PolyString

• Introduction
• Normalization
• Future
• See Also

Introduction

PolyString is an unusual kind of animal. It is a built-in data type for polymorphic string. This string maintains extra values in addition to its original value. The extra values are derived from the original value automatically using a normalization code.

```xml
<user>
  <fullName>
    <orig>Count Felix Téléké from Tőlőkő</orig>
    <norm>count felix teleke from toloko</norm>
  </fullName>
</user>
```

PolyString is currently used to support national characters in strings. The PolyString contains both the original value (with national characters) and normalized value (without national characters). This can be used in expressions e.g. to generate username that does not contain national characters or is a transliteration of the national characters. It deprecates the need to use custom conversion routines and each expression and therefore it brings some consistency into the integration code.

The normalized value can be used for uniqueness checking, therefore avoiding use of object names that can be confusing (e.g. only lowercase/uppercase differences) or names that look almost the same (characters that looks the same but have different unicode codes).

But the most important reason is data storage. All the values are stored in the repository therefore they can be used to look for the object. Search that ignores the difference in diacritics or search by transliterated value can be used even if the repository itself does not support that feature explicitly.

MidPoint data processing layer (Prism) is designed with syntactic shortcuts in mind. And there is a nice syntactic shortcut for PolyString as well. In all data supported data formats simple PolyStrings can be specified as strings:

```xml
PolyString syntactic shortcut

<user>
  <fullName>Count Felix Téléké from Tőlőkő</fullName>
</user>
```

The system will make sure that such PolyString is properly processed and normalized.

Partial implementation

Even though PolyString is a native part of midPoint since midPoint 2.0, it still does not have its full potential (midPoint 3.x, 4.x). But even the basic PolyString capabilities that we currently have make the system more useful. In case that you are interested in completing this feature, please use the influence that comes with Platform subscription.

Normalization

There are several pre-built normalizers for PolyString that can be used in midPoint deployment. And there is an (experimental) option to create a completely custom normalizer.

See PolyString Normalization Configuration page for more details.

Future

See PolyString Improvements page to learn about planned improvements to polystrings.

See Also

• Using Prism Objects
• PolyString Normalization Configuration
• PolyString Improvements