

Custom profile implementation

Instead of using low level [com.nomagic.uml2.ext.jmi.helpers.StereotypesHelper](#) API to access information about profile elements, it is possible to generate a profile implementation class out of the UML Profile model. This implementation class wraps all mapping, naming details with a nice Java API. MagicDraw comes with already generated profiles implementation for standard profiles. See [Standard profiles implementation](#) page for details.

Related pages

- [Standard profiles implementation](#)

To create your own custom profile implementation you need to follow these steps

1. Install Development Tools plugin from Help->Resource Manager
2. Open UML Profile model with MagicDraw.
3. Generate profile implementation code using Tools->Development Tools->Generate Profile Class Implementation

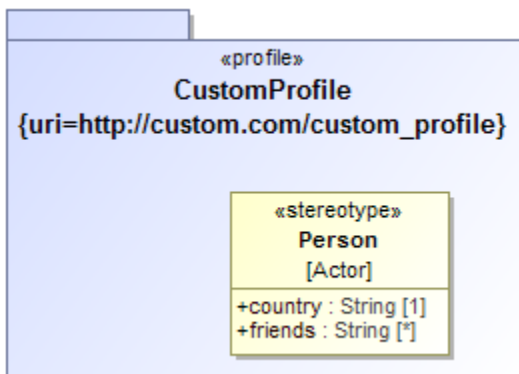
Profile implementation class has these features

- There is a constant defined for every DataType, Enumeration or EnumerationLiteral name
- There is a getter for every DataType element defined in the profile
- There is a getter for every Stereotype element defined in the profile
- There is a static check method to check if Element has specific stereotype applied
- Java enumeration is generated for every Enumeration element defined in the profile
- Inner java class is generate for every Stereotype defined in the profile. This class has following features
 - There is a constant defined for Stereotype name
 - There are constants defined for all stereotype tags names
 - There is a setter/getter method available for every stereotype tag. Signature of these methods corresponds to tag multiplicity type and etc. No more need to work with strings and casting values to primitives as in *StereotypesHelper* case.



It is highly recommended to assign URI to Profile element in the UML model before generation. URI is an unique Profile identifier and helps to avoid conflicts if model has several profiles with same name applied.

Custom sample profile



Usages example

```

Element element = ...;
//check if element has Person stereotype applied
CustomProfile.isPerson(element)
//get Person.country tag value
String country = CustomProfile.Person.getCountry(element);
//set Person.country tag value
CustomProfile.Person.setCountry(element, "US");

```

Generated implementation class for this custom profile

```

import com.nomagic.magicdraw.core.Project;
import com.nomagic.magicdraw.uml.BaseElement;
import com.nomagic.uml2.ext.jmi.helpers.StereotypeByProfileCache;
import com.nomagic.uml2.ext.jmi.helpers.StereotypesHelper;
import com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Element;
import com.nomagic.uml2.ext.magicdraw.mdprofiles.Stereotype;
import javax.annotation.CheckForNull;
import javax.annotation.Nonnull;
import java.util.Collection;
import java.util.HashSet;

public class CustomProfile extends StereotypeByProfileCache
{
    public static final String PROFILE_URI = "http://custom.com
/custom_profile";
    public static final String PROFILE_NAME = "Custom Profile";

    public static CustomProfile getInstance(@Nonnull BaseElement element)
    {
        return getInstance(Project.getProject(element));
    }

    public CustomProfile(@Nonnull Project project)
    {
        super(project, PROFILE_NAME, PROFILE_URI);
    }

    public static CustomProfile getInstance(@Nonnull Project project)
    {
        CustomProfile instance = _getInstance(CustomProfile.class, project);
        if (instance == null)
        {
            instance = new CustomProfile(project);
        }
        return instance;
    }

    public static class Person extends AbstractStereotypeWrapper
    {
        //stereotype Person and its tags
        @SuppressWarnings("UnusedDeclaration")
        public static final String STEREOTYPE_NAME = "Person";
        @SuppressWarnings("UnusedDeclaration")
        public static final String COUNTRY = "country";
        @SuppressWarnings("UnusedDeclaration")
        public static final String FRIENDS = "friends";

        @SuppressWarnings("UnusedDeclaration")
        public static void setCountry(Element element, String value)
        {
            StereotypesHelper.setStereotypePropertyValue(element, getInstance
(element).getPerson(), COUNTRY, value);
        }

        @SuppressWarnings("UnusedDeclaration, unchecked")
        @CheckForNull
        public static String getCountry(Element element)

```

```

        {
            return toString(StereotypesHelper.getStereotypePropertyFirst
(element, getInstance(element).getPerson(), COUNTRY));
        }

        @SuppressWarnings("UnusedDeclaration")
        public static void setFriends(Element element, java.util.
List<String> value)
        {
            StereotypesHelper.setStereotypePropertyValue(element, getInstance
(element).getPerson(), FRIENDS, value);
        }

        @SuppressWarnings("UnusedDeclaration")
        public static void addFriends(Element element, String value)
        {
            StereotypesHelper.setStereotypePropertyValue(element, getInstance
(element).getPerson(), FRIENDS, value, true);
        }

        @SuppressWarnings("UnusedDeclaration")
        public static void removeFriends(Element element, String value)
        {
            java.util.List<String> values = new java.util.ArrayList<>
(getFriends(element));
            values.remove(value);
            setFriends(element, values);
        }

        @SuppressWarnings("UnusedDeclaration, unchecked")
        @Nonnull
        public static java.util.List<String> getFriends(Element element)
        {
            return (java.util.List<String>) (StereotypesHelper.
getStereotypePropertyValue(element, getInstance(element).getPerson(),
FRIENDS));
        }
    }

    @SuppressWarnings({"UnusedDeclaration", "ConstantConditions"})
    public Stereotype getPerson()
    {
        return getStereotype(Person.STEREOTYPE_NAME);
    }

    @SuppressWarnings("UnusedDeclaration")
    public static boolean isPerson(@CheckForNull Element element)
    {
        if (element instanceof com.nomagic.uml2.ext.magicdraw.mdusecases.
Actor)
        {
            CustomProfile instance = getInstance(element);
            return instance.isTypeOf(element, instance.getPerson());
        }
        return false;
    }

    @Override
    protected Collection<Stereotype> generatedGetAllStereotypes()
    {
        if (getProfile() != null)
        {
            final Collection<Stereotype> stereotypes = new HashSet<>();
            stereotypes.add(getPerson());
            return stereotypes;
        }
        return super.generatedGetAllStereotypes();
    }
}

```

