MPS screen-casts

This page lists and categorizes the MPS-related screen-casts published at the MPS channel of JetBrains TV.

Customer stories

Who and how uses MPS. Watch the MPS customers presenting their use of MPS.

- Why JetBrains MPS - an overview video about MPS, who should care and why
- The Voice Menu IDE - a more thorough example of what MPS can do. A voice menu definition language is used as a sample domain to illustrate the potential of MPS.
- Why the mbeddr.com project uses MPS - mbeddr.com explores the benefits of language extension and formal methods for embedded software development. We have implemented the C programming language in MPS, which allows us to extend it in a meaningful way, for example, with state machines, physical units or with support for product line variability. In this screencast we briefly explain why we chose MPS for this project.
- Modelwerkstatt application of MPS (part 1 and part 2) - modellwerkstatt.org in Innsbruck, Austria use MPS to develop powerful DSLs for building enterprise applications. The whole development process is realized with MPS. Watch this two-part screen-cast to get a taste of what MPS brings to the table in this particular domain.

First steps

Two videos giving you the initial insight into how MPS works

- The first run - essential information for total beginners into how MPS is organised, how it works and how users are supposed to interact with it.
- The MPS projectional editor - an explanation of the principles of the MPS projectional editor, the differences from textual editors and the benefits of this approach
- Your first date with JetBrains MPS - an introductory screen-cast showing how to use the bundled sample Robot Kaja lan guage
- Your second date with JetBrains MPS - an introductory screen-cast showing how the bundled sample Robot Kaja langua ge has been defined
- Creating your first language in JetBrains MPS - a gentle introductory guide through creating a minimalistic language that exposes the basics of structure, editor and generator definition in MPS.

Introduction to MPS

Watch this series of short introductory videos to get a bird's-eye-view of how languages get defined in MPS. After watching the whole series you'll know the steps necessary to define a language and you'll better understand the whole picture. (TODO)

1. Introduction to JetBrains MPS, part 1: Projects - This episode covers the MPS project setup and organisation of modules and models including their dependencies.
2. Introduction to JetBrains MPS, part 2: Structure - This episode provides a brief theoretical background into models, meta-models and abstract syntax trees and then applies the knowledge to the MPS structure aspect.
3. Introduction to JetBrains MPS, part 3: Constraints - This episode details the constraints aspect of language definition in MPS, mainly how to restrict properties, links and how to define scopes.
4. Introduction to JetBrains MPS, part 4: Behavior - This episodes adds a few useful tips on adding functionality to concepts and nodes.
5. Introduction to JetBrains MPS, part 5: Editor - Make'em see your code - defining editors that project the AST on the screen
6. Introduction to JetBrains MPS, part 6: Actions - Polish the editors - smoothing the editing experience by defining transformations and substitutions
7. Introduction to JetBrains MPS, part 7: Intentions - Assist the developer with context-sensitive hints and refactorings
8. Introduction to JetBrains MPS, part 8: Generator - Converting models - defining model-to-model transformations
9. Introduction to JetBrains MPS, part 9: Text-Gen - Here's what I've written - converting models to text
10. Introduction to JetBrains MPS, part 10: Dataflow - Go with the flow - defining dataflow definitions so that MPS could automatically detect issues in code structure
11. Introduction to JetBrains MPS, part 11: Type-system - They are just my type - defining types and type-system rules to validate expressions early-on

You may also watch these as a single YouTube playlist.

Commanding the MPS IDE

These screen-casts should help you become fluent with the projectional MPS editor and various IDE aspects of MPS.

- MPS Editor Tips - Watch this introductory screen-cast and learn how to command the MPS editor. You'll see how to

navigate around your code, refactor it, invoke intentions, select regions and a lot more. If you're new to MPS and want to master the MPS projectional editors, this screen-cast is for you.
- Running MPS solutions - This introductory screen-cast aims to show the ways to run MPS solutions. Using the MPS calculator tutorial project as a reference you'll learn how to generate runnable Java code, create and manage run configurations and integrate MPS into your Java projects.
- Context Assistant - This video shows the use and definition of context assistant for the sample robot Kaja language.
- Context Actions Tool - This video shows the use and definition of context actions tool.
- Context Assistant for language definition - This video highlights the guidance that Context Assistant provides within the language definition languages - structure and editor.
- Transformation Menu Language - This video explains the new Transform Menu Language, which is used to define side-transformations, substitutions as well as other context-sensitive actions in the editor.

MPS basics

Videos covering the core principles of building DSLs with MPS

- MPS basics - creating your first language - This introductory screen-cast should help MPS new-bies to get over the common difficulties that they are likely to face when creating their first DSLs. We'll go through the three fundamental steps of creating a DSL - defining the language structure, the editor and the generator. Watch this short demo if you are new to MPS and want to dip your toe in the basics of the DSL definition process.
- MPS basics - enhancing the language of constants - This second in the series of introductory screen-casts aiming at MPS freshmen builds on the simple language for constants created in the previous episode, polishes several of its glitches and enhances the DSL with expressions and variable types. We'll also touch on importing languages and defining simple constraints.
- MPS basics - types and scopes of references - In the third episode of the MPS introductory series of screen-casts we're going to enhance our experimental constant language with references, simple type system rules and scoping constraints.
- MPS basics - intentions and AST manipulation - The fourth episode of the MPS basics series demonstrates Intentions and the ways you can manipulate the AST of your Domain Specific Languages. If you want to provide a smooth and pleasurable user experience to your DSLs, this screen-cast is for you.
- MPS basics - generating text with TextGen - In the fifth episode of the MPS beginners' series you'll familiarize yourself with the TextGen aspect. We'll take our MPS-based DSL and generate Ruby code from it.
- MPS basics - checking rules and quick fixes - Static code analysis helps developers discover and eliminate bugs and problems in the code quickly. By highlighting suspicious pieces of code and offering automatic refactoring to fix the code, modern IDEs safe development time and reduce the number of software defects. MPS expands this capabilities to the field of domain specific language.

MPS basics - screen-casts covering the bundled sample projects

- The Java Extensions Sample
- The Simple State Machine (SecretCompartment Language)

Interoperability

MPS doesn't live in vacuum. It can be used standalone as well as integrated into other development environments. Check out these screen-casts for the details.

- MPS languages inside IntelliJ IDEA - a parallel for loop example - The most notable new feature of MPS 2.5 is its integration with IntelliJ IDEA. You can use your DSLs and language extensions directly in your favorite Java IDE tightly integrated with your Java code. This screen-cast shows how this integration works and feels.
- Building an IntelliJ IDEA language plugin with MPS - This screen-cast will show you how to package your languages to make them available for IntelliJ IDEA. We'll create a simple properties language, define a build script to generate and package the language and then use the properties language in IntelliJ IDEA directly.
- How to package your DSLs for IntelliJ IDEA - Watch this screen-cast to learn how to build and package your languages so that they can be shared with others and used inside both MPS and IntelliJ IDEA. We'll create a build script for a sample language that will compile, build and package the language and create a proper IntelliJ plugin zip file.
- How to enable MPS in IntelliJ IDEA - Watch this screen-cast to learn how to add MPS core capabilities to IntelliJ IDEA and how to import third-party languages. We'll go through the various options that enable the MPS plugin inside IntelliJ IDEA, configure the plugin, create our first demo and run it. We will then import a separate language plugin in order to use the contained language in our code.

Language Extension

Extending languages is one of the fundamental advantages of MPS. Watch these screen-casts and learn how to improve, customize and build upon existing languages.

- Language Extensions - Creating a new statement - Watch this screencast to learn about the MPS basics.We will discuss the internals of a simple language extension and explore the details of the language design process. This screen-cast is

the first part of a series, which aims to gradually introduce the listeners into the fundamental principles of MPS.
- Language Extensions - Creating tabular expressions - In the second installment of the series we explore the implementation of a custom decision table expression. We will show how to define the structure, the editor and the type system rules for tables as well as the implementation of a generator, which transforms the table into a BaseLanguage expression. Watch this video to get a better command of tables or other non-trivial language constructs.
- Language Extensions - Creating new types and literals - In the third installment of the series we explore a custom money language. We will show how to customize the editor behavior through actions, how to override operators for your custom types, how to build a DSL on top of an existing Java library and most of all how to multiply money with MPS.
- Language Extensions - Describing the dataflow - In this episode of the series wehttp://www.vaclavpech.eu/ investigate the dataflow aspect of language extensions. Watch the screencast and learn how to let MPS detect unreachable code, potential NullPointerExceptions or initialization problems for your custom language extensions.
- Language Extensions - Customizing suggested variable names - In the fifth episode of the series we briefly discuss the possibility to customize the list of suggested variable names. Each time the user creates a new variable either directly or through the "Introduce variable" refactoring, he or she is offered suggestions for names. While the default implementation would offer a name based on the actual type, designers of Base Language extensions may hook into the process and offer custom-tailored suggestions. This is a good example of how MPS lets you extend languages on a very fine-grained level.
- Language Extensions - Creating intentions - This time http://www.voelter.de/we will guide you through intentions. Intentions, (also called quick-fixes) offer users quick context-sensitive hints with suggestions for improving or altering the code at hands. Focusing on both plain and surround-with intentions, the screen-cast shows how to create them, hook them to the appropriate language elements and obviously also how to use them to your benefit.
- Memoization for java
- Tail Recursion Optimization for Java
- A parallel for loop for Java

Advanced aspects of language definition

These screen-casts dive deeper into the waters of MPS language definition.

- Aspects of language definition - Typesystem - Watch this episode of the MPS screen-casts to learn more about the MPS typesystem. We discuss several ways to define types for custom languages and various aspects of the type resolution process in MPS.
- Smoothing the Editor Experience
- Optional Visibility in the Editor
- Using Editor Components
- Editor Transform Actions
- Node Substitution in the Editor
- Using the MPS Console to investigate and update the user models
- Aspects of language definition - Editor - In this episode we dive deeply into the editor aspect of language definition. On top of the fundamental principles of the editor specification we will also see ways to define optional elements, to control the cursor position and to customize the layout of the editor components.
- Aspects of language definition - Generator - Watching this episode you will learn about the details of code generation in MPS. The screen-casts will guide you through both text generators and model-to-model transformations. You will hear about the essential concepts of the MPS generator language, such as templates, macros or labels and see them all used in a concrete language definition. If you want to familiarize yourself with the code generation aspect of MPS, this screen-cast is for you.
- Extending existing languages through attributes - Watch this screen-cast to see how to leverage attributes in order to enhance capabilities of languages that you have no control over. Attributes allow you to add extra elements as children to existing concepts from other languages. They give you a powerful tool for language extension. Alongside the main topic you will get a brief tour round the new SampleJavaExtension sample project and find out what interesting aspects of MPS it covers.
- Checkpoints, cross-model generation and generation plans - This video shows how to define generation plans in order to enable cross-model generation. It also highlights th persisted checkpoint models, which help debug the persisted mapping labels.

Tutorials

These videos accompany the introductory Calculator Language tutorial.

- The introductory Calculator Language Tutorial - This screncast provides an alternative way to learn MPS. Folowing the steps described in the on-line introductory MPS Tutorial the screencast will guide you through the process of building a Calculator definition language and show you how to use the basic MPS concepts, such as the structure, the editor and the generator. If you find the on-line tutorial a bit dry and difficult to follow, this screencast may help you get over it.
- The introductory Calculator Language Tutorial - constraints and type-system - This is a second part of the Calculator Language Tutorial covering constraints and type-system. This screncast provides an alternative way to learn MPS. Folowing the steps described in the on-line introductory MPS Tutorial the screencast will guide you through the process of creating scoping constraints and simple type-system rules for the Calculator Language we built in the first part.