

ABAPSpace

ABAP NAMESPACE REFACTORING

MANUAL

TABLE OF CONTENTS

LIST OF FIGURES.....	II
TABLE DIRECTORY.....	III
LIST OF ABBREVIATIONS.....	IV
1. INTRO.....	1
2. ENHANCED OBJECT POLICY.....	1
2.1 ABAP OBJECT IDENTIFIER.....	1
3. REPLACEMENT.....	2
3.1 WITH OBJECT POLICY.....	2
3.2 WITHOUT OBJECT POLICY.....	2
3.3 SUPPLEMENT.....	2
4. PRESET.....	3
5. USER MANUAL.....	7
5.1 PREREQUISITES.....	7
5.2 RUN ABAPSpace.....	7
5.3 USE ABAPSpace.....	7
LIST OF REFERENCES.....	10

LIST OF FIGURES

Illustration 1: ABAPSpace - start screen.....	8
Illustration 2: ABAPSpace - edit screen.....	9

TABLE DIRECTORY

Table 1: ABAP object identifier.....	1
Table 2: Preset XML Schema.....	3

LIST OF ABBREVIATIONS

1. INTRO

ABAPSpace is a Java tool for namespace refactoring.

2. ENHANCED OBJECT POLICY

The enhanced object policy expects an object name as follows:

<NAMESPACE><OBJECT_IDENTIFIER><OBJECT_NAME>

2.1 ABAP OBJECT IDENTIFIER

Specific checks based on specific ABAP objects can only be made, if an object was recognized as such. The detection is only performed on the basis of source files, without the SAP system. To enable such checks, the object name must indicate what the object type is.

Therefore, the following object identifier are required:

ABAP Object	Object Identifier	Maximum Length	Description
Package	PA	30	
Class	CL	30	Abbreviation is SAP default
Interface	IF	30	Abbreviation is SAP default
Exception Class	CX	30	Abbreviation is SAP default
Program	PR	40	
Include	IN	40	
Function Group	FG	26	
Function Module	FM	30	
DB Table	TA	16	
DB View	VW	16	
Table Type	TT	30	
Structure	ST	30	
Data Element	DA	30	
Domain	DO	30	
Message Class	MC	20	

Table 1: ABAP object identifier

There are already standard identifiers like classes (CL) or interfaces (IF). Further ones are defined on this basis.

3. REPLACEMENT

3.1 WITH OBJECT POLICY

<NAMESPACE_NEW><OBJECT_IDENTIFIER><SUPPLEMENT><OBJECT_NAME>

3.2 WITHOUT OBJECT POLICY

<NAMESPACE_NEW><SUPPLEMENT><OBJECT_NAME>

3.3 SUPPLEMENT

The supplement is a specific prefix to group objects. For example, objects belong to a specific development package or project.

4. PRESET

All settings of ABAPSpace are stored in a preset XML file (file suffix: *.xpx).
The following XML schema defines the preset file:

Table 2: Preset XML Schema

```

001 <?xml version="1.0" encoding="UTF-8"?>
002 <xsd:schema xmlns:xsd="http://www.w3.org/2001/XMLSchema"
003     targetNamespace="http://www.abap.space.com/Preset"
004     xmlns:tns="http://www.abap.space.com/Preset"
005     elementFormDefault="qualified">
006
007     <xsd:element name="objectNameIdent">
008         <xsd:simpleType>
009             <xsd:restriction base="xsd:string">
010                 <xsd:minLength value="1" />
011             </xsd:restriction>
012         </xsd:simpleType>
013     </xsd:element>
014
015     <xsd:element name="preIdent">
016         <xsd:simpleType>
017             <xsd:restriction base="xsd:string">
018                 <xsd:minLength value="0" />
019             </xsd:restriction>
020         </xsd:simpleType>
021     </xsd:element>
022
023     <xsd:element name="postIdent">
024         <xsd:simpleType>
025             <xsd:restriction base="xsd:string">
026                 <xsd:minLength value="0" />
027             </xsd:restriction>
028         </xsd:simpleType>
029     </xsd:element>
030
031     <xsd:complexType name="objectTypeGen" mixed="true">
032         <xsd:sequence>
033             <xsd:element ref="tns:preIdent" minOccurs="0"
034 maxOccurs="1"></xsd:element>
035             <xsd:element name="objectIdent" minOccurs="1" maxOccurs="1">
036                 <xsd:simpleType>
037                     <xsd:restriction base="xsd:normalizedString">
038                         <xsd:minLength value="1" />
039                     </xsd:restriction>
040                 </xsd:simpleType>
041             </xsd:element>
042             <xsd:element ref="tns:objectNameIdent" minOccurs="1"
043 maxOccurs="1"></xsd:element>
044             <xsd:element ref="tns:postIdent" minOccurs="0"
045 maxOccurs="1"></xsd:element>
046             <xsd:element name="nameMaxLength" minOccurs="1"
047 maxOccurs="1">
048                 <xsd:simpleType>
049                     <xsd:restriction base="xsd:int">
050                         <xsd:minInclusive value="0" />
051                     </xsd:restriction>
052                 </xsd:simpleType>
053             </xsd:element>
054         </xsd:sequence>
055     </xsd:complexType>

```


056
057

```

    <xsd:element name="preset">
      <xsd:complexType>
        <xsd:sequence>
          <xsd:element name="description" minOccurs="1" maxOccurs="1">
            <xsd:simpleType>
              <xsd:restriction base="xsd:normalizedString">
                <xsd:minLength value="1" />
                <xsd:maxLength value="200" />
              </xsd:restriction>
            </xsd:simpleType>
          </xsd:element>
          <xsd:element name="refactorSourceDir"
type="xsd:normalizedString"
minOccurs="1" maxOccurs="1"></xsd:element>
          <xsd:element name="refactorTargetDir"
type="xsd:normalizedString"
minOccurs="1" maxOccurs="1"></xsd:element>
          <xsd:element name="fileStructure" minOccurs="1"
maxOccurs="1">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="update"
type="xsd:boolean"
minOccurs="1"
maxOccurs="1"></xsd:element>
                <xsd:element name="namespacePlaceholder"
type="xsd:normalizedString"
minOccurs="1"
maxOccurs="1"></xsd:element>
                <xsd:element name="namespaceReplacement"
type="xsd:normalizedString"
minOccurs="1"
maxOccurs="1"></xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
          <xsd:element name="checkNameMaxLength" type="xsd:boolean"
minOccurs="1" maxOccurs="1"></xsd:element>
          <xsd:element name="namespaceNew" type="xsd:normalizedString"
minOccurs="1" maxOccurs="1"></xsd:element>
          <xsd:element name="supplement" type="xsd:normalizedString"
minOccurs="0" maxOccurs="1"></xsd:element>
          <xsd:element name="namespaceOldList" minOccurs="1"
maxOccurs="1">
            <xsd:complexType>
              <xsd:sequence>
                <xsd:element name="namespaceOld"
minOccurs="1"
maxOccurs="unbounded">
                  <xsd:complexType>
                    <xsd:sequence>
                      <xsd:element
name="objectPolicy" type="xsd:boolean"
minOccurs="0"
maxOccurs="1" />
                      <xsd:element
name="namespaceOld" type="xsd:normalizedString"
minOccurs="1"
maxOccurs="1" />
                    </xsd:sequence>
                  </xsd:complexType>
                </xsd:element>
              </xsd:sequence>
            </xsd:complexType>
          </xsd:element>
        </xsd:sequence>
      </xsd:complexType>
    </xsd:element>
  
```

```

        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="objectPolicyGeneral" minOccurs="0"
    maxOccurs="1">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element ref="tns:preIdent"
                minOccurs="0"
                maxOccurs="1"></xsd:element>
            <xsd:element ref="tns:objectNameIdent"
                minOccurs="1"
                maxOccurs="1"></xsd:element>
            <xsd:element ref="tns:postIdent"
                minOccurs="0"
                maxOccurs="1"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>
<xsd:element name="objectPolicy" minOccurs="0"
    maxOccurs="1">
    <xsd:complexType>
        <xsd:sequence>
            <xsd:element name="objectPackage"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectClass"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectInterface"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectExceptionClass"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectProgram"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectInclude"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectFunctionGroup"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectFunctionModule"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectDatabaseTable"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
            <xsd:element name="objectDatabaseView"
                minOccurs="0"
                maxOccurs="1"
                type="tns:objectTypeGen"></xsd:element>
        </xsd:sequence>
    </xsd:complexType>
</xsd:element>

```

```

minOccurs="0"

type="tns:objectTypeGen"></xsd:element>

minOccurs="0"

type="tns:objectTypeGen"></xsd:element>

minOccurs="0"

type="tns:objectTypeGen"></xsd:element>

minOccurs="0"

type="tns:objectTypeGen"></xsd:element>

minOccurs="0"

type="tns:objectTypeGen"></xsd:element>

</xsd:sequence>
</xsd:complexType>
</xsd:element>
<xsd:element name="keywordsExclude" minOccurs="0"
maxOccurs="1">
<xsd:complexType>
<xsd:sequence>
<xsd:element name="keyword" minOccurs="0"

type="xsd:normalizedString">
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:sequence>
</xsd:complexType>
</xsd:element>
</xsd:schema>

```

5. USER MANUAL

5.1 PREREQUISITES

Runtime environment. You have to install the [Java Runtime Environment \(>= Version 8\)](#).

Setup. All settings of ABAPSpace are stored in a Preset XML file. An example file can be found under:

- abapSPACE/bin/preset

5.2 RUN ABAPSpace

Go to the following ABAPSpace directory

- abapSPACE/bin

and run a start script:

- abapSPACE → Linux
- abapSPACE.bat → Windows

5.3 USE ABAPSpace

Start screen. If ABAPSpace has been started, the following screen appears (Illustration 1):

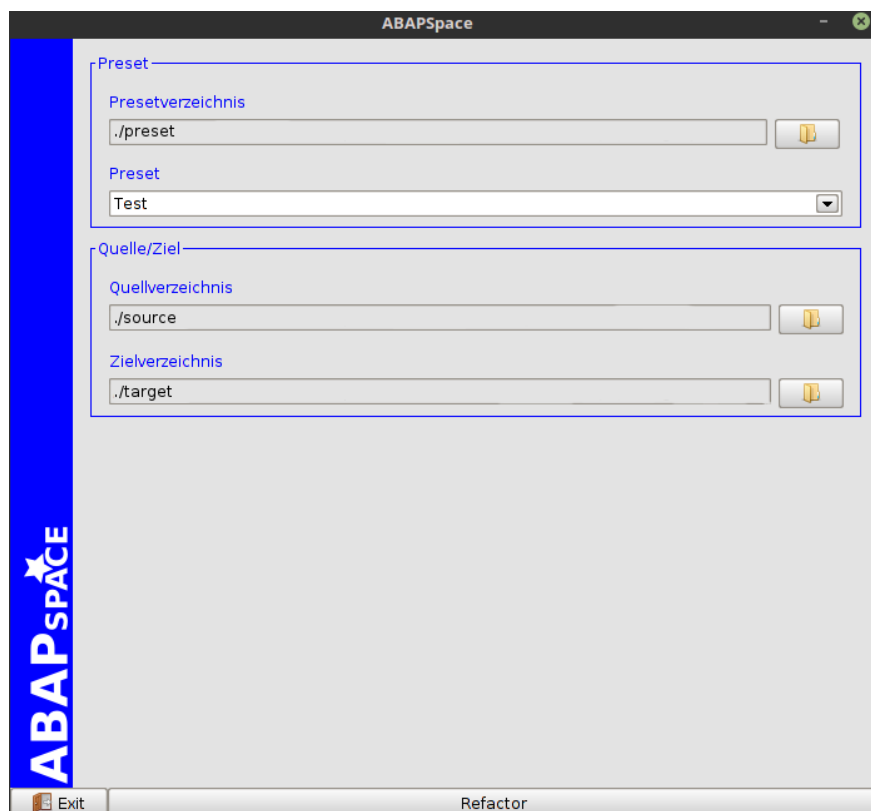


Illustration 1: ABAPSpace - start screen

Preset section. Its possible to select a separate preset directory.
The standard preset directory is:

- abapspace/bin/preset

A preset can be selected via the combobox.

Source/Target section. Here you can change the source and target directory. Target and source directory can be specified as default in the preset file.

Button Refactor. Starts the refactor process based on the selected preset file. The following actions are performed:

- Determines the objects based on the source files
- Checks the maximum object name length (only if the enhanced object policy is used and activated)
- calls the edit screen (below)
- Runs the namespace adaptation. All source files are copied to the target directory. The source files remain untouched.

Button Exit. Closes the application.

Edit screen. Its possible to change the found object names before the changes are made (Illustration 2).

Neuen Objektname anpassen			
Gefundener Objektname	Neuer Objektname	Maximale Länge	Länge
zif_abapgit_definitions	/xxx/if xyz x0001 abapgit definitions	30	37
zcl_abapgit_object_clas_old	/xxx/cl xyz x0001 abapgit object clas old	30	41
zcl_abapgit_object_clas	/xxx/cl xyz x0001 abapgit object clas	30	37
ZCL_ABAPGIT_OBJECT_CLAS	/xxx/cl xyz x0001 abapgit object clas	30	37
zcl_abapgit_oo_class_new	/xxx/cl xyz x0001 abapgit	30	25

Illustration 2: ABAPSpace - edit screen

If the enhanced object policy is used, you can see which object names have exceeded the maximum length.

When you're done with changes, click refactor or cancel.

LIST OF REFERENCES