

Provisioning Users, Permissions, and Roles

Setting up authorization and authentication

Access to data and services in the Marketcetera Automated Trading Platform is managed by users, permissions, and roles.

Each role can have one or more permissions assigned to it. Each user can be assigned one or more roles. The aggregate set of permissions for a user comes from the set of roles to which the user belongs.

In addition, there are supervisor permissions. Supervisor permissions assign a given permission over a user to another user. For example, the *View Execution Reports* permission can be assigned to the user *supervisor1* over the user *trader1*. This would mean that *supervisor1* can see the execution reports that *trader1* can see.

Default Users

User	Password	Description
trader	trader	Trader User
traderAdmin	traderAdmin	Trader Admin User
admin	admin	Admin User

Default Permissions

Permission	Description
AddSessionAction	Access to Add Session action
DeleteSessionAction	Access to Delete Session action
DisableSessionAction	Access to disable session action
EditSessionAction	Access to edit session action
EnableSessionAction	Access to enable session action
UpdateSequenceAction	Access to update sequence numbers action
StartSessionAction	Access to start session action
StopSessionAction	Access to stop session action
ViewSessionAction	Access to view session action
ReadInstanceDataAction	Access to read instance data action
ReadFixSessionAttributeDescriptorsAction	Access to read FIX session attribute descriptors action
CreateUserAction	Access to create user action
ReadUserAction	Access to read user action
UpdateUserAction	Access to update user action
DeleteUserAction	Access to delete user action
ChangeUserPasswordAction	Access to change user password action
ReadUserPermissionsAction	Access to read user permissions action
CreatePermissionAction	Access to create permission action
ReadPermissionAction	Access to read permission action
UpdatePermissionAction	Access to update permission action
DeletePermissionAction	Access to delete permission action
CreateRoleAction	Access to create role action
ReadRoleAction	Access to read role action

UpdateRoleAction	Access to update role action
DeleteRoleAction	Access to delete role action
ViewBrokerStatusAction	Access to view broker status action
ViewOpenOrdersAction	Access to view open orders action
ViewReportAction	Access to view reports action
ViewPositionAction	Access to view positions action
SendOrderAction	Access to send new orders action
ViewUserDataAction	Access to view user data action
WriteUserDataAction	Access to write user data action
AddReportAction	Access to manually add new reports action
DeleteReportAction	Access to manually delete reports action
ReadUserAttributeAction	Access to read a user attribute action
WriteUserAttributeAction	Access to write a user attribute action

Default Roles

Role	Description	Users	Permissions
Admin	Admin role	admin	AddSessionAction DeleteSessionAction DisableSessionAction EditSessionAction EnableSessionAction UpdateSequenceAction StartSessionAction StopSessionAction ViewSessionAction ReadInstanceDataAction CreateUserAction ReadUserAction UpdateUserAction DeleteUserAction ChangeUserPasswordAction ReadUserPermissionsAction CreatePermissionAction ReadPermissionAction UpdatePermissionAction DeletePermissionAction CreateRoleAction ReadRoleAction UpdateRoleAction DeleteRoleAction ViewBrokerStatusAction ViewUserDataAction WriteUserDataAction ReadFixSessionAttributeDescriptorsAction

			ReadUserAttributeAction
			WriteUserAttributeAction
TraderAdmin	Trader Admin role	traderAdmin	ViewBrokerStatusAction
			ViewOpenOrdersAction
			ViewReportAction
			ViewPositionAction
			SendOrderAction
			ViewUserDataAction
			WriteUserDataAction
			AddReportAction
			DeleteReportAction
			ReadUserAttributeAction
			WriteUserAttributeAction
Trader	Trader role	trader	AddReportAction
			ViewBrokerStatusAction
			ViewOpenOrdersAction
			ViewReportAction
			ViewPositionAction
			SendOrderAction
			ViewUserDataAction
			WriteUserDataAction
			ReadUserAttributeAction
			WriteUserAttributeAction

The default roles tie the users and permissions together. In the default setup, *trader* is assigned the permissions associated with the *trader* role and *traderAdmin* is assigned the permissions associated with the *traderAdmin* role. In this case, the naming is perhaps confusing as the roles don't have to have the same names as the users. Another user, *admin*, has strictly administrative permissions and no trading permissions. Note that some of these permissions imply a subject user as well. For instance, *ViewReportAction* doesn't allow viewing all reports, just reports owned by the given user. So, *traderAdmin* can see her own reports but cannot see the reports for *trader*. In order to make the *TraderAdmin* user more effective as an administrator, she needs to have some permissions over *trader*. These are called Supervisor Permissions.

Default Supervisor Permissions

Supervisor Permission Name	Description	Supervisor User	Subject Users	Permissions
TraderSupervisor	Trader supervisor role	traderAdmin	trader	ViewBrokerStatusAction
				ViewOpenOrdersAction
				ViewReportAction
				ViewPositionAction
				ViewUserDataAction

This supervisor permission grants the given permissions to *traderAdmin* over *trader*. This means that *traderAdmin* has the *ViewReportAction* and can see reports for *traderAdmin* and *trader*.

Customization

New users, permissions, roles, and supervisor permissions can be added to augment the defaults. Adding new permissions would be useful only for custom modules as the existing permissions are explicitly tied to existing system behaviors. Existing permissions can be reused in custom modules, for example, you can use *SendOrderAction* in a custom module if you want.

The easiest way to add new users, permissions, roles, and supervisor permissions is to modify *provisioning.xml* and restart DARE. This file is read on startup and any changes in the file will be executed. In reality, the changes can be made to any of the XML config files, but, semantically, it probably makes the most sense to make the changes in *provisioning.xml*.

Add a New Permission and Role

```

<!-- custom permission -->
<bean id="newPermission" class="org.marketcetera.admin.PermissionDescriptor">
    <property name="name" value="CustomAction"/>
    <property name="description" value="Access to some custom permission"/>
</bean>
<!-- assign permission to a new role and add users to that role -->
<bean id="newRole" class="org.marketcetera.admin.RoleDescriptor">
    <property name="name" value="CustomRole"/>
    <property name="description" value="Custom Role"/>
    <property name="permissionNames">
        <set>
            <value>CustomAction</value>
        </set>
    </property>
    <property name="usernames">
        <set>
            <value>trader</value>
            <value>traderAdmin</value>
        </set>
    </property>
</bean>
<!-- action to activate the new permission and role -->
<bean class="org.marketcetera.admin.AuthorizationInitializer">
    <property name="permissions">
        <set>
            <ref bean="newPermission"/>
        </set>
    </property>
    <property name="roles">
        <set>
            <ref bean="newRole"/>
        </set>
    </property>
</bean>

```

Note that each permission and role must have a unique name, including among all default permissions and roles.

Add an Existing User to an Existing Role

```

<bean class="org.marketcetera.admin.AddUserToRoleAction">
    <property name="username" value="SomeExistingUserName"/>
    <property name="roleName" value="SomeExistingRoleName"/>
</bean>

```

Add an Existing User to an Existing Supervisor Permission

```

<bean class="org.marketcetera.admin.AddUserToSupervisorRoleAction">
    <property name="subjectUsername" value="SomeExistingUserName"/>
    <property name="supervisorPermissionName" value="SomeExistingSupervisorPermissionName"/>
</bean>

```

Customizations Without Restarting

It is possible to make changes to users, permissions, and roles without restarting the DARE server.

Locate the directory that your DARE instance is monitoring for action files.

```

$ grep ProvisioningAgent dare/logs/dare-instance1.log
1 Aug 2019 09:35:19,648 INFO [main] marketcetera.core.ProvisioningAgent (:) - Watching /opt/Marketcetera-3.2.0
/dare/instances/instance1/instances/provisioning1 for provisioning files
$ ls -d /opt/Marketcetera-3.2.0/dare/instances/instance1/instances/provisioning1
ls: cannot access '/opt/Marketcetera-3.2.0/dare/instances/instance1/instances/provisioning1': No such file or
directory
$ mkdir -p /opt/Marketcetera-3.2.0/dare/instances/instance1/instances/provisioning1
$ ls -d /opt/Marketcetera-3.2.0/dare/instances/instance1/instances/provisioning1
/opt/Marketcetera-3.2.0/dare/instances/instance1/instances/provisioning1

```

Prepare a file that contains the instructions as indicated above.

```

<?xml version="1.0" encoding="UTF-8" ?>
<beans xmlns="http://www.springframework.org/schema/beans"
       xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
       xmlns:context="http://www.springframework.org/schema/context"
       xsi:schemaLocation="http://www.springframework.org/schema/beans http://www.springframework.org/schema
/beans/spring-beans.xsd
                                         http://www.springframework.org/schema/context http://www.springframework.org/schema
/context/spring-context.xsd">
    <context:component-scan base-package="org.marketcetera.admin"/>
    <!-- custom permission -->
    <bean id="newPermission" class="org.marketcetera.admin.PermissionDescriptor">
        <property name="name" value="CustomAction"/>
        <property name="description" value="Access to some custom permission"/>
    </bean>
    <!-- assign permission to a new role and add users to that role -->
    <bean id="newRole" class="org.marketcetera.admin.RoleDescriptor">
        <property name="name" value="CustomRole"/>
        <property name="description" value="Custom Role"/>
        <property name="permissionNames">
            <set>
                <value>CustomAction</value>
            </set>
        </property>
        <property name="usernames">
            <set>
                <value>trader</value>
                <value>traderAdmin</value>
            </set>
        </property>
    </bean>
    <!-- action to activate the new permission and role -->
    <bean class="org.marketcetera.admin.AuthorizationInitializer">
        <property name="permissions">
            <set>
                <ref bean="newPermission"/>
            </set>
        </property>
        <property name="roles">
            <set>
                <ref bean="newRole"/>
            </set>
        </property>
    </bean>
</beans>

```

Save this file (the name isn't important).

Copy the file to the monitored directory

```
$ cp ~/temp.xml /opt/Marketcetera-3.2.0/dare/instances/instance1/instances/provisioning1/
$ tail -100 /opt/Marketcetera-3.2.0/dare/logs/dare-instance1.log
...
05 Aug 2019 09:45:34,405  INFO [pool-4-thread-1] marketcetera.core.ProvisioningAgent (:) - Reading provisioning
from temp.xml
05 Aug 2019 09:45:34,505  INFO [pool-4-thread-1] marketcetera.admin.AuthorizationInitializer (:) - Adding
permission PermissionDescriptor [getName()=CustomAction, getDescription()=Access to some custom permission]
05 Aug 2019 09:45:34,514  INFO [pool-4-thread-1] marketcetera.admin.AuthorizationInitializer (:) - Adding role
RoleDescriptor [permissionNames=[CustomAction], usernames=[trader, traderAdmin], getDescription()=Custom Role,
getName()=CustomRole]
```

Note that this technique isn't limited to authorization or authentication files. Any command or action that can be formulated as XML-based config can be executed in this fashion. Care must be taken when using this technique.

- [Monitoring Broker Connections](#)
- [Provisioning Users, Permissions, and Roles](#)