



# **Agilingua**

# **ATOM Grammar Tool**

## **User Manual**



## Table of Contents

<a href="#">Agilingua .....</a>	<a href="#">1</a>
<a href="#">ATOM Grammar Tool.....</a>	<a href="#">1</a>
<a href="#">User Manual.....</a>	<a href="#">1</a>
<a href="#">Table of Contents.....</a>	<a href="#">3</a>
<a href="#">1 Installation.....</a>	<a href="#">4</a>
<a href="#">2 Opening and Closing Files.....</a>	<a href="#">4</a>
<a href="#">2.1 Supported Files.....</a>	<a href="#">4</a>
<a href="#">2.2 Creating Files.....</a>	<a href="#">4</a>
<a href="#">3 Editing Grammars.....</a>	<a href="#">6</a>
<a href="#">3.1 Text Editing Functions.....</a>	<a href="#">6</a>
<a href="#">3.2 Navigation.....</a>	<a href="#">7</a>
<a href="#">4 Debugging Grammars.....</a>	<a href="#">8</a>
<a href="#">5 Creating and Running Grammar Tests.....</a>	<a href="#">8</a>
<a href="#">5.1 Create a new test set.....</a>	<a href="#">8</a>
<a href="#">5.2 Adding sentences to a test set.....</a>	<a href="#">9</a>
<a href="#">5.3 Generating Parse Trees and Semantic Representations.....</a>	<a href="#">12</a>
<a href="#">5.4 Running Tests.....</a>	<a href="#">13</a>

# 1 Installation

You can install the ATOM Grammar Tool by double-clicking the installers icon and following the instructions.

## 2 Opening and Closing Files

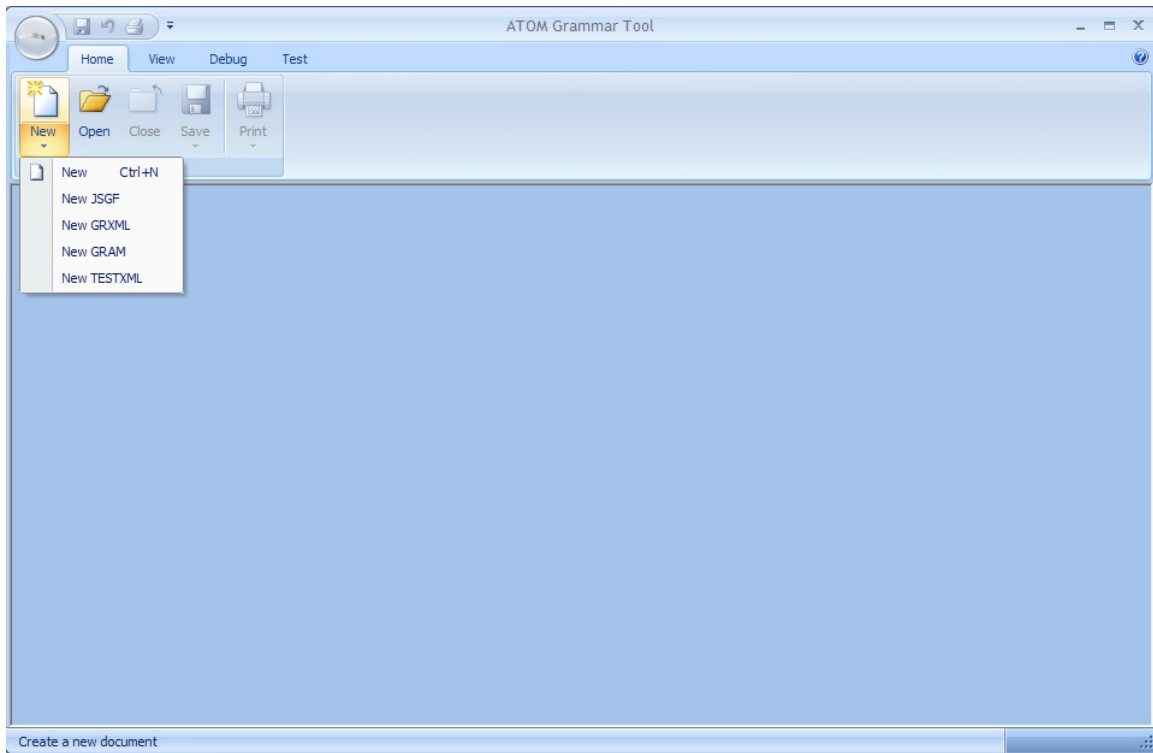
### 2.1 Supported Files

There are four file types supported:

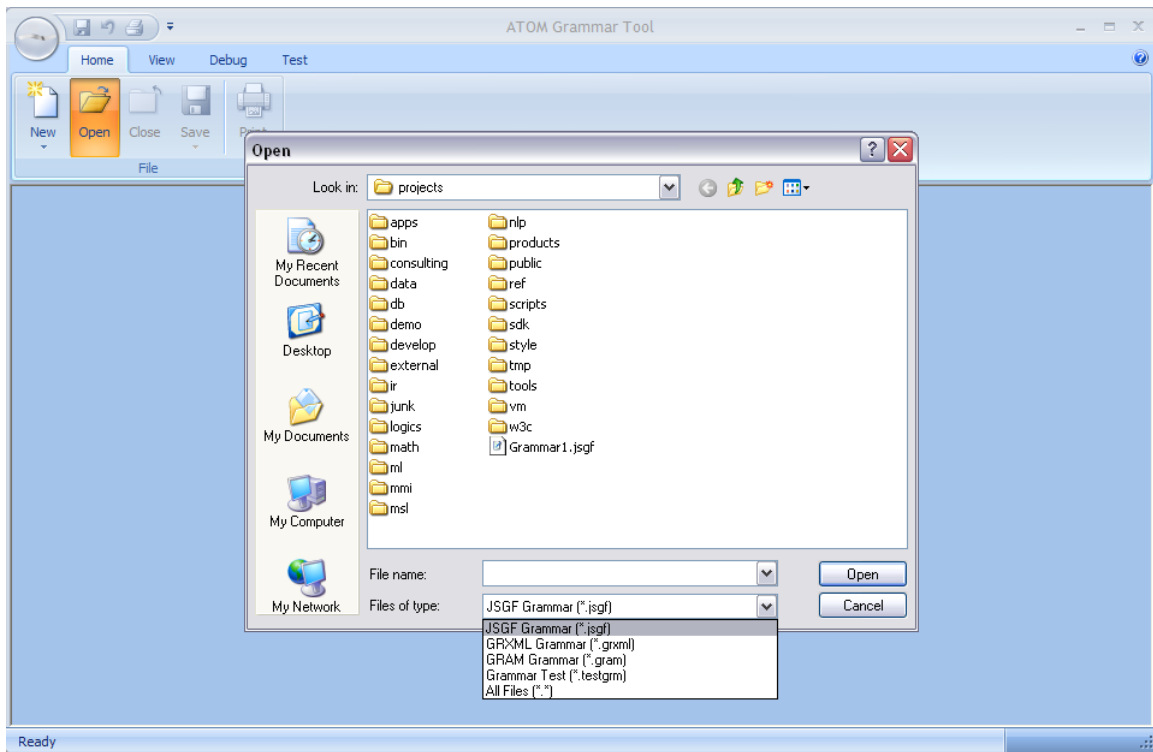
1. JSGF files. These are files in the Java Speech Grammar Format that have the .jsgf extension
2. W3C SRGS ABNF Files. These are files in the W3C Speech Recognition Grammar Specification format that have the .gram extension
3. W3C SRGS XML Files. These are files in the W3C Speech Recognition Grammar Specification XML format that have the .grxml extension
4. Proprietary test files. These files are used to store test sets and have the .testxml extension.

### 2.2 Creating Files

To create files, click on the little arrow underneath the New button in the Home tab. A menu appears from which you can choose the desired file type.



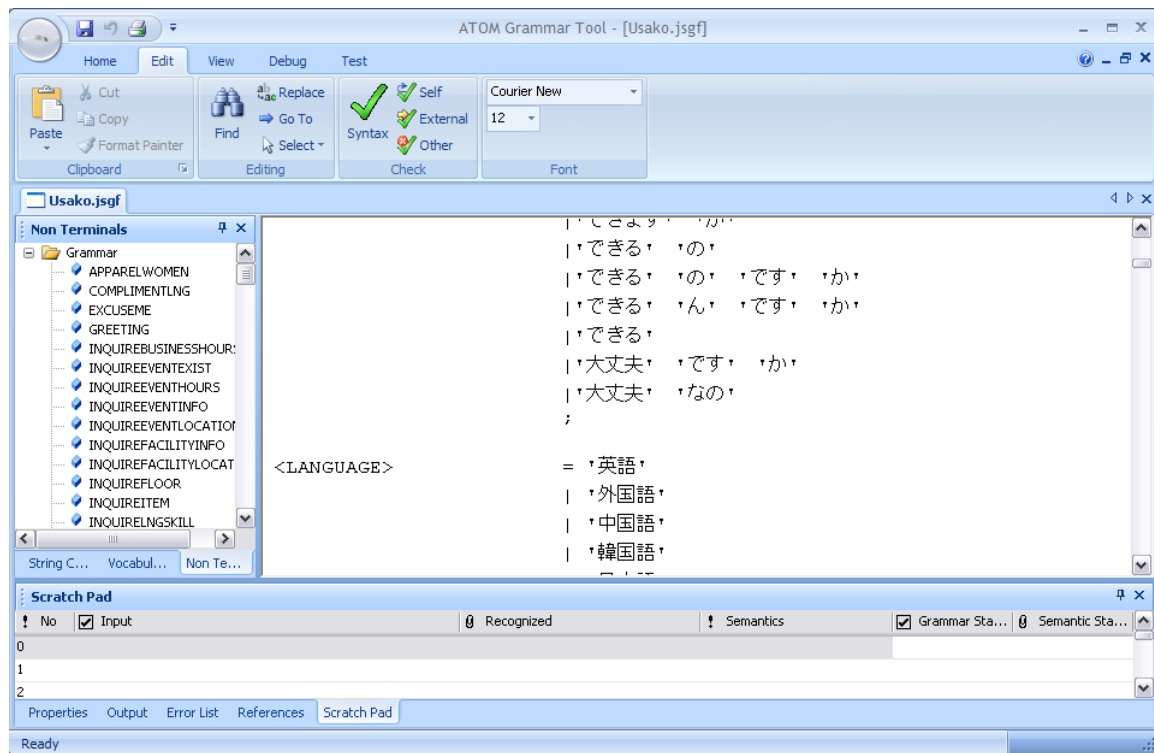
To open an existing file, click on the Open button and choose the desired file type in the file type selector in the dialog box that appears.



Files can be closed by clicking the button marked with an ‘x’ in the upper right hand corner of the files’ window.

### 3 Editing Grammars

The editing window is separated into three different areas. The main area, the grammar editor is used for writing grammars. The left area, the navigation pane, is used to quickly jump to rule definitions or words in the grammar. The lower area displays different information on the grammar being edited.



#### 3.1 Text Editing Functions

The ATOM Grammar Tool provides standard text editing functions. In addition, grammar specific functions for syntax checking are provided. These functions can be accessed through the ribbon on the top of the window.

The leftmost group of the ribbon, labeled ‘Clipboard’ lets you cut, copy and paste parts of the grammar as in any standard text editor.

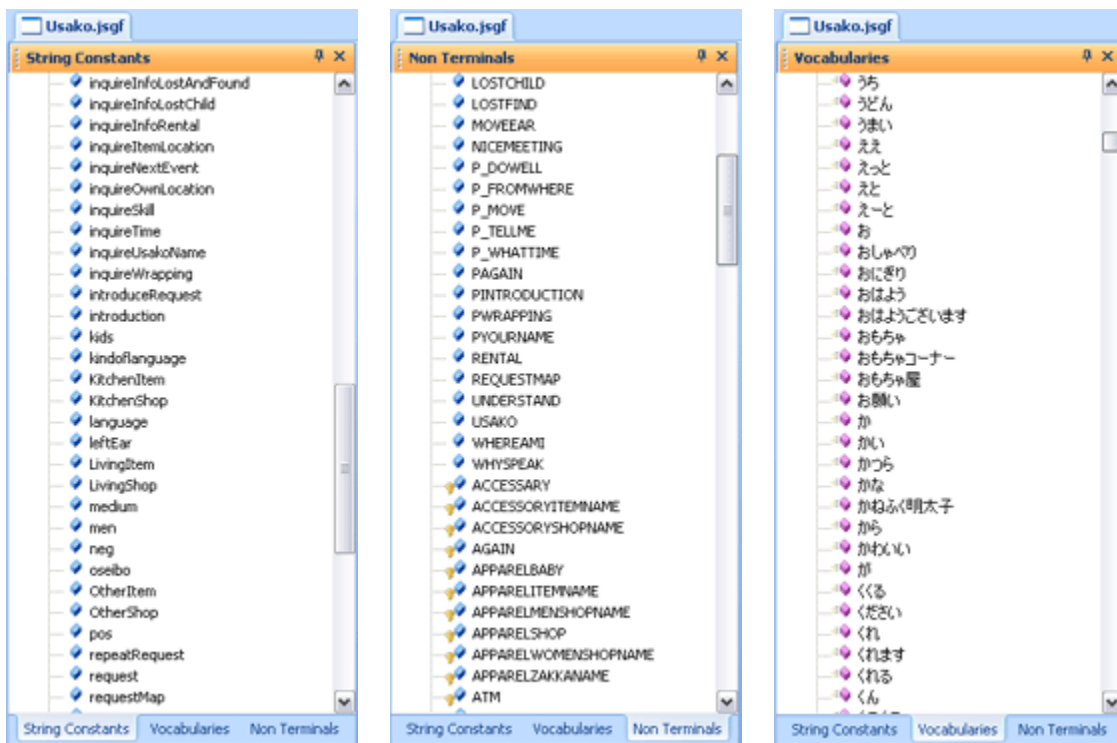
The next group, labeled ‘Editing’, lets you find and replace parts in the grammar text. Using the Go to function, you can go to a specific line in the grammar text.

The third group in the ribbon labeled ‘Check’ lets you perform checks to see whether the grammar file is syntactically correct. The following checks are provided:

1. Syntax. This check function displays errors whenever the grammar file does not conform to the standard.
2. Self. The self test runs test sentences through the grammar and displays whether the sentences are covered or not. The test sentences are associated with the grammar by means of the Meta tag. See the W3C SRGS standard for more information on the Meta tag.
3. The External check determines whether references to external rule definitions are valid.
4. The final check, labeled ‘Other’, performs a series of checks on the grammar and displays the results in the output pane.

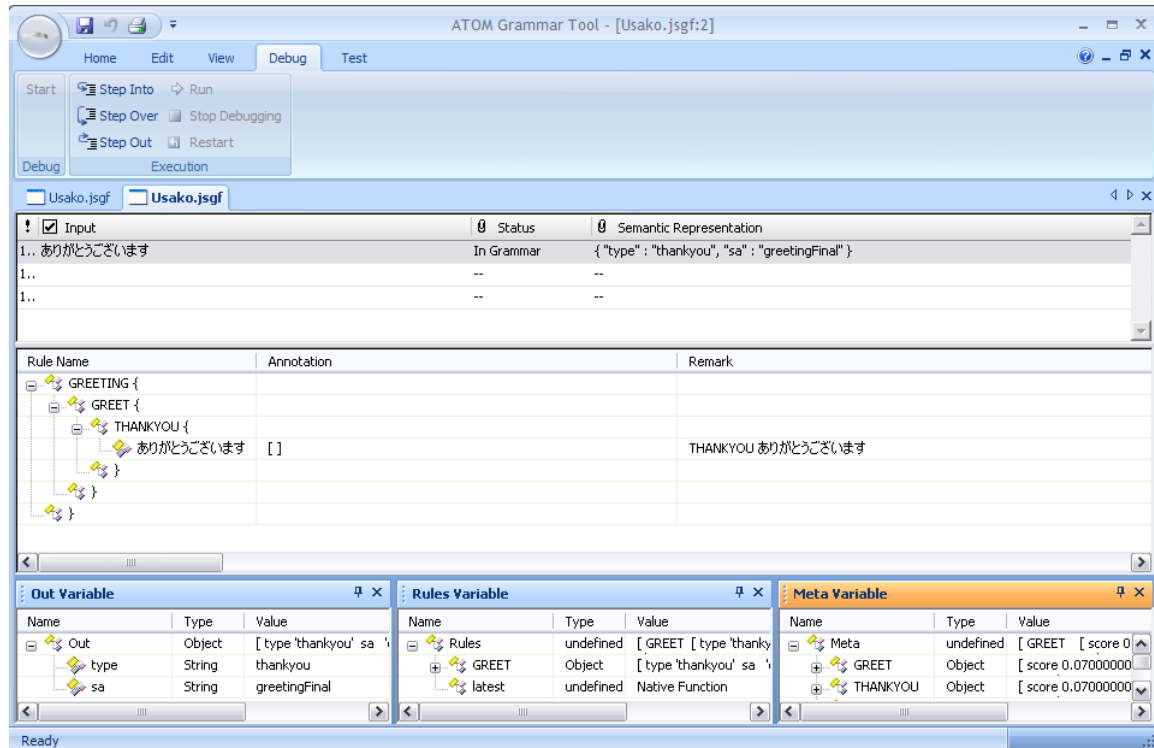
### 3.2 Navigation

The navigation pane to the left of the grammar editor can be used to quickly jump to specific parts of the grammar file. It is possible to jump to rule definition, specific words in the grammar or specific strings in the SISR annotations. The different types of targets can be accessed by clicking the tabs labeled ‘String Constants’, ‘Vocabulary’ and ‘NonTerminals’ at the bottom of the navigation pane. To jump to a specific target, click on the entry in the navigation pane.



## 4 Debugging Grammars

It is possible to interactively debug a grammar. To do so, click on the Start button in the 'Debug' tab. A new window opens. This window is divided into five parts. Test sentences for debugging purposes can be entered in the top part of the window. The middle part displays the parse (if any) that the input sentence in the top part generated. The lower part displays the values of the out, meta and rules variables, respectively. To learn more about these variables, refer to the W3C SISR documentation.

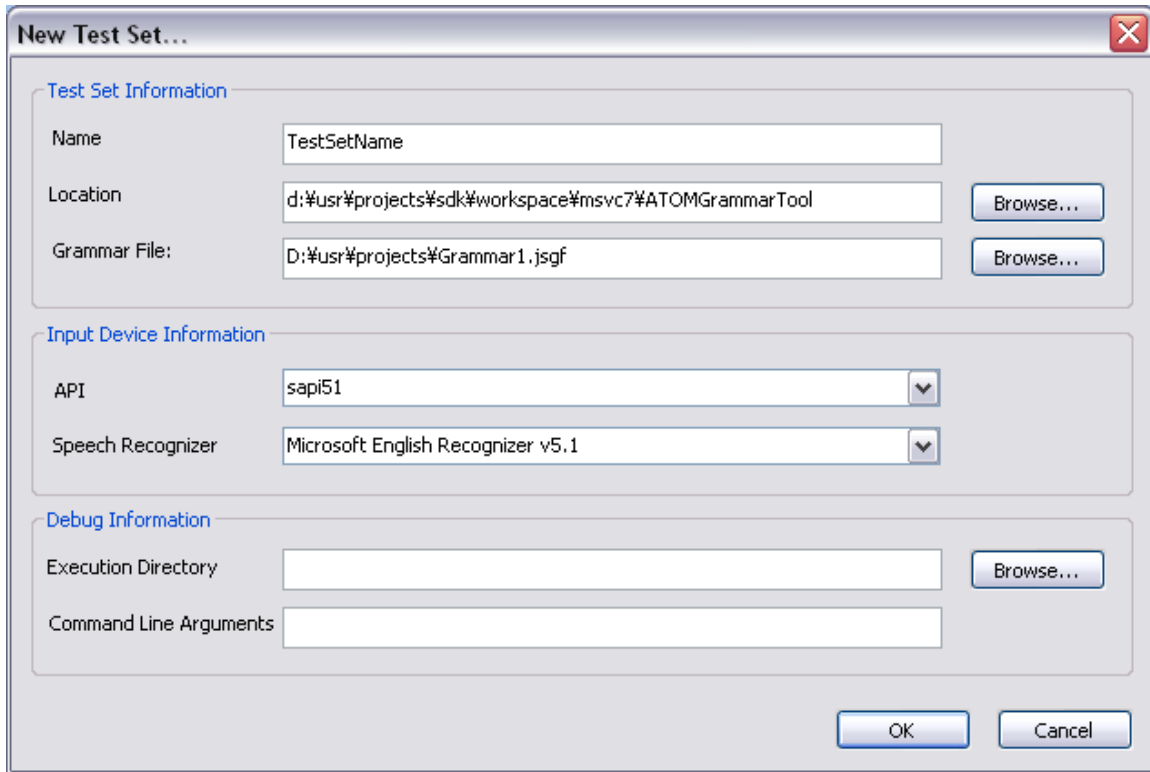


To Debug a sentence, click the first line in the input section and type the sentence to be tested. Then, click the Return button twice. If the sentence is in grammar, a parse tree will appear in the middle part. You can now step through the parse tree using the navigation buttons in the ribbon.

## 5 Creating and Running Grammar Tests

### 5.1 Create a new test set

To create a new grammar test, click on the New button on the Home tab and select New TESTXML. A new dialog appears.



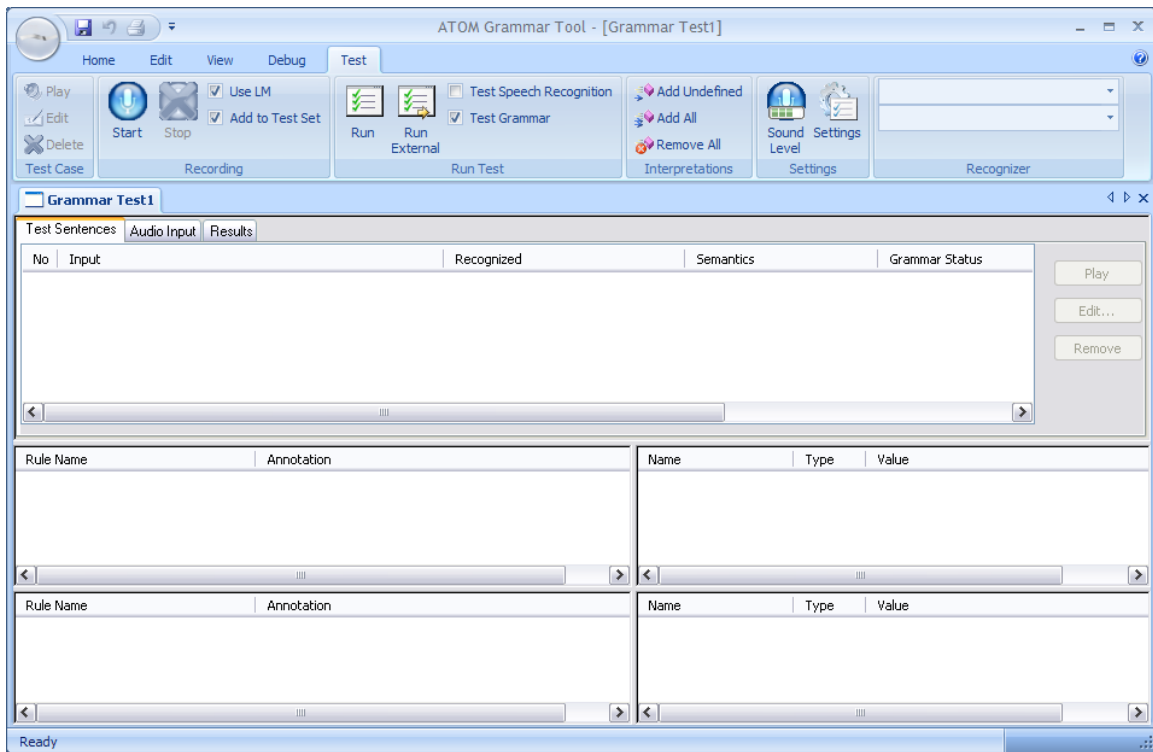
You can fill in the name and location of the test set as well as the grammar file against which the tests should be run. The next section lets you determine the speech recognizer used for testing. You can always change these settings by clicking the 'Settings' button on the Test tab.

Then click OK to dismiss the dialog. You can use the functions on the Test tab to add, modify and delete sentences from your test set.

## **5.2 Adding sentences to a test set**

Since the test set is initially empty, test sentences need to be added. This can be done by using the recording feature of the ATOM Grammar Tool. There are two check boxes controlling the way test sentences are recorded and added to the test set. First, the check box labeled 'Use LM' lets you use the built-in dictation language model of your recognizer. This option should be checked in the beginning of the grammar development process when the coverage of the grammar to be tested is not high. As the quality of the tested grammar improves, uncheck this option, and the test grammar is used for recognition purposes instead of the dictation language model. The second option, labeled 'Add to Test Set' allows you to add recognized sentences to the current test set. Uncheck this option when you want to test the speech recognizer only without adding new sentences to the test set.

Now click the Start button. This will load the specified speech recognition engine and allow you to dictate test sentences. To configure the sound level of the speech recognizer, click the 'Sound Level' button.



Once an utterance is done, a dialog box appears that lets you add transcriptions and semantic representations to the test sentence before adding it to the test set.

The screenshot shows a 'Recognition Results' dialog box with the following fields and buttons:

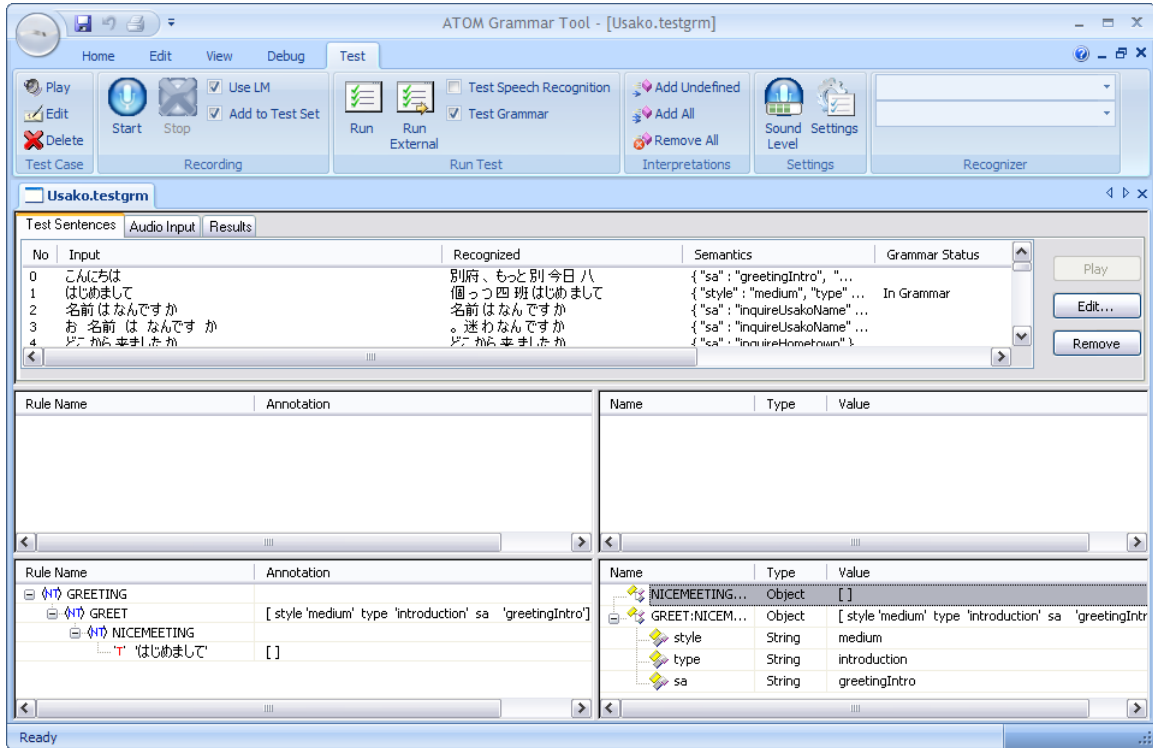
- Sound File:** TestSet/signals/signal-in-00000002.wav
- Recognition Result:** Hello hello
- Transcription:** Hello hello
- Annotation:** (empty text box)
- Interpretation:** (empty text box)
- Interpretation:** (empty table with columns Name, Type, Value)

Buttons on the right side include: Play, Check..., Check..., Copy Down, OK, and Cancel.

The name of the recorded sound file and the recognition result are displayed in the first two lines. You can replay the sound file by clicking the ‘Play’ button in the upper right corner. If the recognition is not correct, you can provide a correct transcription in the third line. You can use the ‘Check’ button to the right of the correct transcription to see whether it is covered by the grammar. If the correct transcription is not covered by the grammar, extending the grammar to cover it will likely cause a correct recognition.

The line labeled ‘Annotation’ allows you to add an annotation to this sentence. Annotations can be helpful hints and notes for grammar developers only and are not used for testing purposes. If the grammar is annotated with SISR tags and the recognition result is covered by the grammar, the semantic representation generated from the SISR tags is displayed in the fifth line. The representation is shown in JSON notation (see [www.json.org](http://www.json.org) for more information). You can provide a correct representation in the field labeled Interpretation. To add this sentence to the test set, click OK. To not add it, click Cancel.

Once you have added enough test sentences, click the Stop Recording button to stop the recording.



After adding test sentences to the test set, you can modify or delete the test sentence by clicking the Edit or Delete buttons on the Test tab. To play the sound file associated with the test sentence, click the 'Play' button.

Since adding semantic annotations may be tedious, you can add semantic annotations automatically by clicking the 'Add Undefined' or 'Add All' buttons. The behavior of these buttons is as follows. For each test sentence in the test set:

1. The semantic representation for the transcription of the test sentence is generated using the test grammar.
2. If either the semantic representation associated with the test sentence is empty, or you click the 'Add All' button, the generated semantic representation is placed in the test sentence.

### 5.3 Generating Parse Trees and Semantic Representations

The four windows below the test sentences display the parse trees and generation of the semantic representations for the currently selected test sentence. The upper two windows show the parse tree and semantic representation for the recognition result, the lower two windows show the parse tree and semantic representation for the transcription. Parse trees are shown on the left, and semantic representations are shown on the right.

Nonterminal grammar symbols in the parse trees are indicated by a blue symbol '<NT>'. Terminal symbols (or words) are indicated by a red symbol 'T' in the parse tree. The semantic representations are shown in the order they are created; that is, in the order the parse tree is traversed.

You can generate parse trees and semantic representations simply by clicking on a sentence in the test set.

## ***5.4 Running Tests***

To run a test, click on the 'Run' button. The test will iterate over all test sentences try to obtain a parse tree. If no parse tree could be obtained, the grammar status is 'Not in Grammar'. If a parse tree could be obtained for only parts of the sentence, the grammar status is 'Partially in grammar', and if a parse tree could be obtained for the entire sentence, the grammar status is 'In Grammar'. In the last two cases, the grammar tool attempts to generate the semantic representation for the test sentence. The generated representation is then compared to the correct semantic representation associated with the test sentence. If they are identical, the interpretation status is 'No Differences' otherwise it is 'Differences'. This is helpful to determine which sentences generate the expected semantic representation.

The button labeled 'Run External' is used in the Business Edition of the grammar tool only. It allows you to start external processes and add the obtained sentences to the test set.