

# What's New in 2020.1

## Ability to use different separators depending on position in list

When editing nodes with cardinality (0,n) or (1,n) you have the ability to provide a custom separator between values by implementing a separator query function. This feature extends this capability by passing more arguments to the query function (namely prevNode and nextNode) so that it is now possible to provide custom separator for each pair of values.

<https://youtrack.jetbrains.com/issue/MPSSPRT-181>

## Transform/substitute menus: merge named and default menus

A technical debt (refactoring) enhancement that removed custom concepts for named (SubstituteMenu\_Named) and default (SubstituteMenu\_Default) editor menus. Both concepts were merged into their parent (SubstituteMenu). The type of the menu is then selected in the menu editor. The same refactoring was applied to Transformation menus.

<https://youtrack.jetbrains.com/issue/MPS-31588>

## Type over existing text

Type over existing text is a new feature that makes projectional editor feel a bit more like a textual editor. When typing into a textual cell and the character you just typed is already present at cursor position then the cursor will move right as if you just overwrote the existing character on screen. This feature is only triggered if the keystroke cannot be gracefully processed and would otherwise be ignored (e.g. in constant cells) or make the cell content invalid.

The feature is ON by default and you can turn it OFF in Settings / Editor / General tab.

<https://youtrack.jetbrains.com/issue/MPS-31209>

## Multi-line comments in BaseLanguage

The functionality of SingleLineComment has been slightly enhanced and new MultiLineComment concept has been introduced into BaseLanguage to support comments delimited with the /\* ... \*/ symbols like Java does.

With new single line comments you can get a new line for a Statement by hitting Enter in the end of the comment. In cases you want to continue writing a comment on a new line hit Ctrl/Cmd+Enter and new instance of SingleLineComment will be created. Also you can easily split a comment (just hit Enter inside the comment line) and merge 2 consequent comments (hit Delete in the end of the first comment or Backspace before // of the second comment).

<https://youtrack.jetbrains.com/issue/MPS-30662>

## Light patterns

Pattern language helps to define structure patterns and match nodes over them. It uses regular concept editors which make the pattern appear closely like the nodes it accepts. But for some complex editor that do not correspond exactly to concept structure it can be more comfortable to use light patterns (pattern builders) that describe required structure literally.

<https://youtrack.jetbrains.com/issue/MPS-31733>

## Error highlighting in Messages tool if editor definition throws an error

If code in editor aspect of a language throws an exception then default editor is shown for a corresponding cell in a model written on the language. Starting from 2020.1 an error is shown in Messages tool window if such broken cell is opened in the editor. It's possible to get stacktrace from the error and navigate to a problem cell.

<https://youtrack.jetbrains.com/issue/MPS-31284>

## BaseLanguage enhancements inspired by the Java language

BaseLanguage now includes such features as try with resources, catch multiple exceptions, local variable type inference, default and private interface methods. No additional language import is required to use these features since they are part of baseLanguage. To provide compatibility with build processes requiring certain java language level, each solution can be configured to prohibit language feature not compatible with required language level.

<https://youtrack.jetbrains.com/issue/MPS-31497>

## Tooltips

The editor extension <http://mbeddr.com/2014/06/25/tooltips.html> has been added to MPS platform. The migration will be provided for the mbeddr language.

## Editor declaration and module dependencies

MPS used to demand 'extends' dependency between languages when an editor of a language provided editor components for concepts of another language. This sort of dependency doesn't necessarily reflect the relation between the languages and sometimes leads to exhaustive language hierarchy processing (Consider INamedConcept from j.m.lang.core. Any MPS language extends j.m.lang.core, therefore editor aspects of all available languages are consulted the moment anyone uses the INamedConcept editor component). In 2020.1, generator for Editor aspect detects editor extensions for other languages and generates respective code that manifests the dependency explicitly, so that language designer doesn't need to care about particular dependency kind between the languages. 'Extends' dependency between language modules is no longer needed to get editor extensions activated.

## Javadoc comments in Java stub models

MPS bridges Java libraries to MPS world using Java stub models. These models are built from .class information and therefore lack any documentation like javadoc. In 2020.1, there's a new mechanism to specify a zip that holds sources of a library along with the library jar, and MPS is capable to extract javadoc from sources to present it along with information extracted from class files. This functionality comes handy when one uses a Java library from within MPS and needs to access library's javadoc - it's much easier to navigate to relevant code and its documentation right in MPS rather than switch to external location, perform lookup and read it there. It's even more important for MPS itself, as it provides essential parts of its API as Java stubs. Over the years, API's change, parts get deprecated and replaced with an alternative. Even though the team strives to provide 'deprecated' comments and transition guidance, it was quite inconvenient for MPS adopters to keep MPS sources around to access these valuable pieces of knowledge. Now, with the support to extract javadoc for Java stub models, MPS can serve its adopters better.

<https://youtrack.jetbrains.com/issue/MPS-29945>