvine"
desc="Intercare Irvine"/>
  <typecode code="tpa_team" name="TPA General Team" desc="TPA General
Claims Team"/>
  <typecode code="tpa_losttime" name="TPA Lost Time Team" desc="TPA
Lost Time Team"/>
  <typecode code="tpa_medical" name="TPA Medical Team" desc="TPA
Medical Team"/>
  <typecode code="tpa_ops" name="TPA Operations Team" desc="TPA
Operations Team"/>
  <typecode code="tpa_other" name="TPA Other Team" desc="TPA Other Team
(Misc) - Used to establish a Reporting hierarchy"/>
  <typecode code="tpa_allclaims" name="TPA All Claims" desc="TPA All
Claims"/>
  <typecode code="partneragent" name="Partner Agent" desc="Partner
Agent"/>
  <typecode code="it" name="ABC IT" desc="ABC IT Team"/>
  <typecode code="uw_ops" name="ABC Underwriting Ops" desc="ABC
Underwriting Ops"/>
  <typecode code="actuary" name="ABC Actuarial" desc="ABC Actuarial
Team"/>
  <typecode code="inactive" name="ABC Inactive Users" desc="ABC
Inactive Users"/>
  <typecode code="PEO" name="PEO Group" desc="All ABC PEO's"/>
  <typecode code="peo_org" name="PEO Organization" desc="PEO
Organization"/>
  <typecode code="ABCclaims" name="ABC Claims Department" desc="ABC
Claims Department"/>
  <typecode code="ABC_wc" name="ABC WC Claims" desc="ABC WC Claims"/>
  <typecode code="ABC_pcl" name="ABC PCL Claims" desc="ABC PCL
Claims"/>
  <typecode code="wcadmin" name="ABC WC Administration" desc="ABC WC
Administration"/>
  <typecode code="wcanalysts" name="ABC WC Claim Analysts" desc="ABC WC
Claim Analysts"/>
  <typecode code="partner_agents" name="ABC Partner Agents" desc="ABC
Partner Agents Master Group"/>
  <typecode code="partner" name="ABC Partner Agent" desc="Individual
ABC Partner Agent"/>
  <typecode code="peo_client" name="PEO Client Organization"
desc="Individual client companies of PEO Groups"/>
  <typecode code="ABC_Security" name="ABC Security Group" desc="Group
created to enforce Claim Level Security - Adding users to this group
```

```
will grant that User the security afforded by the Group's Security
Zone"/>
  <typecode code="cobroker" name="Co-Broker Group" desc="Co-Broker
Group"/>
  <typecode code="peo_client_loc" name="PEO Client Locations"
desc="Individual Location for a PEO Client"/>
          <!-- END ABC CUSTOMIZATIONS -->
```

9 Rules

- **Claim Pre-Update Rules** The following rules are part of security solution. Pre update rules are used to add Security dummy user and role to “Parties involved” section on the claim and associate claims with specific group users.
  All the following rules are created part of the Rule set “**ABC-Claim Level Security**”
  This rule set has the **Conditions:** claim.LossType=="WC"

1. **ABC-CLMPREUP_001 - Assign a user and role specific to Policy code(for PEO and PEO Clients)**
   Conditions:

```
//This rule assigns a Dummy user from a specific PEO group  to claims
//users from that group can access this claim
claim.Policy.Ex_PlanCode!=null
and claim.ValidationLevel=="newloss"

//By enabling following condition it provides security across PEO and
PEO Client.
//This can be deleted if not used.
//and (claim.Policy.ex_PEOClient == claim.Policy.insured
//OR claim.Policy.ex_PEOClient== null
//)
```

   Actions:

```
/*****************************************************************************
// ABC Phase 1
// DESCRIPTION: This code creates user role in Parties involved for
specific
//            PEO group based on the PEOPlan Code.The Dummuy
Security Users
//            are created with PEO<Plancode> suffix in first name
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/03/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {
    var plancode = "PEO"+ claim.Policy.Ex_PlanCode
    var query2 = find ( usr in user.Contact where usr.firstName contains
plancode);
    var z = query2.getFirstResult()
    var usrid=z.PublicID
var query1 = find ( grp in group.Users.User.Contact where grp.FirstName
contains plancode);
    var y = query1.getFirstResult()
    var grpid=y.PublicID
if (usrid !=null)
{       Claim.assignUserRole( User(usrid ), Group(grpid),
"ABC_Sec_Role_View" )  }
```

```
    }
  catch (e) {
   util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
          }
```

2. **ABC-CLMPREUP_002 :Rebuild ACL when claim transferred**

Conditions:

```
claim.isFieldChanged( "AssignedUser" )
and
claim.getOriginalValue( "AssignedUser" ) != null

// If a claim has been transferred then reset the ACL
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
// If a claim has been transferred then reset the ACL
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/03/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
                        +actions.getRule().displayName + " Rule at: " +
currentDate + " - " + Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {
          claim.rebuildClaimACL()

 }
catch (e) {

util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
          }
```

3. ABC-CLMPREUP_003: Rebuild ACL on change of Sec Zone

Conditions:

Claim.AssignedGroup.SecurityZone
!=Claim.PreviousGroup.SecurityZone
// If security zone is changed then rebuild the ACL

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
// If security zone is changed then rebuild the ACL
// Modified Date    Modified By     Description
// -------------    ------------    ------------
// 04/03/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING


try {
    Claim.rebuildClaimACL()
    Libraries.Logger.logInfo( "Flushing the ACL" )
}
catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

}
```

4. ABC-CLMPREUP_004 : Rebuild ACL on claim security level change

Conditions:

```
claim.State != "draft"
and
claim.isFieldChanged("State")
```

//rebuild the ACL when confidentiality level changes on the claim

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//rebuild the ACL when confidentiality level changes on the claim
// Modified Date    Modified By     Description
// -------------          -----------          -----------
// 04/02/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING



try {
    if (claim.isFieldChanged( "PermissionRequired" ) )
     {
      claim.rebuildClaimACL()
      }
}
catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

}
```

5. ABC-CLMPREUP_005: Partner Agent Security

Conditions:

```
//This rule assigns a Dummy user from a specific Partner Agent group  to
claims
//so users from that group can access this claim
claim.Policy.ex_PartnerAgent!= null
and claim.ValidationLevel=="newloss"
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//This rule assigns a Dummy user from a specific Partner Agent group to
claims
//so users from that group can access this claim
// Modified Date    Modified By     Description
// -------------          -----------          -----------
// 04/04/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {
   var pa = claim.Policy.ex_PartnerAgent
   var query2 = find ( usr in user.Contact where usr.lastName contains
pa);
   var z = query2.getFirstResult()
   var usrid=z.PublicID
   var query1 = find ( grp in group.Users.User.Contact where
grp.lastName contains pa);
   var y = query1.getFirstResult()
   var grpid=y.PublicID

   if (usrid !=null){
     Claim.assignUserRole( User(usrid ), Group(grpid),
"ABC_Sec_Role_View")
      }

}

catch (e) {
  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
}
```

6. **ABC-CLMPREUP_006: PEO Client Security**

Conditions:

```
//This rule assigns a Dummy user from a specific PEO Client group  to
claims
//so users from that group can access this claim
claim.Policy.Ex_PlanCode!= null
and claim.Policy.ex_PEOClient!= null
and claim.ValidationLevel=="newloss"
and claim.Policy.ex_PEOClient!= claim.Policy.insured
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//This rule assigns a Dummy user from a specific PEO Client group  to
claims
//so users from that group can access this claim.
// Modified Date    Modified By     Description
// -------------    ------------     ------------
// 04/04/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {

   var PEOClient =  Claim.Policy.ex_PEOClient
//var plancode = "PEO"+ claim.Policy.Ex_PlanCode
   var query2 = find ( usr in user.Contact where usr.lastName contains
PEOClient);
   var z = query2.getFirstResult()
   var usrid=z.PublicID
   var query1 = find ( grp in group.Users.User.Contact where
grp.lastName contains PEOClient);
   var y = query1.getFirstResult()
   var grpid=y.PublicID

   if (usrid !=null)
         {    Claim.assignUserRole( User(usrid ), Group(grpid),
"ABC_Sec_Role_View" )    }

    }

catch (e) {
```

```
util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

    }
```

7. ABC-CLMPREUP_007: Sensitive claim add ABC_Sensitive_Role.

Conditions:

```
//This rule assigns a Dummy Security user from a ABC Security
Sensitivity Group to claims //with sensitive claim level.
//Users associated with this group can access sensitive claims.
// At present it is ABC Employees from WC group.

claim.State != "draft"
//and claim.isFieldChanged( "PermissionRequired" )
and Claim.PermissionRequired == "sensitiveclaim"
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//This rule assigns a Dummy Security user from a ABC Security
Sensitivity Group to claims //with sensitive claim level.
//Users associated with this group can access sensitive claims.
// At present it is ABC Employees from WC group.
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/17/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {

     Claim.assignUserRole( User( "cc:1021" /* SecurityUser
SensitiveClaim */ ), Group( "cc:425" /* ABC Security Sesitivity Group */ ),
"ABC_Sens_Role_Ed" )


}

catch (e) {
```

```
    util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
  }
```

8. ABC-CLMPREUP_008: Rebuild ACL on Policy refresh

Conditions:

// Rule to rebuild ACL when the Policy Refresh functionality is invoked

claim.PolicyChanged

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
// Rule to rebuild ACL when the Policy Refresh functionality is invoked
// Modified Date    Modified By     Description
// -------------    ------------    ------------
// 04/19/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING
try {
   claim.rebuildClaimACL()

   }
catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
    }
```

9. **ABC-CLMPREUP_009:Co-broker Security**

Conditions:

```
//This rule assigns a Dummy security user from a specific Co-Broker
group to claims
//so users from that group can access this claim

claim.Policy.Ex_CoBroker!= null
and claim.ValidationLevel=="newloss"
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//This rule assigns a Dummy security user from a specific Co-Broker
group to claims
//so users from that group can access this claim
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/23/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {
   var CoBroker = claim.Policy.Ex_CoBroker
   var query2 = find ( usr in user.Contact where usr.lastName contains
CoBroker);
   var z = query2.getFirstResult()
   var usrid=z.PublicID
   var query1 = find ( grp in group.Users.User.Contact where
grp.lastName contains CoBroker);
   var y = query1.getFirstResult()
   var grpid=y.PublicID

   if (usrid !=null){
     Claim.assignUserRole( User(usrid ), Group(grpid),
"ABC_Sec_Role_View")
      }

}

catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

}
```

10. **ABC-CLMPREUP_010:Security for Subrogation activated claims**

Conditions:

```
//If the Subrogation Status on Loss Details is selected as Open or
Review
//Then that claim is made available to the users of Subrogation group.
//Any user can be member of Subro group

claim.SubrogationStatus=="open"
OR
claim.SubrogationStatus=="review"
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//If the Subrogation Status on Loss Details is selected as Open or
Review
//Then that claim is made available to the users of Subrogation group.
//Any user can be member of Subro group
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/30/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {


    Claim.assignUserRole( User( "cc:202" /* SecurityUser SubroUser */ ),
Group( "cc:121" /* Subro */ ),  "ABC_Sec_Role_Edit")


}

catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

}
```

11. ABC-CLMPREUP_011:PEO Client Location

Conditions:

```
//This rule assigns a Dummy user from a specific PEO Client location
group to claims
//so users from that group can access this claim
claim.Policy.Ex_PlanCode!= null
and claim.Policy.ex_PEOClient!= null
and claim.ValidationLevel=="newloss"
and claim.Policy.ex_PEOClient!= claim.Policy.insured
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//This rule assigns a Dummy security user from a specific PEO Client
location group to
//claims
//  so users from that group can access this claim.
//  Remove access to Old Client Location group user.
//  Provide access to current Client Location group user.
//
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 05/02/2007       Anil Deshpande   Initial Release
//*****************************************************************************/
//START LOGGING

var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING


try {
if (claim.isFieldChanged("LocationCode"))
// to remove access to old Client-location Group user
{
   var oldlocationcd = claim.getOriginalValue( "LocationCode")
   var  query1 = find ( lcn in claim.LocationCode where
lcn.PublicID=="cc:"+ oldlocationcd);
   var u=query1.getFirstResult()
  var oldlocation =u.LocationCode.Location
   var query2 = find ( usr in user.Contact where usr.LastName contains
oldlocation);
   var x = query2.getFirstResult()
   var ousrid=x.PublicID
    if (ousrid !=null)
    { claim.unassignUserRole( User(ousrid ), "ABC_Sec_Role_View" )
      }
```

```
        }


            var PEOClocation = claim.LocationCode.Location
            var query3 = find ( usr in user.Contact where usr.LastName contains
PEOClocation);
            var z = query3.getFirstResult()
            var usrid=z.PublicID
            var query4 = find ( grp in group.Users.User.Contact where
grp.lastName contains PEOClocation);
            var y = query4.getFirstResult()
            var grpid=y.PublicID

            if (usrid !=null)
            // to provide access to current Client Location group user
              {
              Claim.assignUserRole( User(usrid ), Group(grpid),
"ABC_Sec_Role_View" )
              }

        }



      catch (e) {

        util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

            }
```

12. ABC-CLPREUP_012 : Employer Liability Exposer Security

Conditions:

```
//checking for any EmployerLiability exposures on the claim
exists (exp in claim.Exposures where
        exp.ExposureType== "EmployerLiability")
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
// this rule assigns a Dummy Security user from Employer Liability
Adjusters group
// to all the claims that have Exposure with type employer Liability.
// user who are part of this group will now get access to the claim.
//
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 05/04/2007       Anil Deshpande   Initial Release
//*****************************************************************************/
//START LOGGING

var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {
   for(exp in claim.Exposures)
        {
           if (exp.ExposureType== "EmployerLiability")

   {
        claim.assignUserRole( User( "ABC_data:118" /* SecurityUser Emp
Liability */ ), Group( "ABC_data:144" /* Employer Liability Adjusters */ ),
"ABC_EL_Role_Edit" )
                    }

          }

    }

catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

       }
```

13. **ABC-CLMPREUP_013 : Subro, Change of access on Subro status change.**

Conditions:

```
//If the Subrogation Status on Loss Details is changed to closed or Not
selected
//Then claim is made not available to the users of Subrogation group.

claim.isFieldChanged("SubrogationStatus")
and (claim.SubrogationStatus=="closed"
OR
claim.SubrogationStatus== null)
```

Actions:

```
/*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//If the Subrogation Status on Loss Details is changed to closed or Not
selected
//Then claim is made not available to the users of Subrogation group.
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 05/08/2007       Anil Deshpande   Initial Release
*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {


   //  Claim.assignUserRole( User( "cc:202" /* SecurityUser SubroUser */ ),
Group( "cc:121" /* Subro */ ),  "ABC_Sec_Role_Edit")

claim.unassignUserRole( User( "ABC_data:90" /* SecurityUser SubroUser */
),  "ABC_Sec_Role_Edit")
}

catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);

}
```

- **Claim Exception Rules**
  All the following rules are created part of the Rule set "**ABC-Claim Level Security**" this rule set has the **Conditions:** claim.LossType=="WC"

14. **ABC-CLMEXPT-001:Claim Exception rule for converted claims**

Conditions:

```
claim.LoadCommandID != null
```

Actions:

```
//*****************************************************************************
// ABC Phase 1
//
// DESCRIPTION:
//This rule triggers claim validation  in the claim validation we trigger
pre_update rules by
//changing field on Claim
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/03/2007       Anil Deshpande   Initial Release
//*****************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING

try {


claim.validate( 0 )



   }
catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
    }
```

- **Claim Validation Rules**

All the following rules are created part of the Rule set “**ABC-Claim Level Security**” this rule set has the **Conditions:** claim.LossType=="WC"

15. ABC-CLMVAL-0001:Claim Validation rule for converted claims

Conditions:

```
//this is the condition for converted claims
claim.LoadCommandID != null
```

Actions:

```
//*************************************************************************************
// ABC Phase 1
//
// DESCRIPTION: This rules is wirten to trigger Pre_update Rules for
Converted Claims.
// I am updating  claim.FireDeptInfo=claim.FireDeptInfo+" .".This Rule
should be run for
// once  for converted claim.After that it should be disabled.
//
// Modified Date    Modified By     Description
// -------------    -----------     -----------
// 04/20/2007       Anil Deshpande   Initial Release
//*************************************************************************************/
//START LOGGING
var currentDate = Libraries.Date.currentDate()
Libraries.Logger.logInfo("Entered into the "
 +actions.getRule().displayName + " Rule at: " + currentDate + " - " +
Libraries.Date.getMinute(CurrentDate) + ":" +
Libraries.Date.getMinute(CurrentDate))
//END LOGGING


try {

claim.createCustomHistoryEvent( "a_n_f_r", "This claim has been run
through the ABC Security Process" );
claim.FireDeptInfo=claim.FireDeptInfo+" ."// this is added to trigger
Pre_update rules
   }
catch (e) {

  util.CustomLogger.logRule(Claim, Actions, "Caught exception:" + e);
    }
```

10 Admin setup and Maintenance

- **Admin setup**

- **Create the organization structure** as per the “ABC Organization Structure WC - FINAL.doc” by importing ABC_AdminData.xml , this includes all the Claim level Security Zones, Groups, Group Types, “Dummy Security users” and association of security users to specific groups in PEO ,PEO Client, Partner Agent, ABC Employee groups and ABC Security Sensitivity Group. All these users first name starts with SecurityUser.
- **Configuring Security Zones**
  We use the security-zones.xml file to manage security zones. This is discussed in Configuration section.
  Check that all the zones show up in the security zone dropdown.
- **Associating Security Zones to groups**
  To provide the necessary access to users within and across security zones, we need to associate Security groups to the users which need access, For example ABC employees are associated with all the 4 security groups. For the first time it is achieved by using Admin Data loader and importing it. For the subsequent user creation please refer the maintenance section of this document.

    - From Administration, assign the group to corresponding security zones.
    - TPA groups are to be associated with their specific security zones.

- **Add all TPA groups to the ABC group users**
  The users in the ABC Employee group should be able to access all the Claims view and edit .We can achieve this by associating all the TPA groups to ABC employee users.

- **Maintenance of security configuration**

  This section addresses business process required to maintain security intact, if there are any changes in the personnel, security model or admin data at ABC.

Creating new groups:

1. **Creating New PEO group**
   When you create new PEO you need to create a dummy security user in that group. The User first name should be "SecurityUserPEO<*PlanID*>" .Last name can be name of the PEO group. Refer the existing PEO group.
2. **Creating New PEO Client Group**
   When you create new PEO Client Group, you need to create a dummy security user with **First name** "SecurityUser" and **Last Name <PEOClient name as it appears in Policy>**
3. **Creating New PEO Client Location Group**
   When you create new PEO Client Location Group, you need to create a dummy security user with First name "SecurityUser" and Last Name <PEOClient Location as it appears in CC_Property.location. This is populated by Policy adopter>.Please check the organization structure in admin for example.
4. **Creating New Partner Agent Group**
   When you create new Partner agent Group, need to create a dummy security user with **First name** "SecurityUser" and **Last Name < Partner Agent name as it appears in Policy>**
5. **Creating New CO-Broker group**
   When you create new Co-Broker Group, you need to create a dummy security user with **First name** "SecurityUser" and **Last Name < Co-Broker name as it appears in The Policy>**
6. **Creating new TPA group:**
   a. When you create new TPA group you need to create new security zone with name specific to that TPA group. Then associate security zone to that group.
   b. Then add this newly created TPA group to users in "ABC employee" group, so that "ABC Employee" users will have access to the claims assigned to this TPA group.
7. **Creating New ABC User**
   When you add new ABC user in the Claims management group, you need to associate all the Security groups to that user .This provides the ABC user to have access to all the claims assigned to TPA groups and sensitive claims.
   Please refer to the "ABC Organization Structure WC - FINAL.doc" for organization structure of ABC Claim +
   Creating New Sensitive Claim users
   To make any user to access the sensitive claims all we need to is; add the "ABC Security Sensitivity Group" to that user.
8. **Creating New Security User**
   All the SecurityUsers created through Admin Data Loader have the password "20ABC07", when you add new SecuriyUser in any group as required by security(mentioned above), keep the password to "20ABC07" to be consistent with other SecurityUSers .

Rule Maintenance:

I have tried to make all the rules requiring no or less maintenance. I have avoided hard coding .If there is change in the business requirement or logic then modify specific rule. All the rules are commented with details on functionality it performs. All the rules are maintained in rule sets “ABC-Claim Level Security” under Claim Pr Update, Claim Exception and Claim Validation rules. All the security related rules are made part of “ABC Claim Level Security Rules” Resource Context

Maintenance of Config files.

To achieve required security following config files are modified. The details are explained in the configuration section.

1. Security-config.xml
2. ClaimSecurityType.xml
3. GroupType.xml
4. UserRole.xml
5. security-zones.xml
6. config.xml
7. ResourceContext

11 Use case

Functionality Tested for

- Claim Level Security for un-secured claims
- Across TPA,
- Across PEO,
- Across PEO Clients and Parent PEO
- Across PA s
- All Access to ABC Employees
- Claim Level Security for secured claims
- Coverage, Employee clam and SIU
- TPAs and ABC employee only
- Claim Level Security for Sensitive claims
- Handling Adjuster and ABC employee only
- Claim Level Security Across Loss types
- Claim Level Security on Reassignment
- Claim Level Security on change of TPA Group/ Sec Zone
- Claim Level Security on change of sensitivity level
- Claim Level Security to view and edit to specific groups
- Conversion claims
- Co Brokers
- subrogation
- PEO Client Location
- Employer Liability Adjusters