LanguageWare Resource Workbench 7.2

Creating and using character rules

Introduction

- **Module overview:**
  - What are character rules?
  - Create a character rules database
  - Add the character rules database to the UIMA pipeline
  - Add and update character rules
- **Target audience:**
  - All audiences
- **Prerequisites:**
  - Install LanguageWare® Resource Workbench
  - Create a LanguageWare project
- **Version release date:** LRW 7.2, ICA 2.2, released October, 2010

Module objectives

After this module you will be able to:

- Understand what character rules are and when to use them
- Create a character rules database
- Add character rules to the character rules database
- Modifying a character rule to expand match scope

Module roadmap

- **Creating character rules**
  - What are character rules?
  - Creating a character rules database
  - Create and add character rules to the character rules database
  - Modifying a character rule to expand match scope
- **Summary and best practices**
- **Sample exercises**

Character rules,

What are they?

- **General idea**
  - – Character rules are character expressions created to match desired sequences of characters such as telephone numbers, email address, product identifiers, and so on.
  - – Some examples of interesting character sequences
    - 704.501.1500
    - (704)501-1500
    - username@gmail.com
    - username.lastname@us.ibm.com
    - ISBN 0-13-629841-9
    - PHILLIPS 838.450 LY
    - azur 340A
  - – Character rules create annotations when text sequences are found to match the character expression.

Character rules,
What are they?

- **Specifically**
  - You may want a character rule that identifies US telephone numbers such as **704-501-1500**. A character rule with the following character class sequence could be used:

• Decimal_Number X 3	covers the three decimal numbers **704**
• Dash_Punctuation	covers the dash character: **-**
• Decimal_Number X 3	covers the three decimal numbers **501**
• Dash_Punctuation	covers the dash character: **-**
• Decimal_Number X 4	covers the four decimal numbers **1500**

  - Written in the style of a POSIX regular expression, this would look something like
    - [:digit:]{3}[-][:digit:]{3}[-][:digit:]{4}

Character rules,
What are they?

- **Specifically**
  - Of course, this is an extremely simple case as US telephone numbers can take many forms:
    - (704)501-1500,
    - 704.501.1500,
    - 1.704.501.1500, and so on
  - Writing a regular expression to find these other forms can be challenging.
  - With the Character Rules Database editor, you can generate these character rule expressions graphically by dragging and dropping character sequences onto the Character Rules editor.
  - The generated expression can then be modified to match similar sequences of characters, such as the variations on the US telephone number presented earlier.
  - Finally, one or more annotations can be associated with the character rule expression. When the character rule finds a match, the annotations are created.

Character rules database

- A Character rules database is simply a collection of character rules. Generally these character rules are related in some way. For example, a particular database in question may contain a series of rules created to match all of the various US-style telephone number formats.
- After adding character rules to your character rule database, it must be built in order for it to be used in a UIMA pipeline. Building a database is simply a matter of right-clicking the database and choosing **Build LanguageWare Resource**.
- Using your character rules database to analyze text is simply a matter of including it in the lexical analysis stage of the UIMA pipeline. [Details follow later in this course.]

Module roadmap

- **Creating character rules**

  What are character rules?

  Creating a character rules database

  Create and add character rules to the character rules database

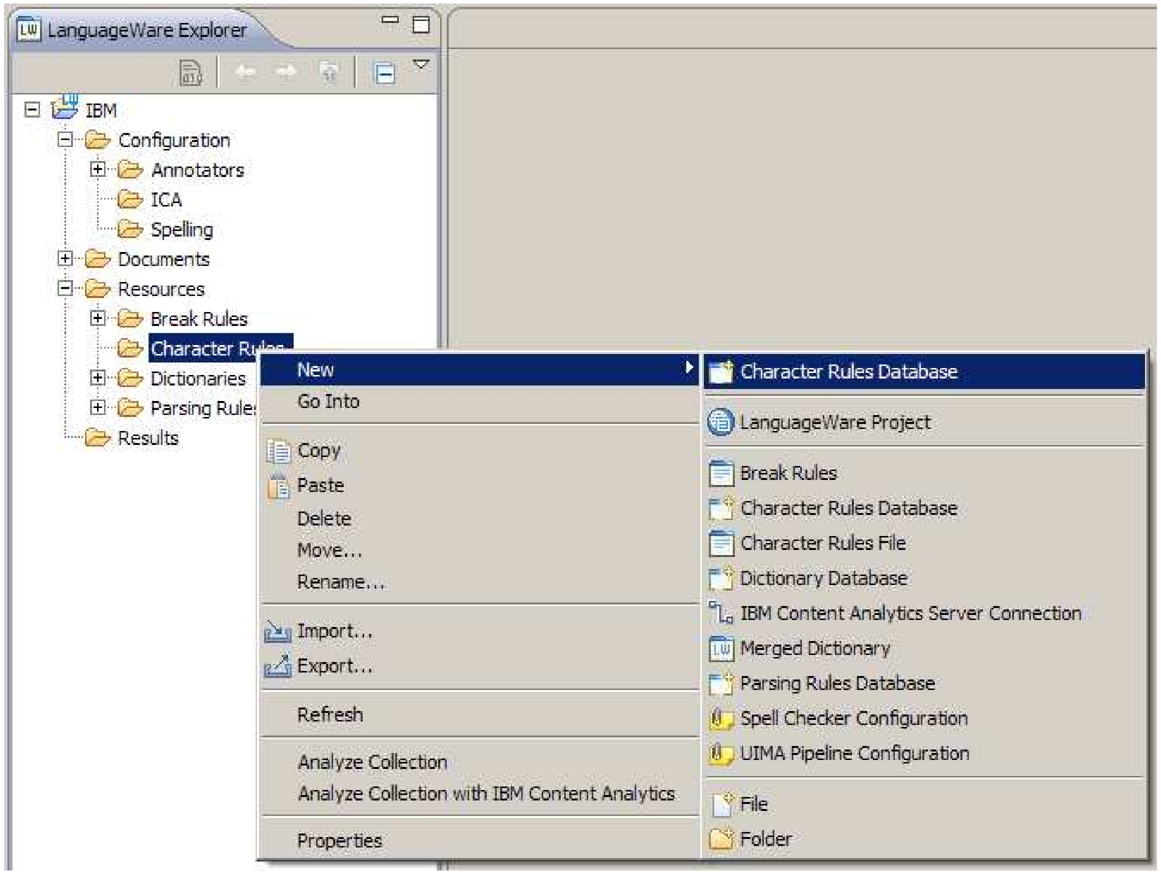
  Modifying a character rule to expand match scope

- **Summary and best practices**

- **Sample exercises**

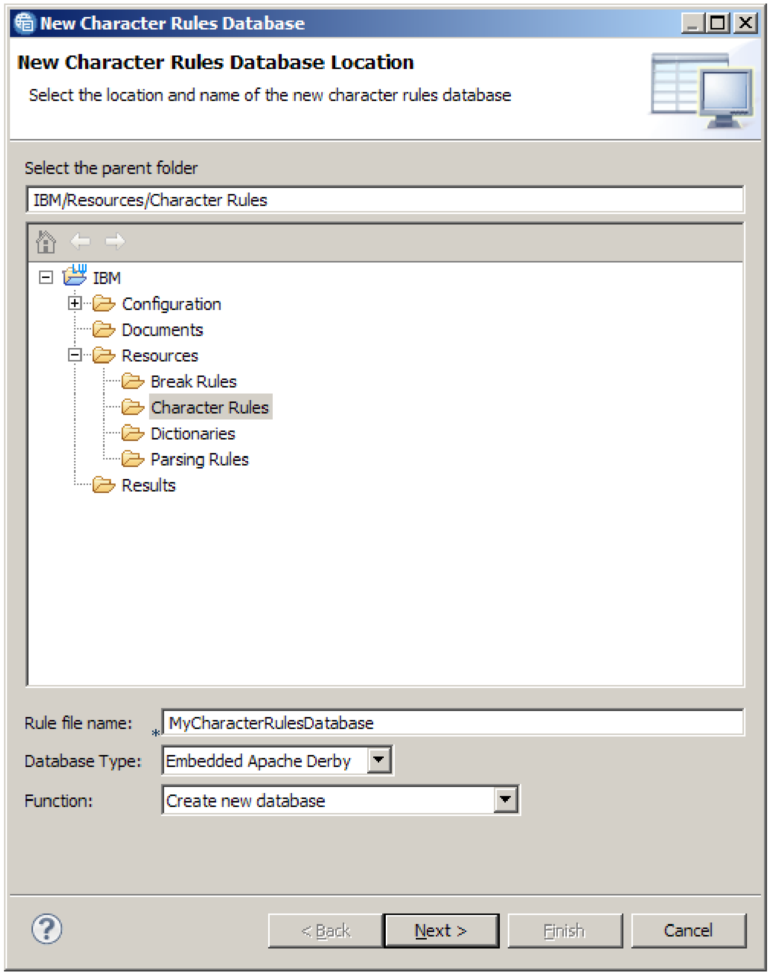
Creating a character rules database

- To create a character rules database, right-click the Character Rules folder in the Resources directory and select **New/Character Rules Database**.



Select the location for the new character rules database

- In the new character rules database location dialog:
  - Specify the parent folder where the database will be created.
  - Enter the name to use for the new character rules database. This name will also be used for the corresponding dictionary built from this database.
  - For database type use embedded Apache Derby.
  - For function use create new database.
- Click **next.**



New Character Rules Database

New Character Rules Database Location

Select the location and name of the new character rules database

Select the parent folder

IBM/Resources/Character Rules

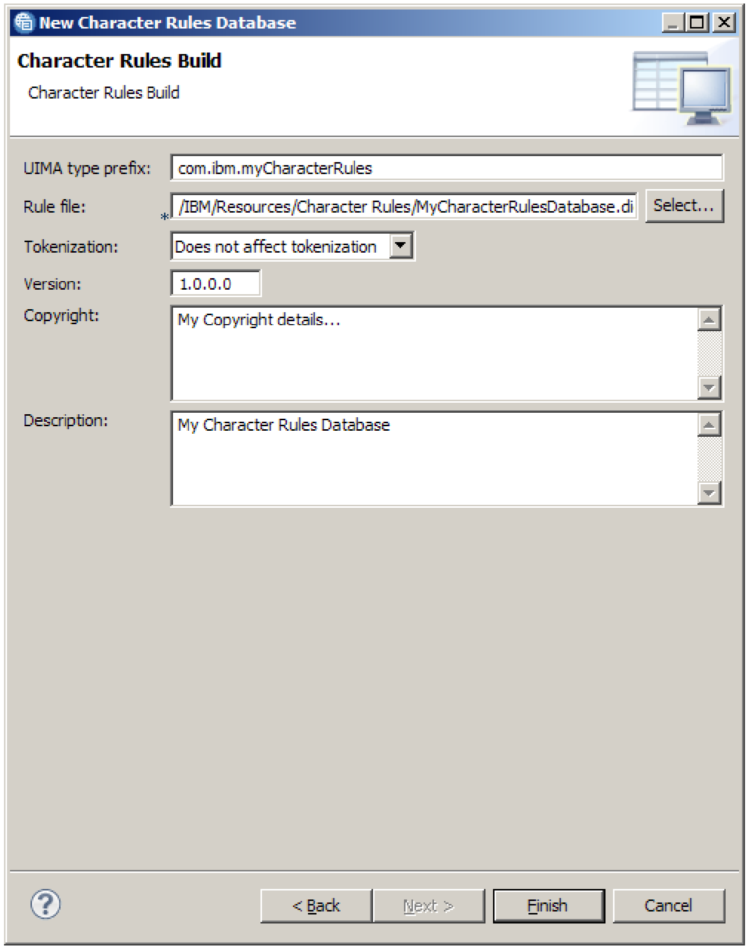
Rule file name: MyCharacterRulesDatabase

Database Type: Embedded Apache Derby

Function: Create new database

Specify the character rules database build options

- In the Character Rules Build dialog:
  - Enter the UIMA type prefix to be used as the default for this database. The annotations created by your character rules will begin with this prefix.
  - The rule file is the name used for the compiled dictionary file. Generally the default value is fine to use.
  - For Tokenization, use Does not affect tokenization.
  - Enter the Version, Copyright, and Description values to be associated.
- Click **Finish** to create the new Database.



New Character Rules Database

Character Rules Build

Character Rules Build

UIMA type prefix: com.ibm.myCharacterRules

Rule file: /IBM/Resources/Character Rules/MyCharacterRulesDatabase.di

Tokenization: Does not affect tokenization

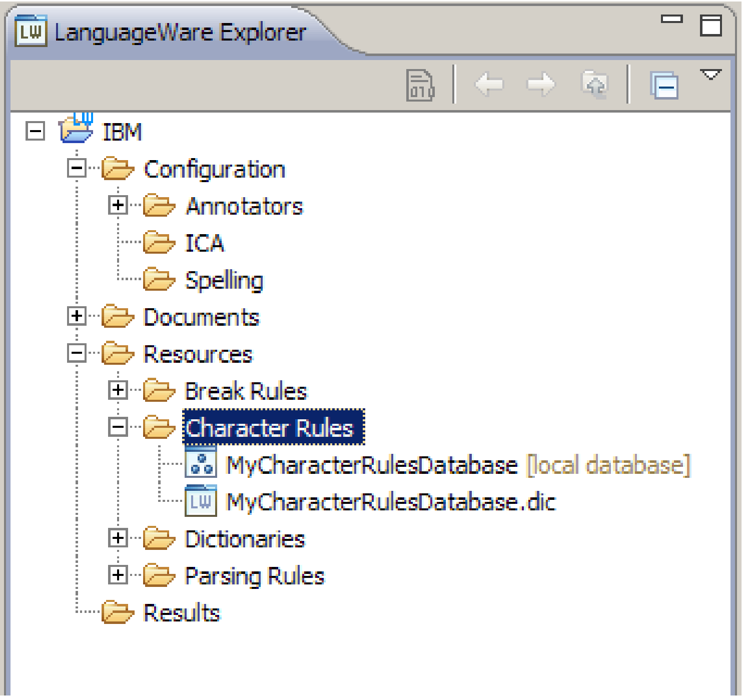
Version: 1.0.0.0

Copyright: My Copyright details...

Description: My Character Rules Database

Character rules database files are created

- New character rules database files generated when creating a new character rules database:
  - The local character rules database,
  - The associated character rules database dictionary file. The **.dic** file contains the compiled Character Rules used by the annotator.



LanguageWare Explorer

- IBM
  - Configuration
    - Annotators
    - ICA
    - Spelling
  - Documents
  - Resources
    - Break Rules
    - Character Rules
      - MyCharacterRulesDatabase [local database]
      - MyCharacterRulesDatabase.dic
    - Dictionaries
    - Parsing Rules
  - Results

Module roadmap

- **Creating character rules**

  What are character rules?

  Creating a character rules database

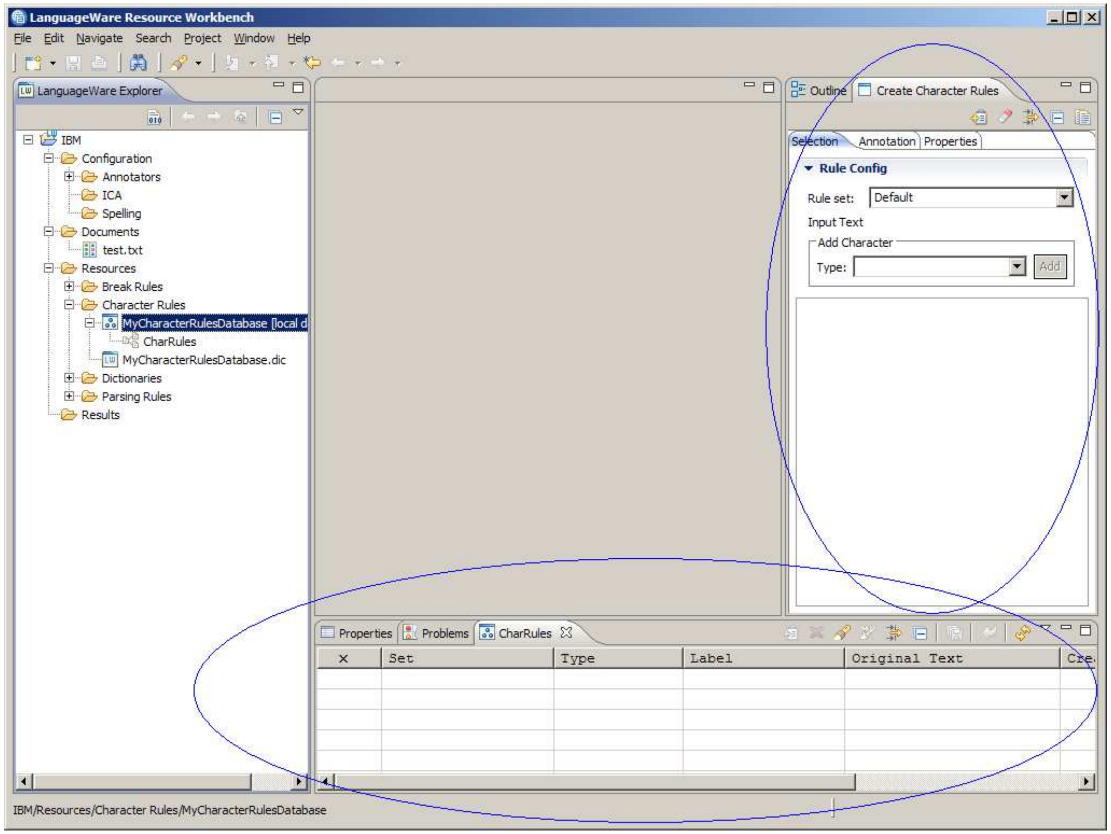
  Create and add character rules to the character rules database

  Modifying a character rule to expand match scope
- **Summary and best practices**
- **Sample exercises**

Opening the character rules database

Before you can create and add new character rules to your database, first open the character rules database.

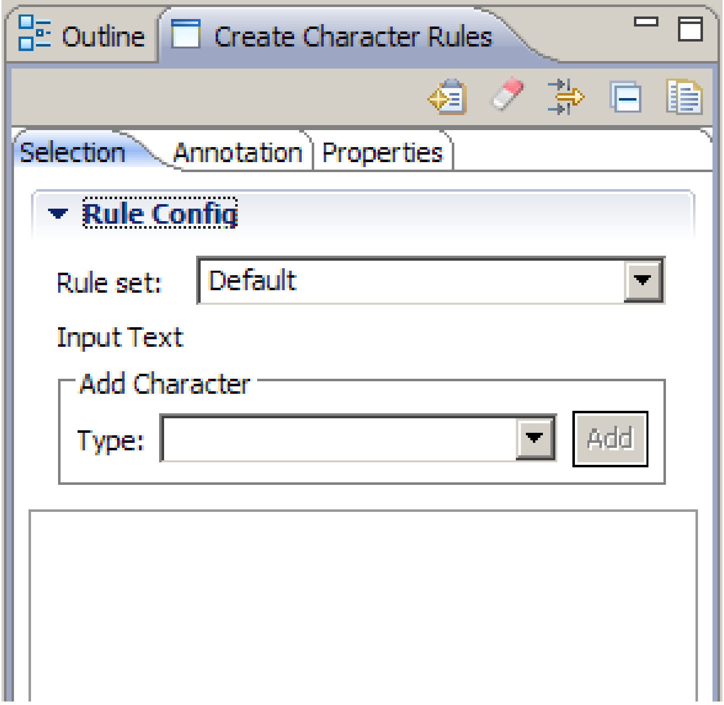
- Open the character rules database by double-clicking the database in the LanguageWare Explorer pane.
- Two views open:
  - the Database editor, displayed in the Database view, and
  - the Character Rules Editor, displayed in the Create Character Rules view.



Creating character rules: Overview

Character rules are created and modified using the character rules editor. This editor contains three tabs letting you specify various attributes of the character rule.

- When first opening the character rules database, the editor opens to its default setting—no rule is currently being edited.
- The four general steps to creating a new character rule are:
  - defining the character sequence for desired matches,
  - specify one or more annotations to be created when a match is found,
  - set additional properties for the new rule,
  - save rule and build the dictionary file.
- Before you create your first rule, you must add the character rules database dictionary file to the UIMA pipeline. This is covered on the next slide.



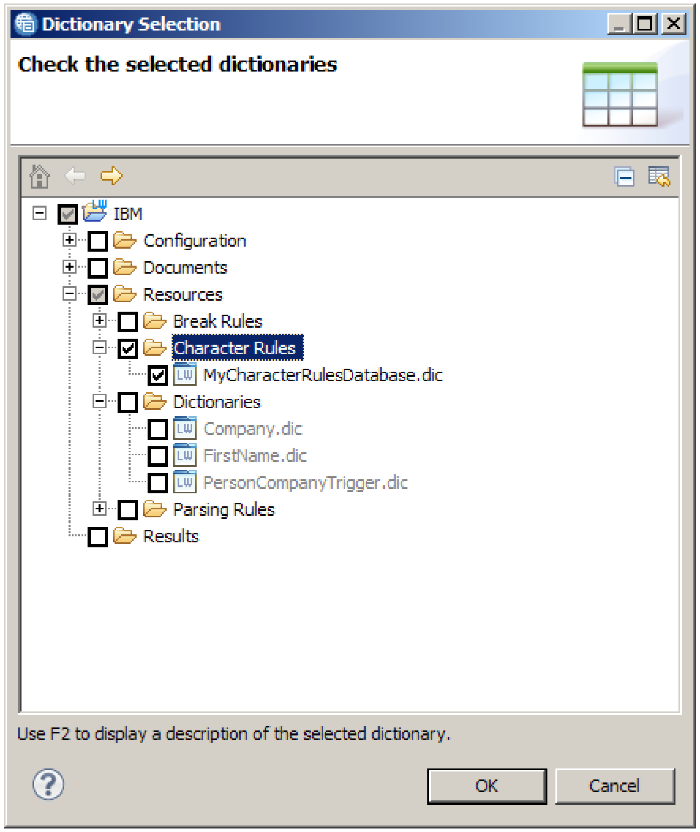
Rule set: Default

Type:

Creating character rules: Getting into the UIMA pipeline

Before you can take advantage of the drag-and-drop capabilities of the character rules editor, the text file being used needs to be annotated with the character rules database dictionary file included in the UIMA pipeline.

- Double-click **MyAnnotator.annoconfig** in the Configuration/Annotators directory of the LanguageWare Explorer pane. This opens the UIMA Pipeline Configuration Editor.
- Select the Lexical Analysis stage and choose the **Select...** button.
- In the Dictionary Selection dialog, place a check beside your Character Rules Database dictionary.
- Save these updates to the UIMA Pipeline Configuration.
- Your Character Rules Database dictionary is now included in the Pipeline.


Dictionary Selection

Check the selected dictionaries

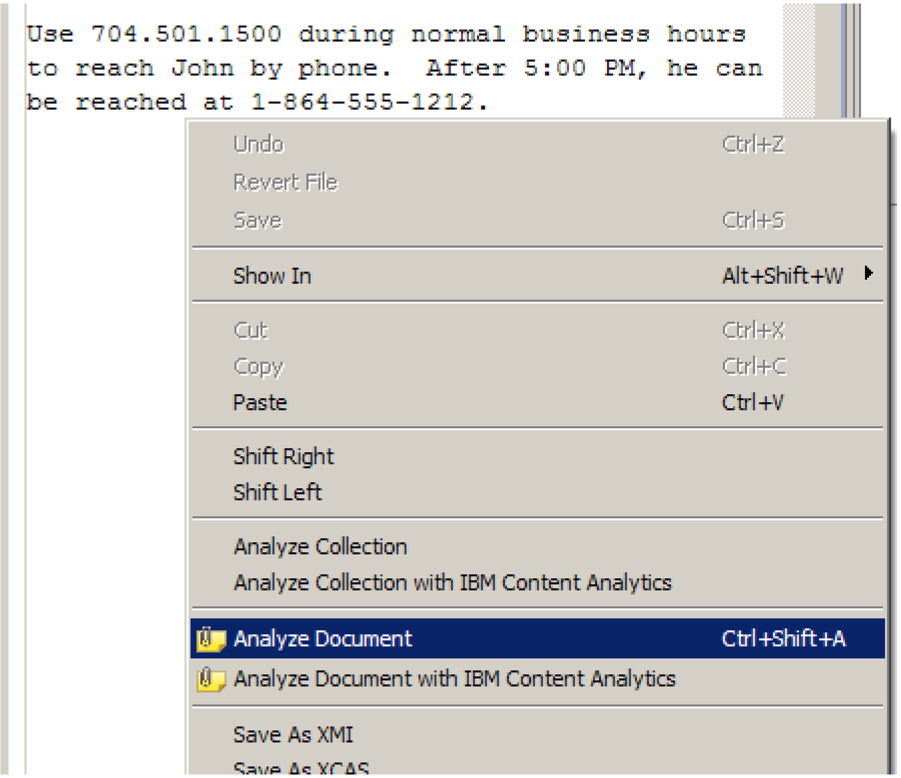
```
⊟ ☑ IBM
   ⊞ ☐ Configuration
   ⊞ ☐ Documents
   ⊟ ☑ Resources
      ⊞ ☐ Break Rules
      ⊟ ☑ Character Rules
            ☑ MyCharacterRulesDatabase.dic
      ⊟ ☐ Dictionaries
            ☐ Company.dic
            ☐ FirstName.dic
            ☐ PersonCompanyTrigger.dic
      ⊞ ☐ Parsing Rules
     ☐ Results
```

Use F2 to display a description of the selected dictionary.

Creating character rules: Running the initial analysis

After adding your character rules database dictionary file to the UIMA pipeline, an initial analysis is executed on your text document so that it can be tokenized.

- Under the Documents folder in the LanguageWare Explorer, open the text file containing the character sequences you want to match.
- Right-click anywhere in the text file editor and choose the **Analyze Document** menu item.
- In the “Select a UIMA Pipeline configuration file dialog,” choose the **MyAnnotator.annoconfig** configuration to use. Select **OK**.
- Your text document has been analyzed. You can now create some character rules.

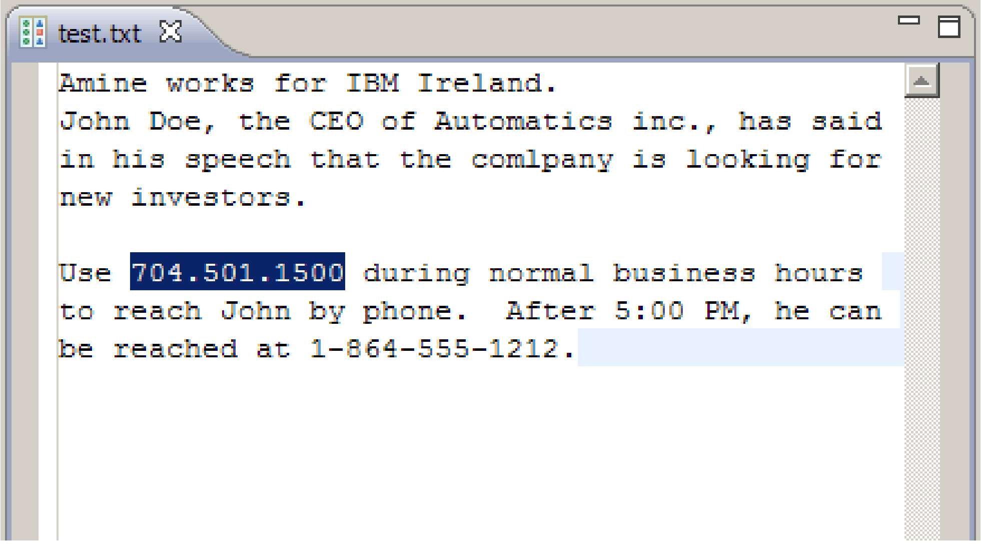


Use 704.501.1500 during normal business hours to reach John by phone. After 5:00 PM, he can be reached at 1-864-555-1212.

Creating character rules: Defining a character sequence

By far the easiest way to create a character rule is to drag an example of the character sequence that the character rule should match. In this example, you will create a character rule to match US-style telephone numbers.

- Find the telephone number **704.501.1500** in your text file and select it.

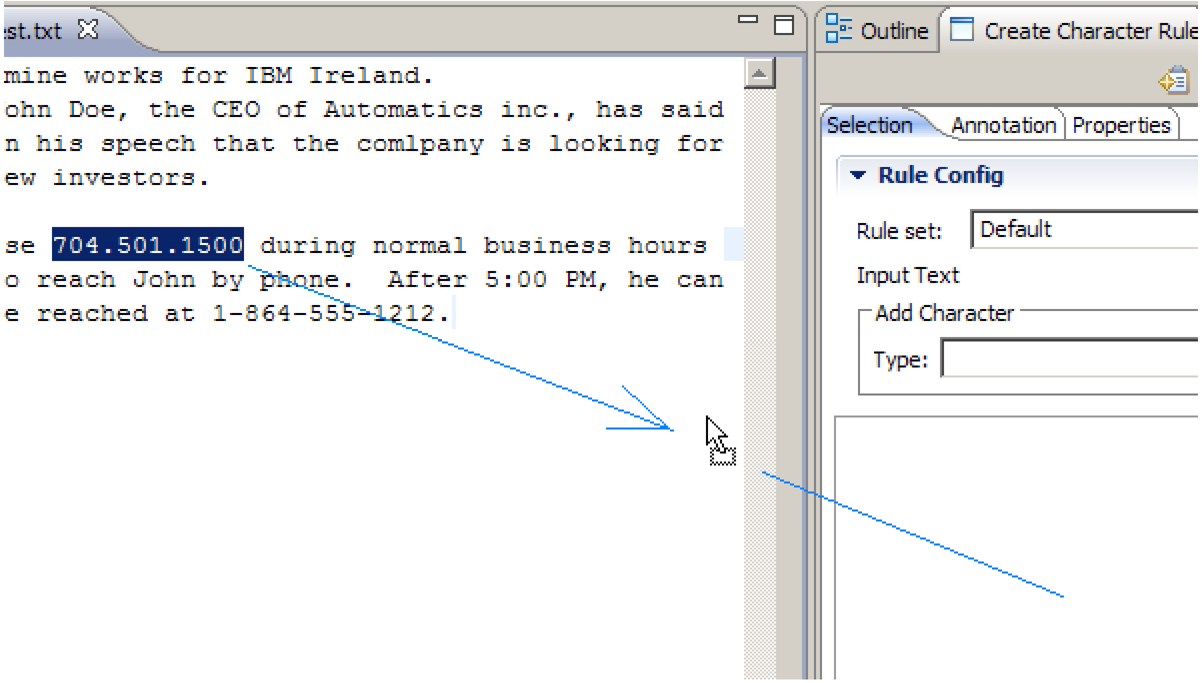


```
Amine works for IBM Ireland.
John Doe, the CEO of Automatics inc., has said
in his speech that the comlpany is looking for
new investors.

Use 704.501.1500 during normal business hours
to reach John by phone.  After 5:00 PM, he can
be reached at 1-864-555-1212.
```

Creating character rules: Defining a character sequence

- Next, drag the selected text onto the Selection tab of the Create Character Rules pane.



```
mine works for IBM Ireland.
ohn Doe, the CEO of Automatics inc., has said
n his speech that the comlpany is looking for
ew investors.

se 704.501.1500 during normal business hours
o reach John by phone.  After 5:00 PM, he can
e reached at 1-864-555-1212.
```

Rule set: Default

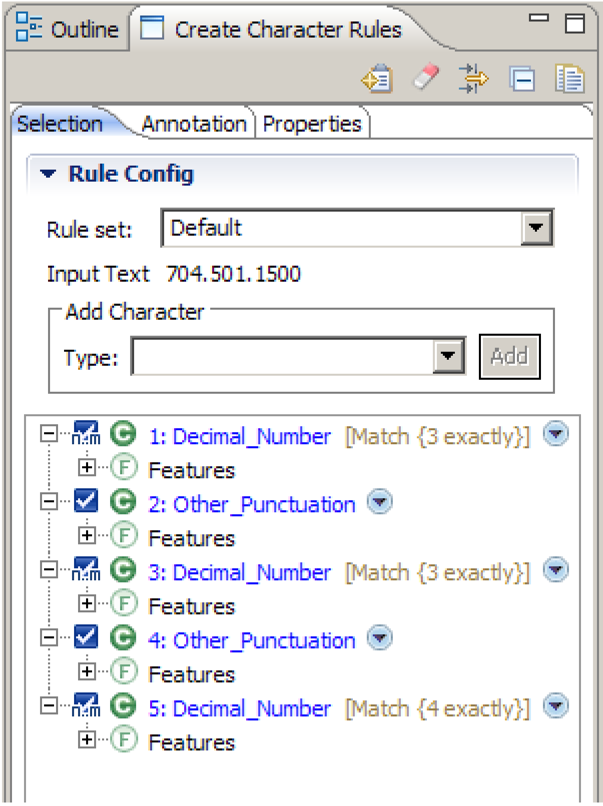
Input Text

Add Character

Type:

Creating character rules: Defining a character sequence

After dropping the selected text onto the Selection tab of the Create Character Rules pane, the text gets parsed into its initial character sequence tree.



Rule set: Default

Input Text 704.501.1500

Type:

- ☑ 1: Decimal_Number [Match {3 exactly}]
  - Features
- ☑ 2: Other_Punctuation
  - Features
- ☑ 3: Decimal_Number [Match {3 exactly}]
  - Features
- ☑ 4: Other_Punctuation
  - Features
- ☑ 5: Decimal_Number [Match {4 exactly}]
  - Features

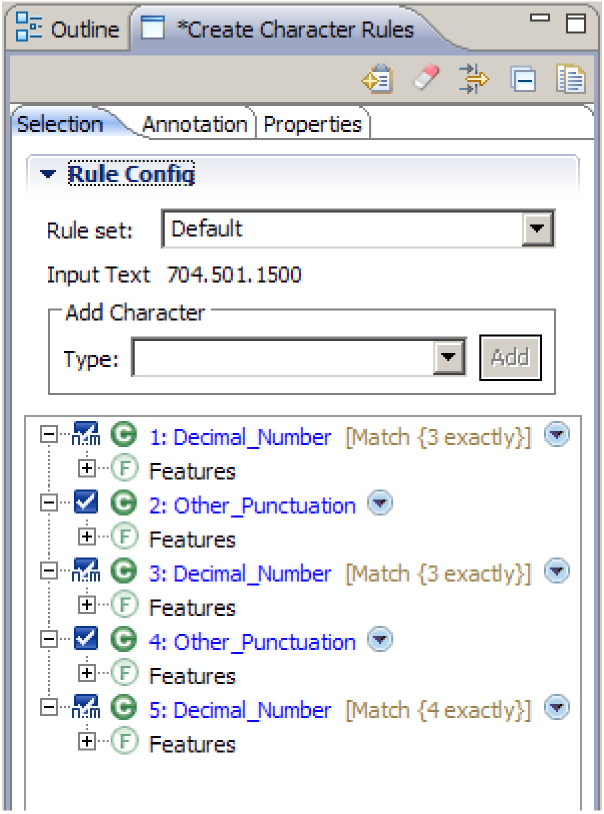
Note that the **Input Text** field displays the text it received during the drag-and-drop event.

Below the Input Text field, the text is represented as a tree of Unicode Character Class nodes that were created as part of the drag-and-drop step. Each element in the tree represents one or more characters that share the same character class traits. Using the editor, these Character nodes can be modified, or removed completely, to meet the needs of the Character Rule. Of course, additional Character Class nodes can be added.

Creating character rules: Defining a character sequence

Before making any modification to your initial character sequence tree, take a closer look at the individual component nodes.

- The first **Decimal_Number** [Match {3 exactly}] node represents the digits 704 in your input text.
- Next you have an **Other_Punctuation** node that represents the first . ("dot" or "period") in the input text. (Characters such as '!', "", '#', '%', and '&' fall into this Unicode Character class designation.)
- The second **Decimal_Number** [Match {3 exactly}] node represents the digits 501 in your input text.
- Again you have an **Other_Punctuation** node to represent the second . in your input text.
- The final node, **Decimal_Number** [Match {4 exactly}], represents the digits 1500 in our input text.



Rule set: Default

Input Text 704.501.1500

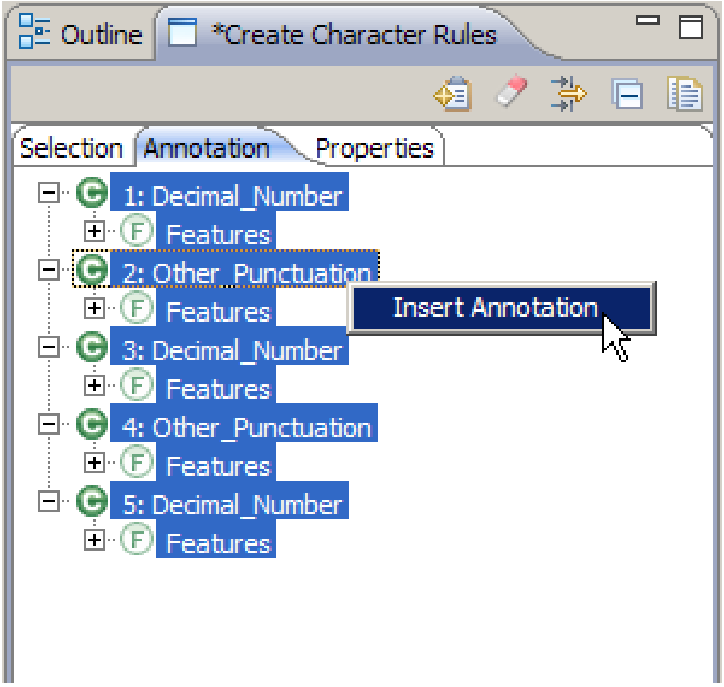
Type:

- 1: Decimal_Number [Match {3 exactly}]
  - Features
- 2: Other_Punctuation
  - Features
- 3: Decimal_Number [Match {3 exactly}]
  - Features
- 4: Other_Punctuation
  - Features
- 5: Decimal_Number [Match {4 exactly}]
  - Features

Creating character rules: Specifying annotations

Leaving the character sequence tree in its initial state, switch over to the annotation tab of the character rules editor. Add a single annotation to the rule.

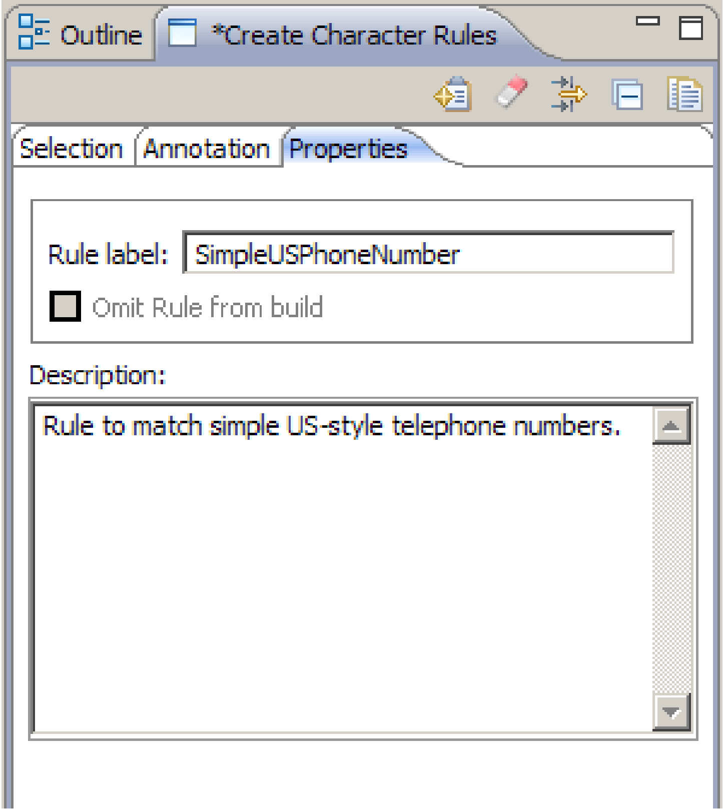
- Right-click any of the character class nodes that are presented in the Annotation tab. All nodes will become selected.
- Choose the **Insert Annotation** menu item.
- When the Insert Annotation dialog appears, add “SimpleUSPhoneNumber” to the default prefix. (You may recognize this prefix as the **UIMA type prefix** value you provided when creating the Character Rules Database earlier.)
- Select **OK**.
- Note that the Character class nodes now appear under the new Annotation node you just created.



Creating character rules: Setting rule properties

A few properties can be assigned to a character rule to help you organize and manage your rules. For example, you can assign a label and text description. You can also disable a character rule by omitting it from the build.

- Enter the value “SimpleUSPhoneNumber” for **Rule label**.
- Enter a **Description** value like “Rule to match simple US-style telephone numbers.”
- Leave the **Omit Rule from build** check box unchecked. You want this character rule included in the database build you will perform on the next slide. (Note, however, this is a very useful option that can be used when testing various character rules being designed. You can see which of the rules are locating matches that best meets your needs without having to delete and re-create rules for each test.)



Rule label: SimpleUSPhoneNumber

□ Omit Rule from build

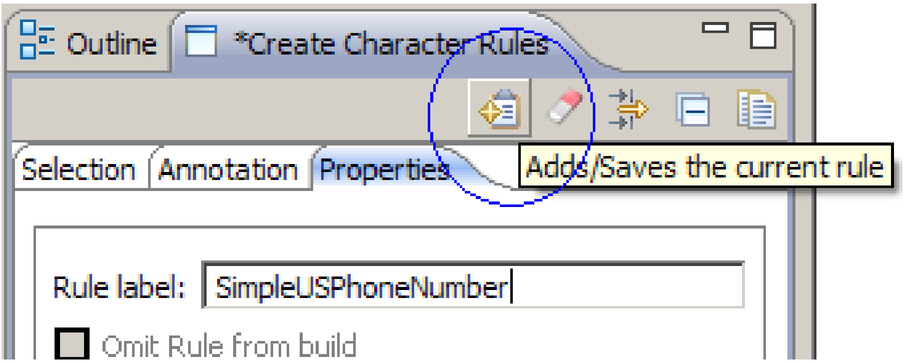
Description:

Rule to match simple US-style telephone numbers.

Creating character rules: Saving and building

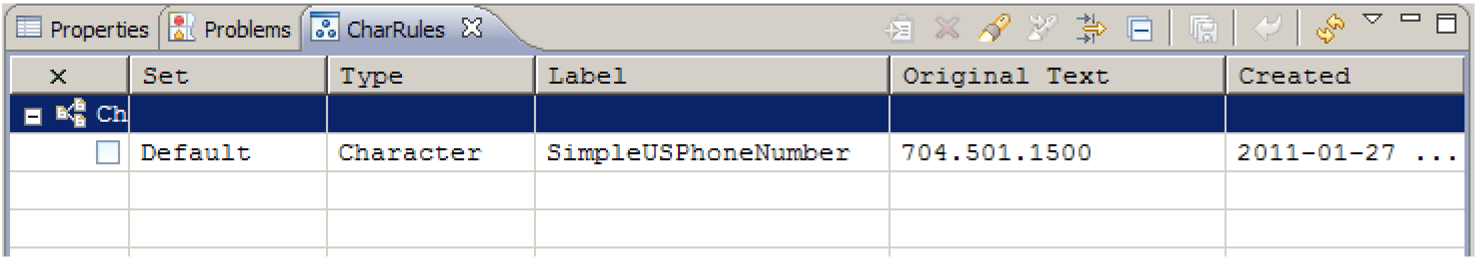
Before a character rule can be built and tested, it needs to be saved to the character rule database.

- The leftmost button in the character rules editor is the **Add/Save** button. Select this button to save your new character rule.
- After the save completes, you can find your new character rule displayed in the database editor view.



Rule label: SimpleUSPhoneNumber

□ Omit Rule from build

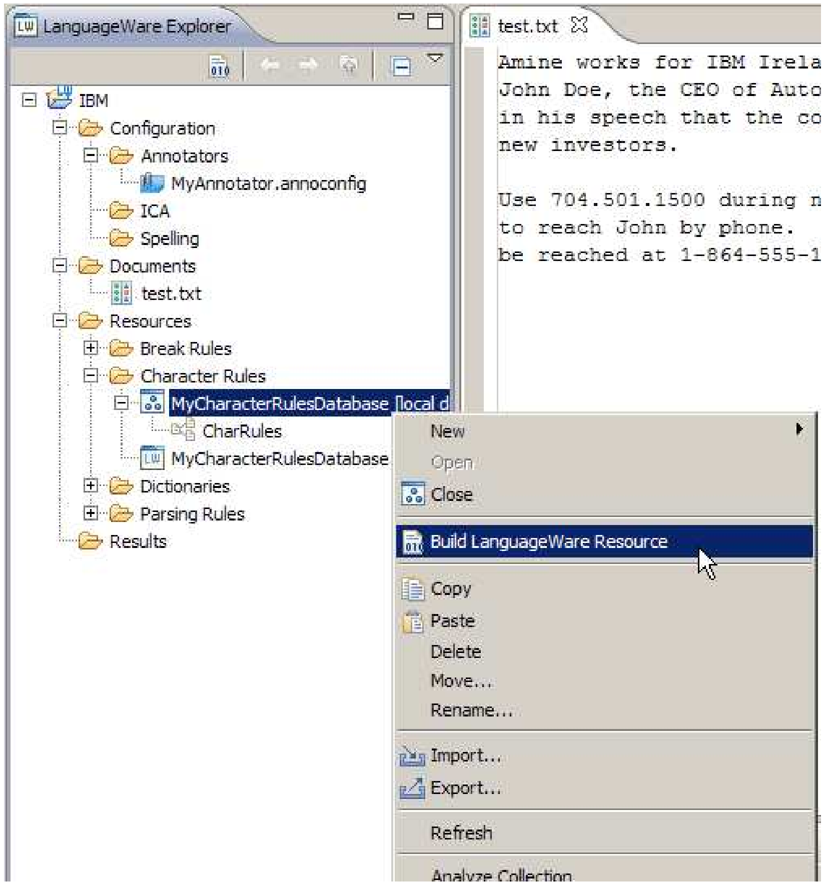


x	Set	Type	Label	Original Text	Created
⊟ Ch					
□	Default	Character	SimpleUSPhoneNumber	704.501.1500	2011-01-27 ...

Creating character rules: Saving and building

To verify that your new character rule works as desired, the character rule database needs to be built.

- Right-click the character rules database you are working on in the LanguageWare explorer pane.
- Choose **Build LanguageWare Resource** to compile and build the database.
- After the progress information dialog displays, your character rules database is ready for use. (in fact, after the LanguageWare Resource Workbench detects an update to an active pipeline resource, it will automatically re-analyze the text document for you.)



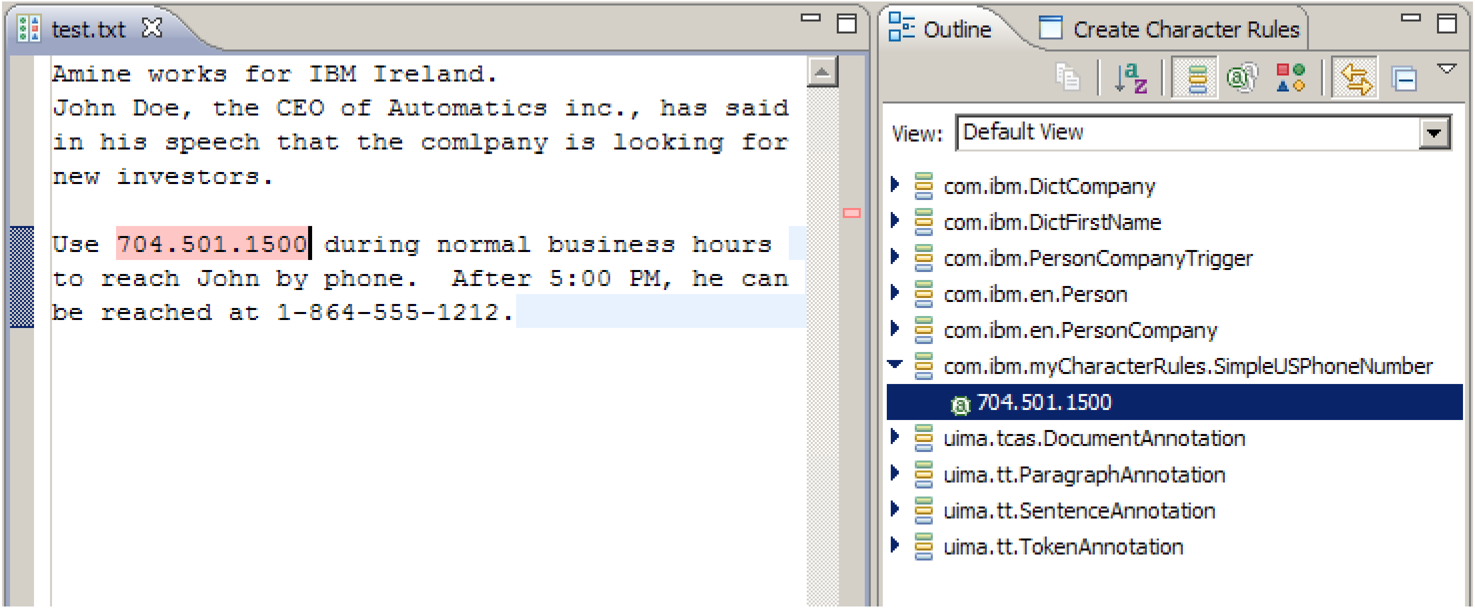
```
Amine works for IBM Irela
John Doe, the CEO of Auto
in his speech that the co
new investors.

Use 704.501.1500 during n
to reach John by phone.
be reached at 1-864-555-1
```

Creating character rules: Saving and building

After the character rules database has been built, and the automatic analysis of the text document has completed, your new character rule will post an annotation called **com.ibm.myCharacterRules.SimpleUSPhoneNumber** in the Outline tab. It should be no surprise that it found a match on the 704.501.1500 telephone number.

Next you will modify your Character Rule to match on other telephone number formats such as the 1-864-555-1212 example in the text file.



```
Amine works for IBM Ireland.
John Doe, the CEO of Automatics inc., has said
in his speech that the comlpany is looking for
new investors.

Use 704.501.1500 during normal business hours
to reach John by phone.  After 5:00 PM, he can
be reached at 1-864-555-1212.
```

View: Default View

- com.ibm.DictCompany
- com.ibm.DictFirstName
- com.ibm.PersonCompanyTrigger
- com.ibm.en.Person
- com.ibm.en.PersonCompany
- com.ibm.myCharacterRules.SimpleUSPhoneNumber
  - 704.501.1500
- uima.tcas.DocumentAnnotation
- uima.tt.ParagraphAnnotation
- uima.tt.SentenceAnnotation
- uima.tt.TokenAnnotation

Module roadmap

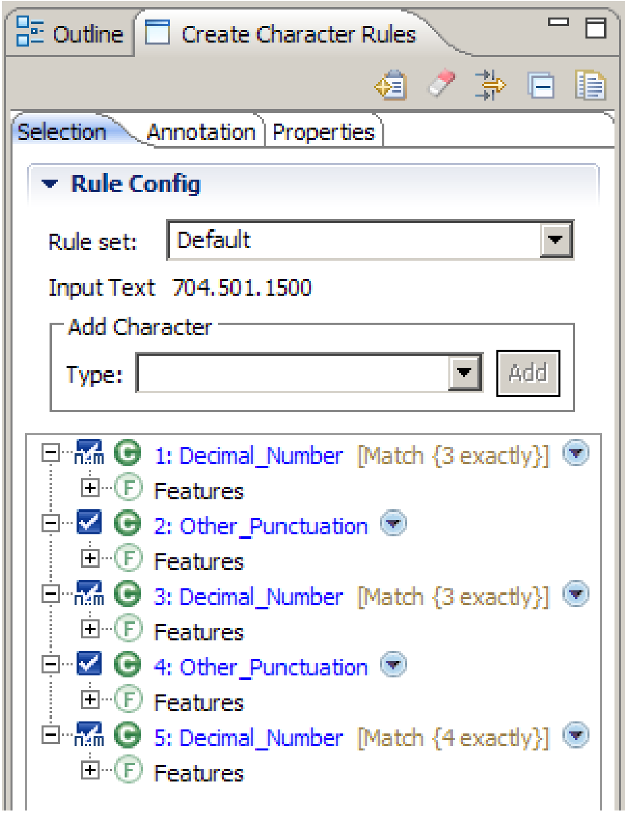
- **Creating character rules**
  - What are character rules?
  - Creating a character rules database
  - Create and add character rules to the character rules database
  - Modifying a character rule to expand match scope
- **Summary and best practices**
- **Sample exercises**

Modifying a character rule

Often a character rule will need to be modified slightly in order to match variations of the original character sequence used to create the rule. You will now modify your character rule to match US-style telephone numbers like 1-864-555-1212.

- If you are still viewing the Annotations on the Outline tab, switch over to the Create Character Rules tab to get back to the Character Rules Editor.

- The modifications you will make to your Rule:
  - add a Decimal_Number node,
  - add a Dash_Punctuation node,
  - group the two new nodes together,
  - move the Group to the top of the tree, and
  - replace the Other_Punctuation nodes with Any Character nodes.



Rule set: Default

Input Text 704.501.1500

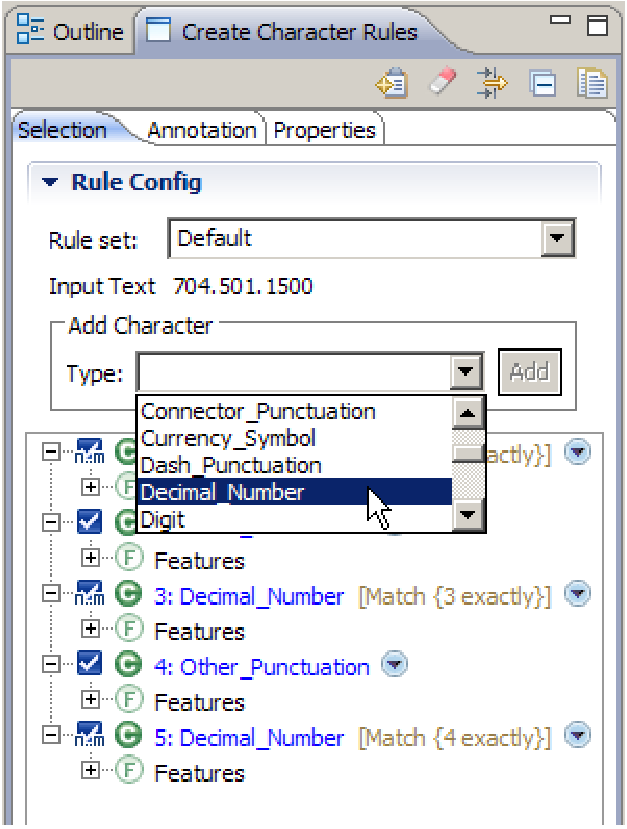
Type:

1: Decimal_Number [Match {3 exactly}]
Features
2: Other_Punctuation
Features
3: Decimal_Number [Match {3 exactly}]
Features
4: Other_Punctuation
Features
5: Decimal_Number [Match {4 exactly}]
Features

Modifying a character rule

Adding a Decimal_Number node:

- Using the Add Character, Type drop-down box, select **Decimal_Number** and then **Add**.
- A new Decimal_Number node appears at the bottom of your character sequence tree.
- (You can collapse the Features sub-node for readability. You will not need it for this exercise.)



Rule set: Default

Input Text 704.501.1500

ctly}]

Features

3: Decimal_Number [Match {3 exactly}]

Features

4: Other_Punctuation

Features

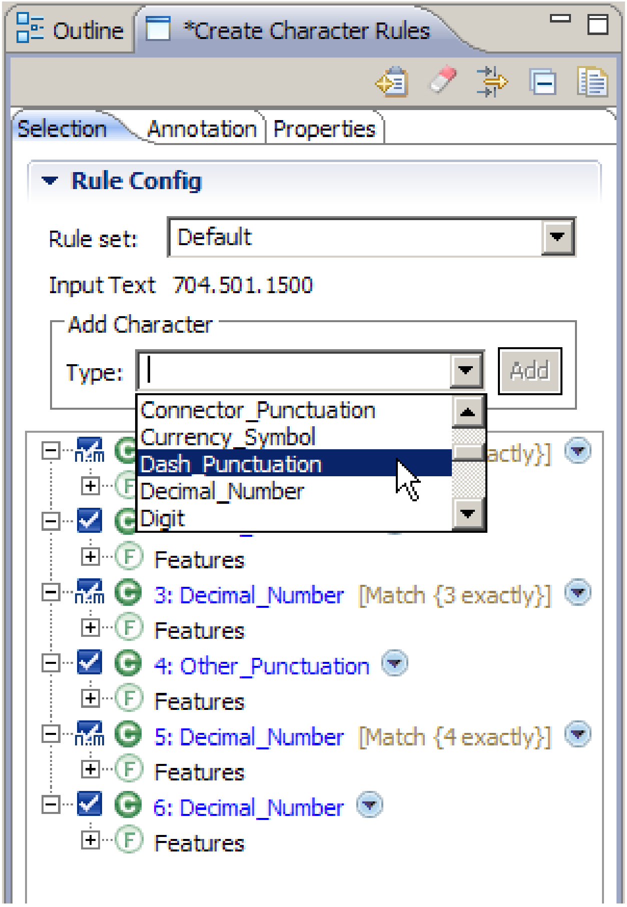
5: Decimal_Number [Match {4 exactly}]

Features

Modifying a character rule

Adding a Dash_Punctuation node:

- Using the Add Character, Type drop-down box, select **Dash_Punctuation** and then **Add**.
- A new Dash_Punctuation node appears at the bottom of your character sequence tree.
- (You can collapse the Features sub-node for readability. You will not need it for this exercise.)



Rule set: Default

Input Text 704.501.1500

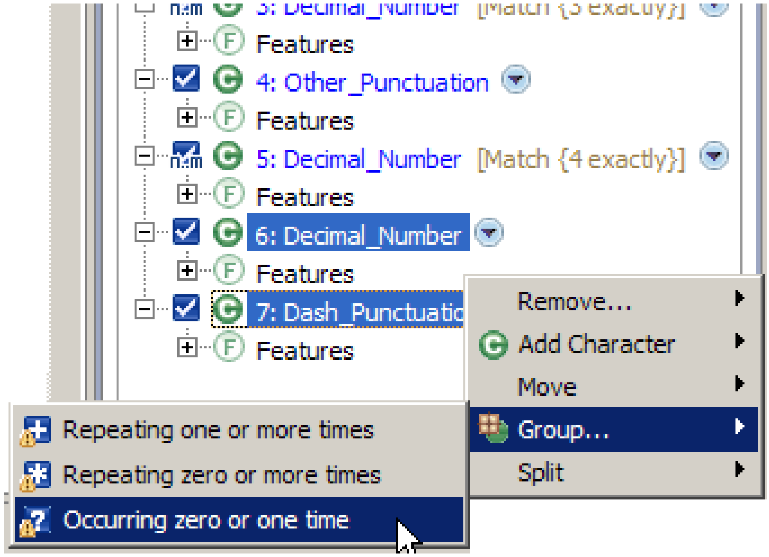
Modifying a character rule

Grouping the Decimal_Number and Dash_Punctuation nodes together:

- Select the two new nodes you just created, the 6: Decimal_Number and 7: Dash_Punctuation nodes.
- Right-click and choose **Group > Occurring zero or one time**.

The Group you have just created will be used to match the "1-" portion that often appears at the beginning of US-style telephone numbers.

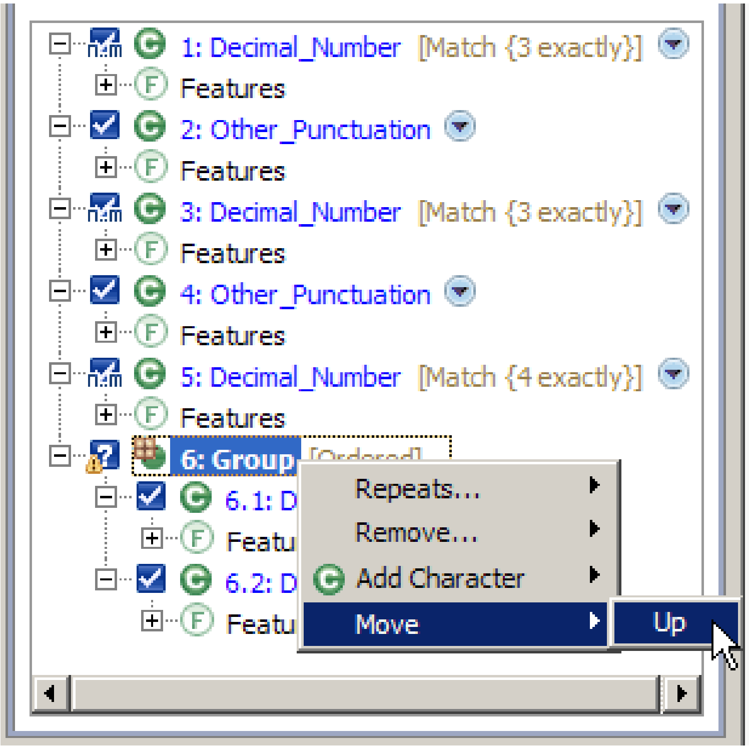
Grouping these two nodes together indicates that your character rule should only consider the pair of characters together. Additionally, you only want to match one of these groups if present—not more than one pair.


3: Decimal_Number [Match {3 exactly}]
Features
4: Other_Punctuation
Features
5: Decimal_Number [Match {4 exactly}]
Features
6: Decimal_Number
Features
7: Dash_Punctuatio
Features
Remove...
Add Character
Move
Group...
Split
Repeating one or more times
Repeating zero or more times
Occurring zero or one time

Modifying a character rule

Moving the Group Up to the top of the Character sequence tree:

- Right-click the Group node you just created and choose **Move > Up**..
- Perform this Move > Up operation 6 times to get the Group node to the top of the Character sequence tree.

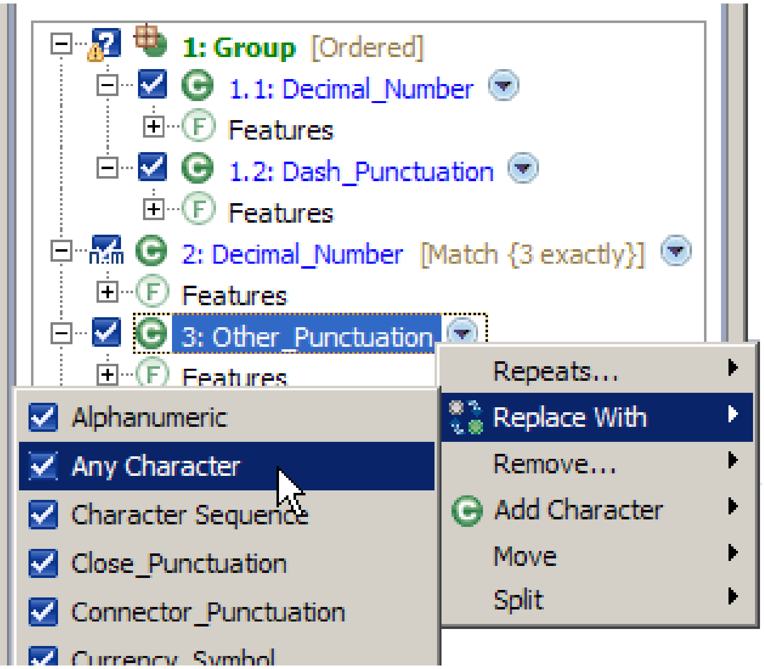


Modifying a character rule

Replacing the Other_Punctuation nodes: Since you want your character rule to match the telephone number 1-864-555-1212, you need your rule to consider the two “-” (dash) punctuation characters used after the two sets of three decimal numbers.

Remember you also want your Rule to continue matching when the “.” (“dot” or “period”) is used.

For this exercise, you will use the “Any Character” character class to allow your rule to match any type of character used to separate the digits in a number.



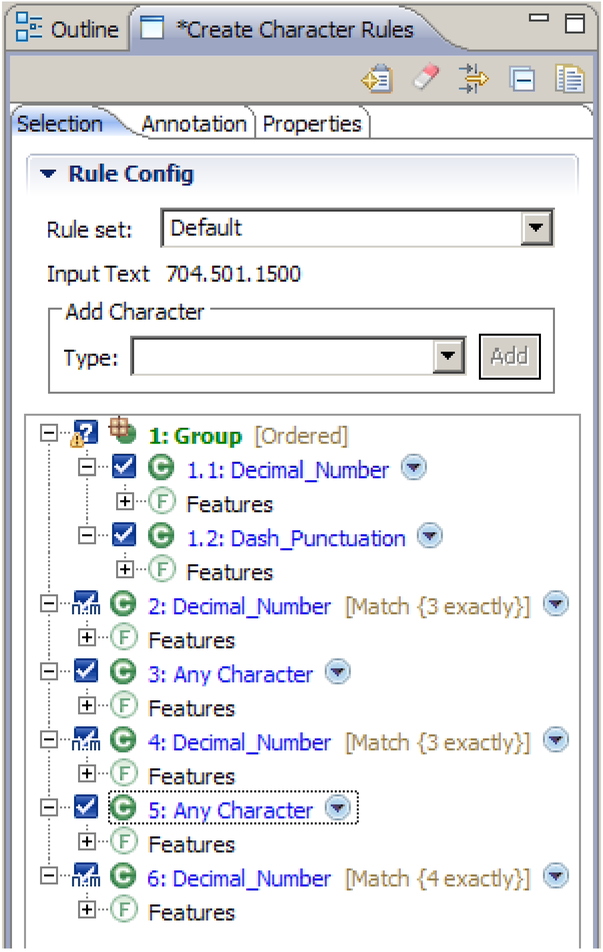
- Right-click 3: Other_Punctuation and choose **Replace With > Any Character**.
- Right-click on 5: Other_Punctuation and choose **Replace With > Any Character**.
- (You can collapse the Features sub-node for readability. You will not need it for this exercise.)

Modifying a character rule

Save and build updated character rule:

After replacing the Other_Punctuation nodes with Any Character nodes, you have completed the modifications for this exercise.

- Save the modifications made to the Character Rule again using the leftmost button in the Character Rules Editor, the **Add/Save** button.
- Right-click the Character Rules Database in the LanguageWare Explorer pane.
- Choose **Build LanguageWare Resource** to compile/build the Database. The Progress Information dialog will appear momentarily.



Rule set: Default

Input Text 704.501.1500

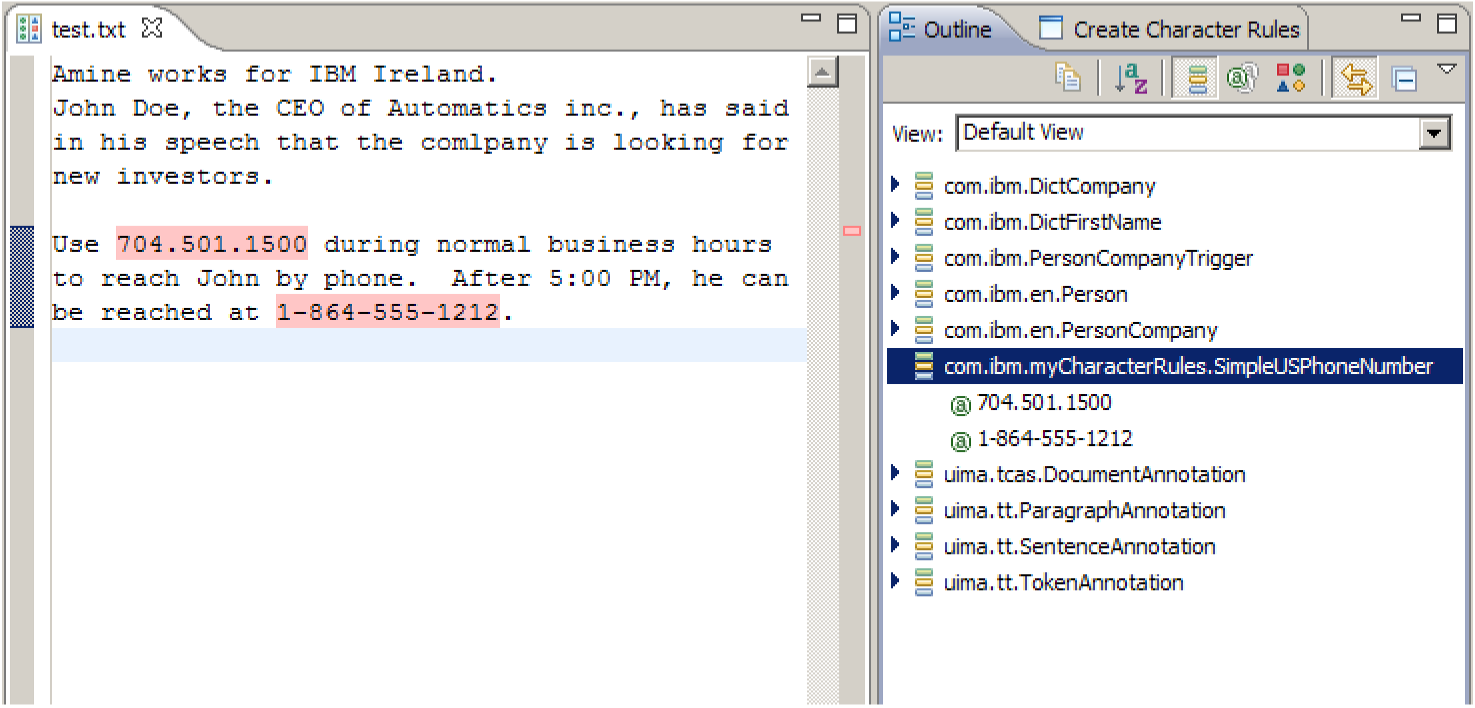
Type:

1: Group [Ordered]
1.1: Decimal_Number
Features
1.2: Dash_Punctuation
Features
2: Decimal_Number [Match {3 exactly}]
Features
3: Any Character
Features
4: Decimal_Number [Match {3 exactly}]
Features
5: Any Character
Features
6: Decimal_Number [Match {4 exactly}]
Features

Modifying a character rule

After the character rules database has been rebuilt, and the automatic analysis of the text document has completed, your modified character rule will post an additional annotation in the Outline tab.

Both the 704.501.1500 and 1-864-555-1212 telephone numbers were annotated.



```
Amine works for IBM Ireland.
John Doe, the CEO of Automatics inc., has said
in his speech that the comlpany is looking for
new investors.

Use 704.501.1500 during normal business hours
to reach John by phone.  After 5:00 PM, he can
be reached at 1-864-555-1212.
```

- com.ibm.DictCompany
- com.ibm.DictFirstName
- com.ibm.PersonCompanyTrigger
- com.ibm.en.Person
- com.ibm.en.PersonCompany
- com.ibm.myCharacterRules.SimpleUSPhoneNumber
  - 704.501.1500
  - 1-864-555-1212
- uima.tcas.DocumentAnnotation
- uima.tt.ParagraphAnnotation
- uima.tt.SentenceAnnotation
- uima.tt.TokenAnnotation

Course roadmap

- **Creating character rules**
  - What are character rules?
  - Creating a character rules database
  - Create and add character rules to the character rules database
  - Modifying a character rule to expand match scope
- **Summary and best practices**
- **Sample exercises**

Module summary

You have completed this module and

- you understand what character rules are and when to use them,
- can create a character rules database,
- can add character rules to the character rules database,
- can modify a character rule to expand the scope of the rule.

See the LanguageWare help for more tips and advanced use cases.

Best practices

- When naming databases, it is good practice to use explicit, self-documenting names indicating the types of rules contained in the database.
- Be certain to group sets of like character rules in databases where they make sense to help with maintenance of the rules themselves. If you believe a particular rule, or group of rules, belong in a separate database, create a new database to contain them.
- It is easier to manage or maintain groups of simple character rules. If a rule's character sequence tree is becoming too complex to manipulate, consider creating a new character rule in the same database to handle alternate cases. There are no problems having multiple rules creating annotations with the same annotation name.
- Take advantage of the “Omit Rule from build” option on the Properties tab of the Character Rules Editor. This is very handy when testing various forms of a rule to see which best satisfies your needs.

Module roadmap

- **Creating character rules**

  What are character rules?

  Creating a character rules database

  Create and add character rules to the character rules database

  Modifying a character rule to expand match scope
- **Summary and best practices**
- **Sample exercises**

Practice exercise

- Create a character rules database named “Highways”.
- Add the character rules dictionary file to the “AnalyseHelpline” annotator.
- Create a character rule that identifies US Interstate Highways like “I-123”.
- Build your new highways character rule database and check that the annotations show in the outline and in the text.

Contacts

- If you have any questions, comments or suggestions, contact us using the LanguageWare email address *EMEALAN@ie.ibm.com* or on the developerWorks® forum.

Trademarks, copyrights, and disclaimers

IBM, the IBM logo, ibm.com, developerWorks, and LanguageWare are trademarks or registered trademarks of International Business Machines Corp., registered in many jurisdictions worldwide. Other product and service names might be trademarks of IBM or other companies. A current list of other IBM trademarks is available on the web at "Copyright and trademark information" at http://www.ibm.com/legal/copytrade.shtml

Other company, product, or service names may be trademarks or service marks of others.

THE INFORMATION CONTAINED IN THIS PRESENTATION IS PROVIDED FOR INFORMATIONAL PURPOSES ONLY. WHILE EFFORTS WERE MADE TO VERIFY THE COMPLETENESS AND ACCURACY OF THE INFORMATION CONTAINED IN THIS PRESENTATION, IT IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EXPRESS OR IMPLIED. IN ADDITION, THIS INFORMATION IS BASED ON IBM'S CURRENT PRODUCT PLANS AND STRATEGY, WHICH ARE SUBJECT TO CHANGE BY IBM WITHOUT NOTICE. IBM SHALL NOT BE RESPONSIBLE FOR ANY DAMAGES ARISING OUT OF THE USE OF, OR OTHERWISE RELATED TO, THIS PRESENTATION OR ANY OTHER DOCUMENTATION. NOTHING CONTAINED IN THIS PRESENTATION IS INTENDED TO, NOR SHALL HAVE THE EFFECT OF, CREATING ANY WARRANTIES OR REPRESENTATIONS FROM IBM (OR ITS SUPPLIERS OR LICENSORS), OR ALTERING THE TERMS AND CONDITIONS OF ANY AGREEMENT OR LICENSE GOVERNING THE USE OF IBM PRODUCTS OR SOFTWARE.