

# Custom profile implementation

## Related pages

- [Standard profiles 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.

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 (wrappers) ...

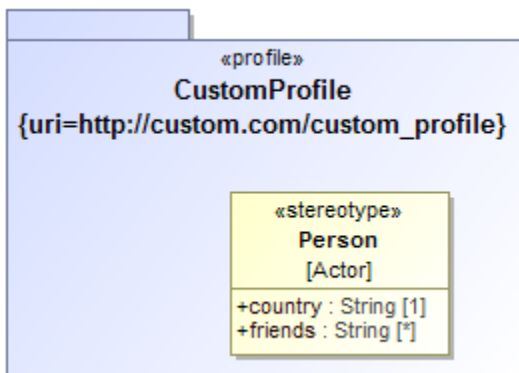
Profile implementation class has these features

- There is a constant defined for every DataType, Enumeration name
- There is a getter for every DataType element defined in the profile
- Java enumeration is generated for every Enumeration element defined in the profile. This class has following features
  - There are methods to get enumeration literals by generated enumeration and vice versa
- Inner java class is generated 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.
  - There is a check method to check if Element has specific stereotype applied
  - There is a getter for Stereotype element itself



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 = ...;
CustomProfile customProfile = CustomProfile.getInstance(element);
//check if element has Person stereotype applied
customProfile.person().is(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.uml.BaseElement;
import com.nomagic.magicdraw.uml2.Profiles;
import com.nomagic.profiles.ProfileCache;
import com.nomagic.profiles.ProfileImplementation;
import com.nomagic.profiles.ProfilesBridge;
import com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Element;
import com.nomagic.uml2.ext.magicdraw.classes.mdkernel.Property;
import com.nomagic.uml2.ext.magicdraw.mdprofiles.Stereotype;

import javax.annotation.CheckForNull;
import java.util.ArrayList;
import java.util.Collection;
import java.util.Collections;
import java.util.HashSet;

@SuppressWarnings("WeakerAccess, unused")
public class CustomProfile extends ProfileImplementation
{
    public static final String PROFILE_URI = "http://custom.com
/custom_profile";

    public static final String PROFILE_NAME = "CustomProfile";

    private final PersonStereotype personStereotype;

    public static CustomProfile getInstance(BaseElement baseElement)
    {
        CustomProfile profile = ProfilesBridge.getProfile(CustomProfile.
class, baseElement);
        if (profile == null)
        {
            return ProfilesBridge.createProfile(CustomProfile.class,
baseElement, CustomProfile::new, PROFILE_NAME, PROFILE_URI);
        }
        return profile;
    }

    public CustomProfile(ProfileCache cache)
    {
        super(cache);
        personStereotype = new PersonStereotype(this);
    }

    public PersonStereotype person()
    {
        return personStereotype;
    }

    public static class PersonStereotype extends StereotypeWrapper
    {
        //stereotype Person and its tags
        public static final String STEREOYPE_NAME = "Person";
        public static final String COUNTRY = "country";
        public static final String FRIENDS = "friends";
    }
}

```

```

private final CustomProfile _p;
@CheckForNull
private Property country;
@CheckForNull
private Property friends;

protected PersonStereotype(CustomProfile profile)
{
    super(profile);
    _p = profile;
}

@Override
@SuppressWarnings("ConstantConditions")
public Stereotype getStereotype()
{
    return getElementByName(STEREOTYPE_NAME);
}

@CheckForNull
public Property getCountryProperty()
{
    if (country == null)
    {
        country = getTagByName(getStereotype(), COUNTRY);
    }
    return country;
}

@CheckForNull
public Property getFriendsProperty()
{
    if (friends == null)
    {
        friends = getTagByName(getStereotype(), FRIENDS);
    }
    return friends;
}

public void setCountry(Element element, @CheckForNull String value)
{
    Profiles.setValue(element, getStereotype(), getCountryProperty(),
value);
}

public void clearCountry(Element element)
{
    Profiles.clearValue(element, getCountryProperty());
}

@CheckForNull
public String getCountry(Element element)
{
    return toString(Profiles.getFirstValue(element,
getCountryProperty()));
}

public void setFriends(Element element, @CheckForNull java.util.
Collection<String> value)
{
    Profiles.setValue(element, getStereotype(), getFriendsProperty(),
value);
}

public void clearFriends(Element element)
{
    Profiles.clearValue(element, getFriendsProperty());
}

public void addFriends(Element element, String value)

```

```

    {
        Profiles.addValue(element, getStereotype(), getFriendsProperty(),
value);
    }

    public void removeFriends(Element element, String value)
    {
        java.util.List<String> values = getFriends(element);
        values.remove(value);
        setFriends(element, values);
    }

    @SuppressWarnings("unchecked")
    public java.util.List<String> getFriends(Element element)
    {
        return (java.util.List<String>) Profiles.getValue(element,
getFriendsProperty());
    }

    @Override
    protected void clear()
    {
        super.clear();
        country = null;
        friends = null;
    }

    @Override
    public boolean is(@CheckForNull Element element)
    {
        return element instanceof com.nomagic.uml2.ext.magicdraw.
mdusecases.Actor &&
        _p.isTypeOf(element, getStereotype());
    }

    public static boolean isInstance(@CheckForNull Element element)
    {
        if (element instanceof com.nomagic.uml2.ext.magicdraw.mdusecases.
Actor)
        {
            CustomProfile instance = getInstance(element);
            return instance.isTypeOf(element, instance.person().
getStereotype());
        }
        return false;
    }
}

@Override
protected Collection<ProfileElementWrapper>
generatedGetAllElementWrappers()
{
    Collection<ProfileElementWrapper> wrappers = new ArrayList<>();
    wrappers.add(personStereotype);
    return wrappers;
}

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

```

