Programming for IBM InfoSphere Streams V4 with SPL

Information

Length:	4.0 Days
__Ref:	DW724G-X
Delivery method:	Classroom
Price:	EUR

Overview

This course is designed to teach you about the Streams Processing Language. It will begin with the basic concepts of InfoSphere Streams and the basic Streams Processing Language operators used in a Streams program. You will learn how to access data from an external source using the Source type operators and write an output stream using the Sink type operators.

You will then learn how and when to use the various Stream operators, like the Functor, Punctor, Aggregation, Sort, Join, Split, Barrier, Delay, and Switch operators. Lab exercises will use the InfoSphere Streams IDE that is based upon Eclipse as the development and testing environment, but you will get the opportunity to invoke the compilation of a Streams program from the command line as well. In the labs you will be given the choice to develop the applications using the SPL Graphical Editor, introduced in Version 3, that allows drag and drop or the original SPL Editor that is text based.

The second half of the course shows how to control the placement of processing elements and the debugging capabilities of the Streams Processing Language. You will learn about consistent regions and how to use them to process tuples at-least-once.

You will be introduced to the various toolkits supplied with InfoSphere Streams and work with data mining and database toolkits in a lab.

Finally, you will be is shown how to extend the Streams Processing Language through the development of user-defined functions and both generic and non-generic primitive operators. Both C++ and Java non-generic primitive operators are covered.

Public

This basic course is designed for those who are planning on developing InfoSphere Streams applications.

Prerequisits

There are no prerequisites are required; however, use of an Eclipse-based tool would be beneficial as well as exposure to the C++, Java, and Perl languages.

Objective

- Explain how operators observe data on streams to create other streams
- Define the format for both the built-in Source and a Sink edge adapter operators
- List the types of URIs supported by Source and Sink operators
- Explain the use of sliding and tumbling windows in the Streams Processing Language
- Describe how to control the timing of tuples using the Delay operator
- Explain the use of the following operators: Functor, Punctor, Split, Join, Aggregation, Sort, Barrier, and Delay
- Explain the preprocessing capabilities of the Streams Processing Language and how those capabilities are used to generate Streams source code
- Explain how Streams uses Consistent Regions to ensure at-least-once processing
- Describe how to use the Streams debugging capabilities
- Explain how to control the placement of operators onto processing specific nodes and how to fuse operators into specific processing elements
- List the toolkits supplied with InfoSphere Streams
- Explain how to debug a Streams application
- Describe how to create a user-defined function
- List the steps necessary to create an SPL non-generic primitive operator written in C++
- Explain how to create an SPL non-generic primitive operator written in Java
- Describe how to create an SPL generic primitive operator

Topics

- Unit 1 - InfoSphere Streams Overview
- Unit 2 - Streams Processing Language Basics
- Unit 3 - Streams Processing Language Development
- Unit 4 - SPL Programming Introduction
- Unit 5 - Adapter Operators
- Unit 6 - Relational and Utility Operators: The Journey Begins
- Unit 7 - Windowing and Joins
- Unit 8 - Punctuation, Aggregation and Sorting
- Unit 9 - Timing and Coordination
- Unit 10 - Lists, Sets, and Maps
- Unit 11 - Consistent Regions
- Unit 12 - Resources, Partitions, and Other Configs
- Unit 13 - Debugging
- Unit 14 - Toolkits
- Unit 15 - SPL Functions
- Unit 16 - SPL C++ Non-generic Primitive Operators
- Unit 17 - SPL Java Non-generic Primitive Operators

- Unit 18 - SPL Generic Primitive Operators