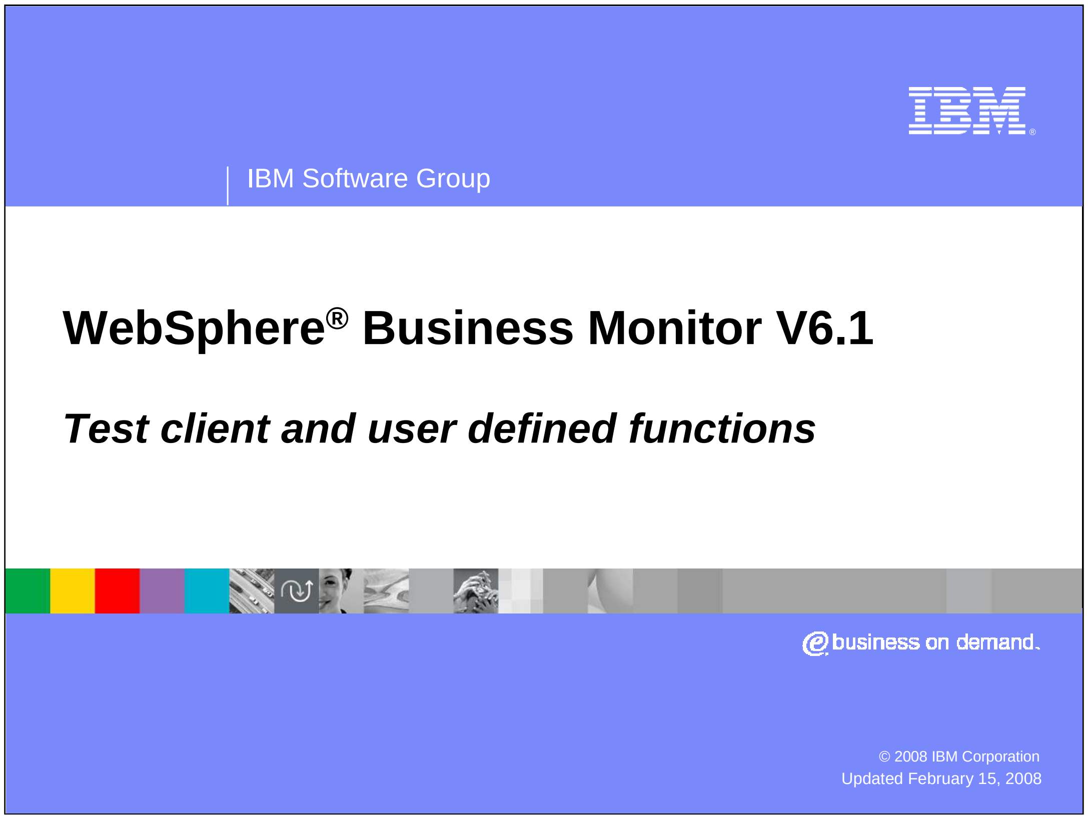


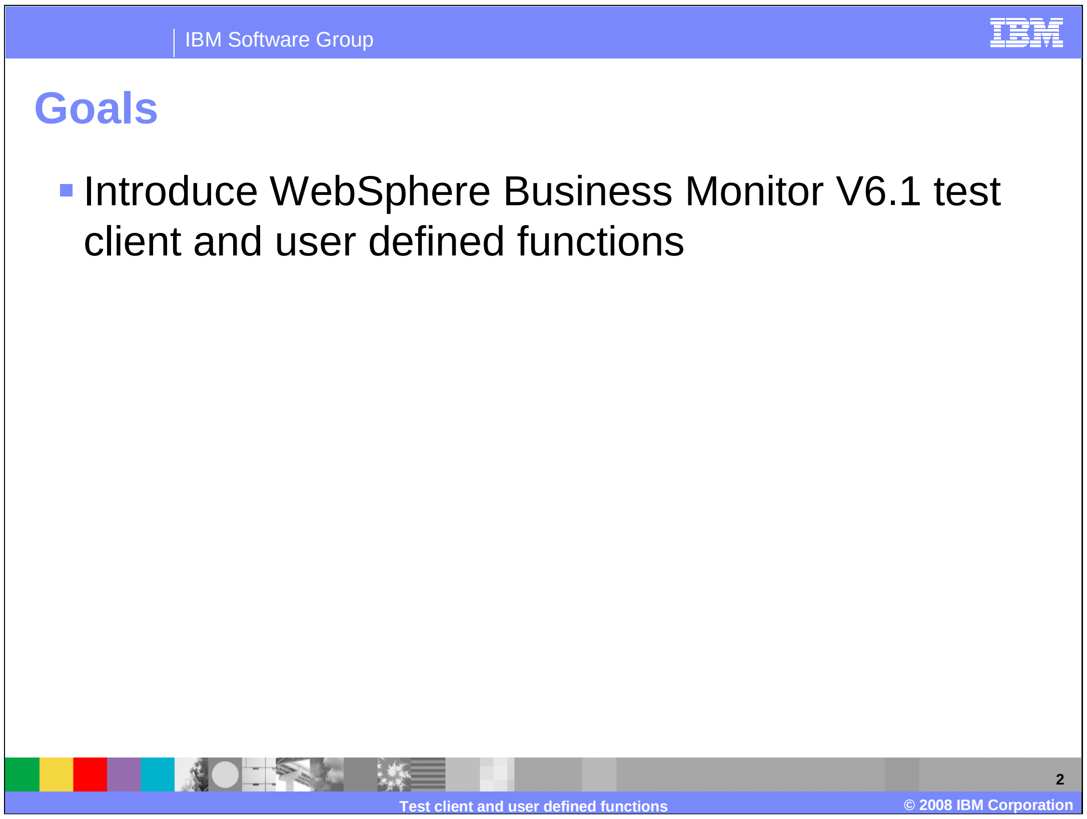
IBM Software Group

WebSphere® Business Monitor V6.1

Test client and user defined functions

Updated February 15, 2008

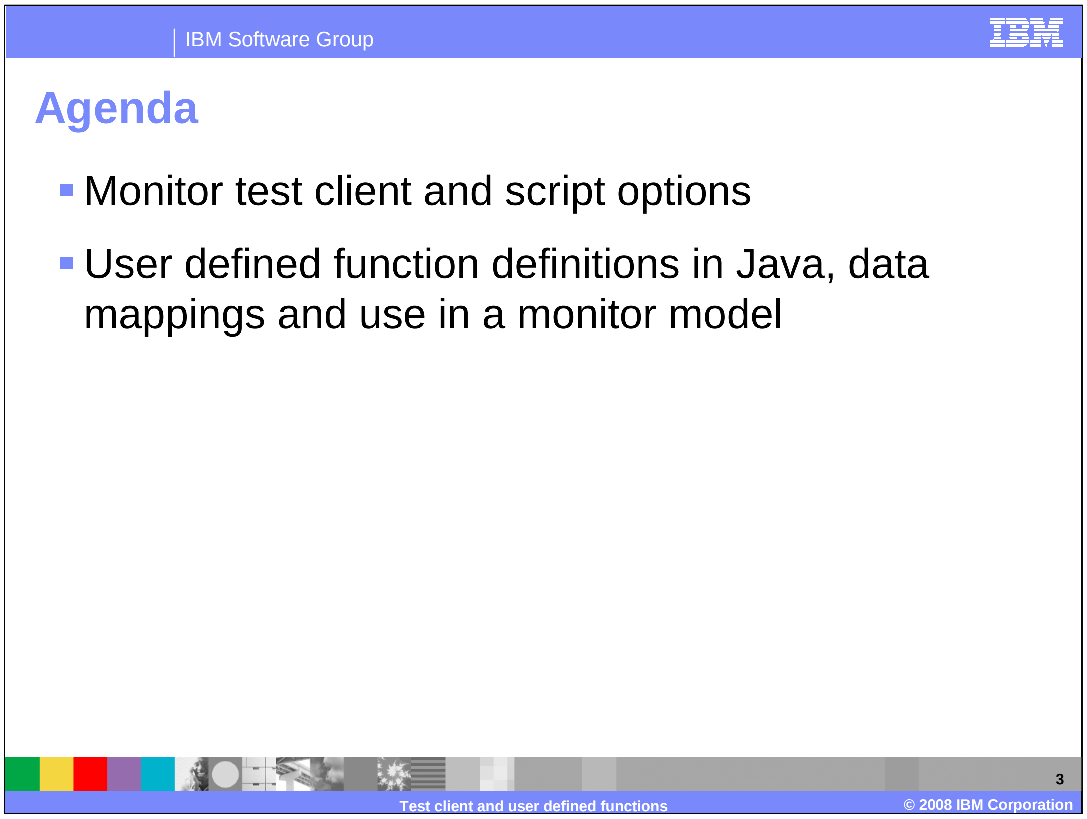
This presentation introduces using the test client and user defined functions in WebSphere Business Monitor Version 6.1.



Goals

- Introduce WebSphere Business Monitor V6.1 test client and user defined functions

This presentation should give you a good understanding of the Monitor support for the test client and user defined functions.


IBM Software Group

Agenda

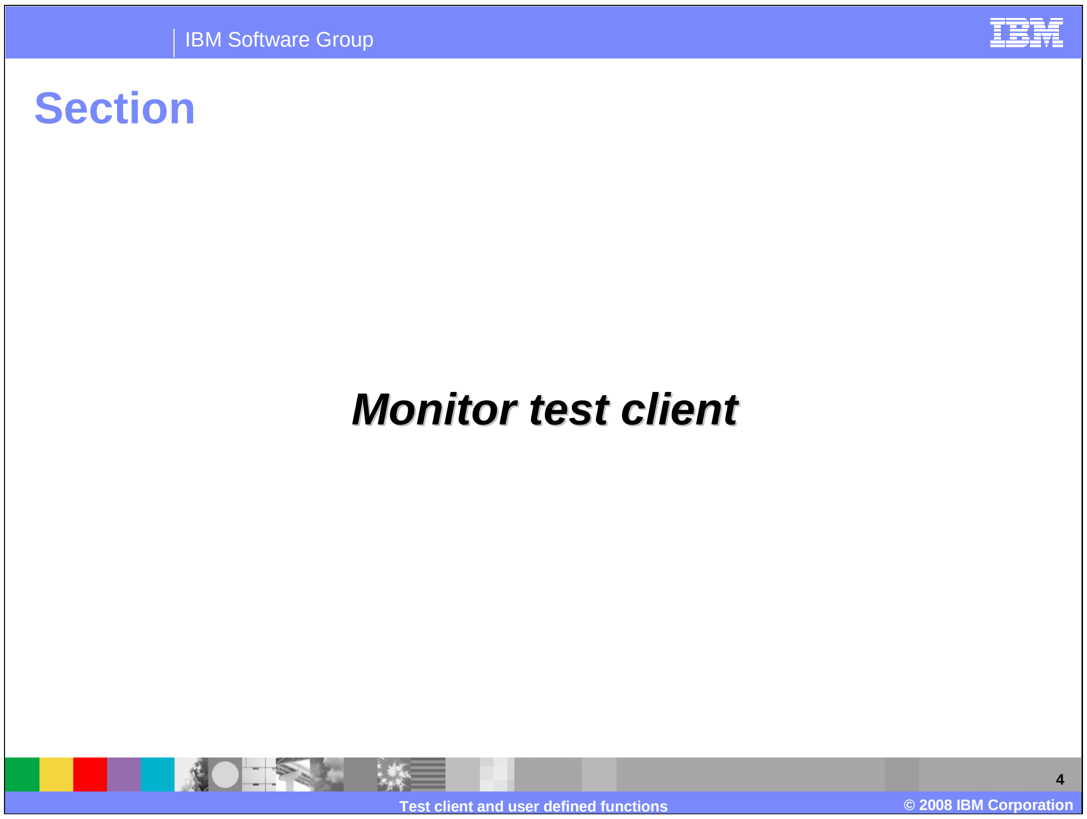
- Monitor test client and script options
- User defined function definitions in Java, data mappings and use in a monitor model

3

Test client and user defined functions

© 2008 IBM Corporation

You will see the monitor test client and some of the options to use when creating test scripts. You will also review user defined functions definitions in Java, how to map Java data to XPath data and how to use a user defined function in a monitor model.



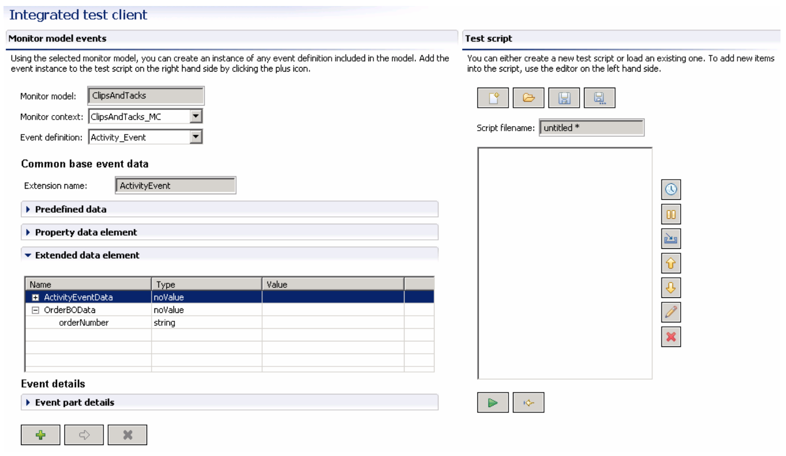
Section

Monitor test client

This section will delve into the details of the Monitor test client.

Integrated test client

- Create/emit events using the integrated test client
- You can create scripts containing the events that you create


Integrated test client

Monitor model events

Using the selected monitor model, you can create an instance of any event definition included in the model. Add the event instance to the test script on the right hand side by clicking the plus icon.

Monitor model: ClipsAndTacks
Monitor context: ClipsAndTacks_MC
Event definition: Activity_Event

Common base event data

Extension name: ActivityEvent

Predefined data
Property data element
Extended data element

Name	Type	Value
ActivityEventData	noValue	
OrderBOData	noValue	
orderNumber	string	

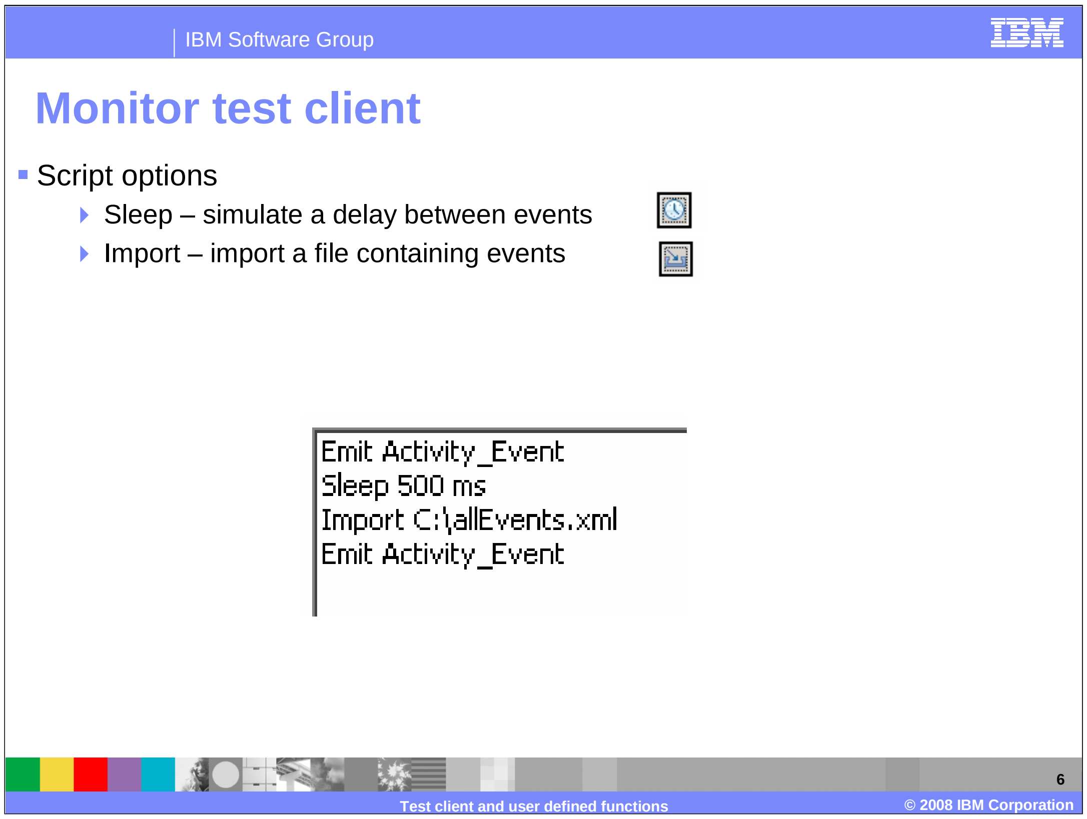
Event details
Event part details

Test script

You can either create a new test script or load an existing one. To add new items into the script, use the editor on the left hand side.

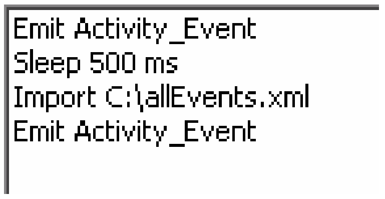
Script filename: untitled *

There is an integrated test client which can be beneficial in the development environment. In the project explorer, you can launch the test client, then you can specify the event definitions that you want to work with. For each event definition, you can supply values for each field in the event, and then you can save a script that saves the events with the values along with the ordering of the events. So you can easily create a test sequence of events, save them and re-run them at a later time.

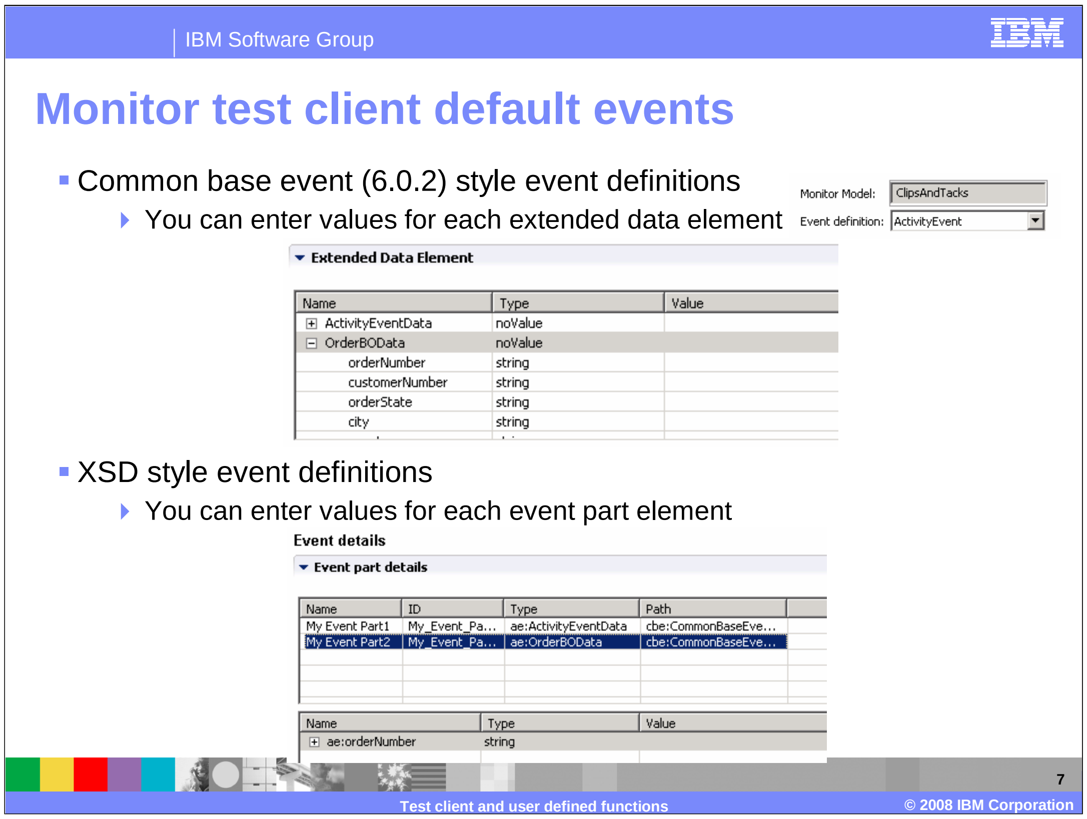


Monitor test client

- Script options
  - Sleep – simulate a delay between events
  - Import – import a file containing events


```
Emit Activity_Event
Sleep 500 ms
Import C:\allEvents.xml
Emit Activity_Event
```

There are two options which are available to your test script. The sleep option allows you to simulate a delay between event submissions. You are prompted for the number of milliseconds for the delay. The import option allows you to specify a path to an XML file where your events are defined.



IBM Software Group

Monitor test client default events

- Common base event (6.0.2) style event definitions
  - You can enter values for each extended data element

Monitor Model: ClipsAndTacks

Event definition: ActivityEvent

Extended Data Element

Name	Type	Value
⊞ ActivityEventData	noValue	
⊟ OrderBOData	noValue	
orderNumber	string	
customerNumber	string	
orderState	string	
city	string	

- XSD style event definitions
  - You can enter values for each event part element

Event details

Event part details

Name	ID	Type	Path
My Event Part1	My_Event_Pa...	ae:ActivityEventData	cbe:CommonBaseEve...
My Event Part2	My_Event_Pa...	ae:OrderBOData	cbe:CommonBaseEve...

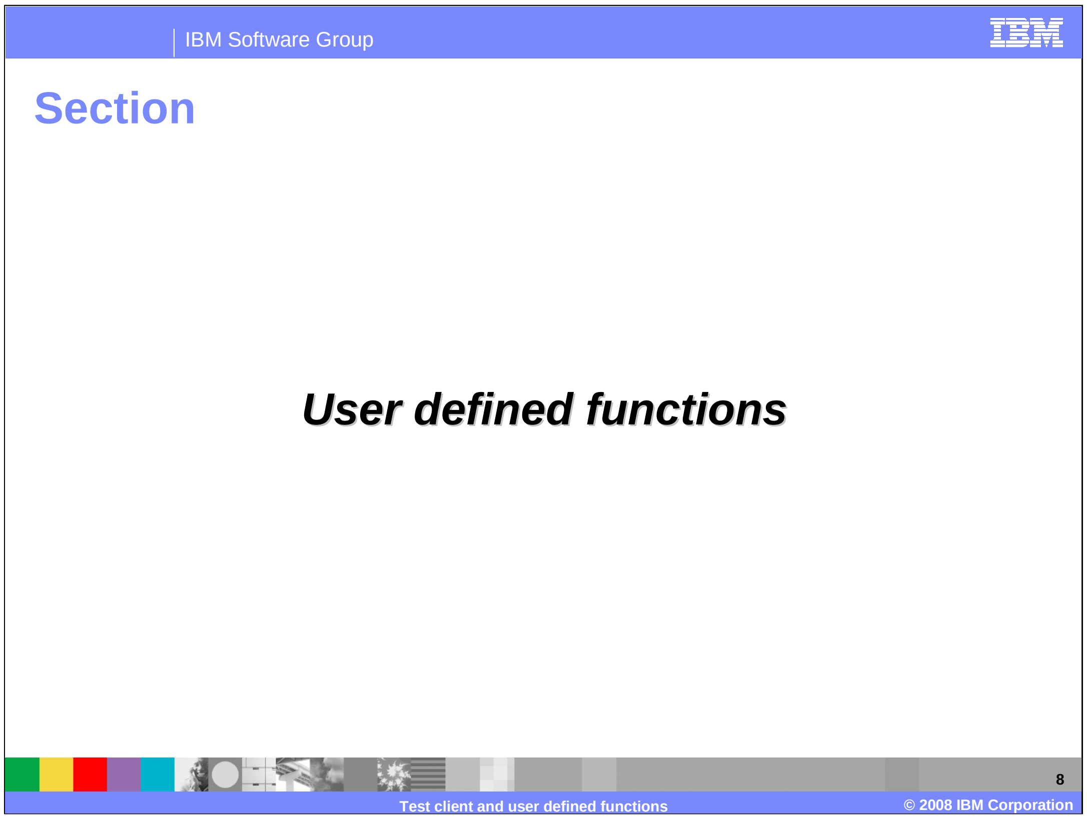
Name	Type	Value
⊞ ae:orderNumber	string	

7

Test client and user defined functions

© 2008 IBM Corporation

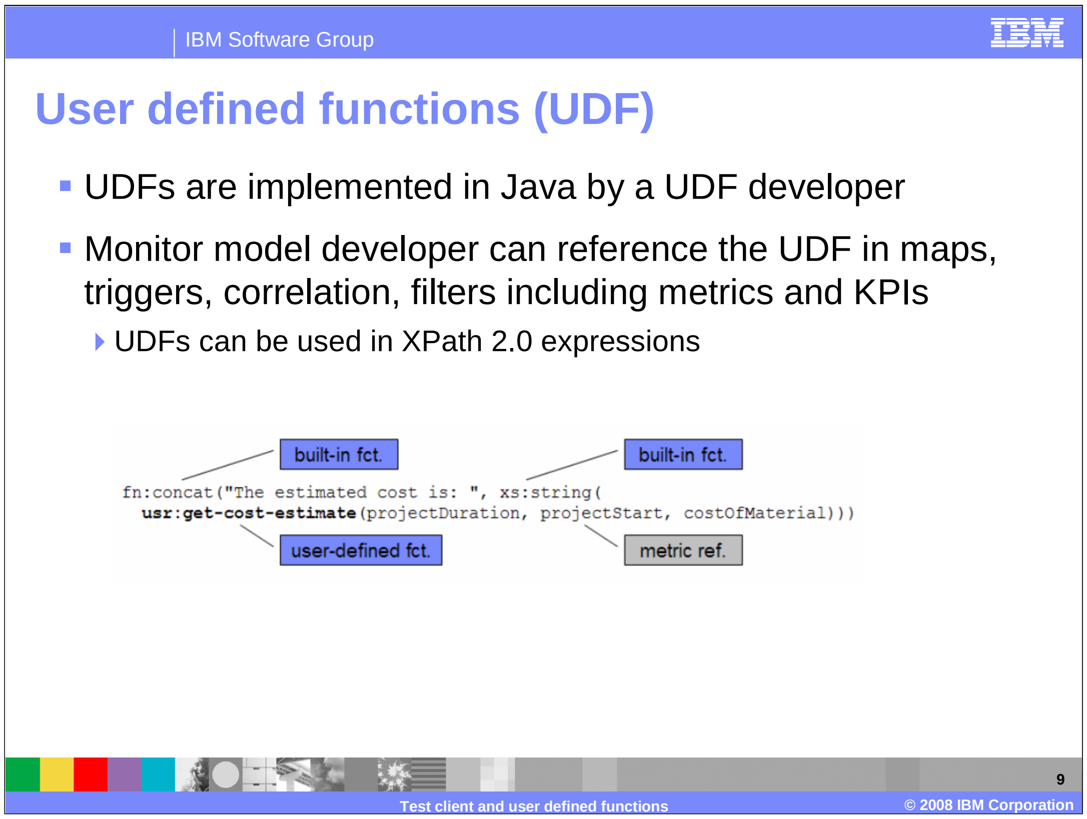
In the event definitions drop down, you will also see the event definitions for all event definitions in the model. For common base event version 6.0.2 style events, you can enter values for data elements in the section for extended data elements. For XSD style events, you can enter values for each event part element.



Section

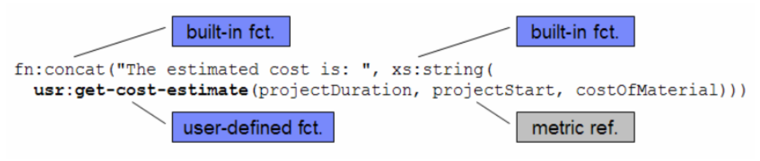
User defined functions

This section will delve into the details of user defined functions.



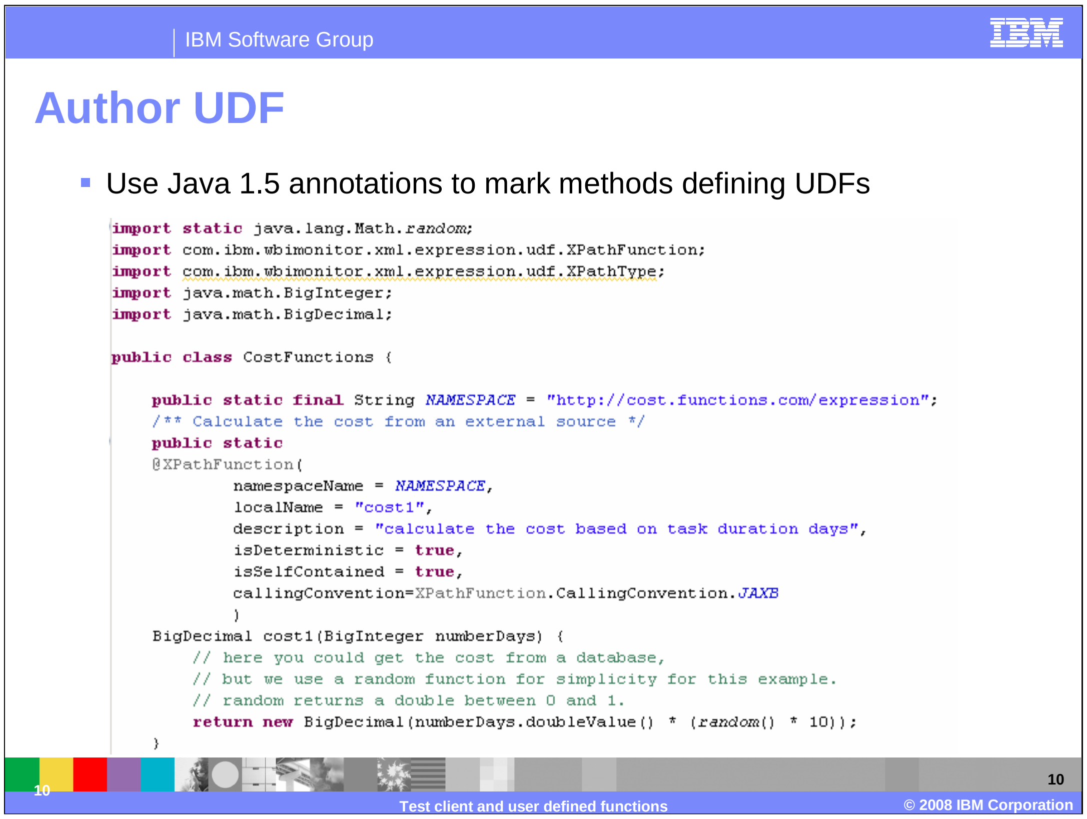
User defined functions (UDF)

- UDFs are implemented in Java by a UDF developer
- Monitor model developer can reference the UDF in maps, triggers, correlation, filters including metrics and KPIs
  - UDFs can be used in XPath 2.0 expressions



```
fn:concat("The estimated cost is: ", xs:string(
  usr:get-cost-estimate(projectDuration, projectStart, costOfMaterial)))
```

New in 6.1 is the ability to create user defined functions in Java. These can be used to access external data or to do specialized calculations. They can be referenced in expressions which are used in metric maps, triggers and in KPI expressions. Expression support is based on XPath 2.0. In this example, the return value of a user-defined XPath function named **get-cost-estimate** is used as an argument to the built-in function *concat* - which concatenates two String values. Since the value returned by **get-cost-estimate** is not a String, the return value is passed first to the built-in function string to convert the value to a String.



Author UDF

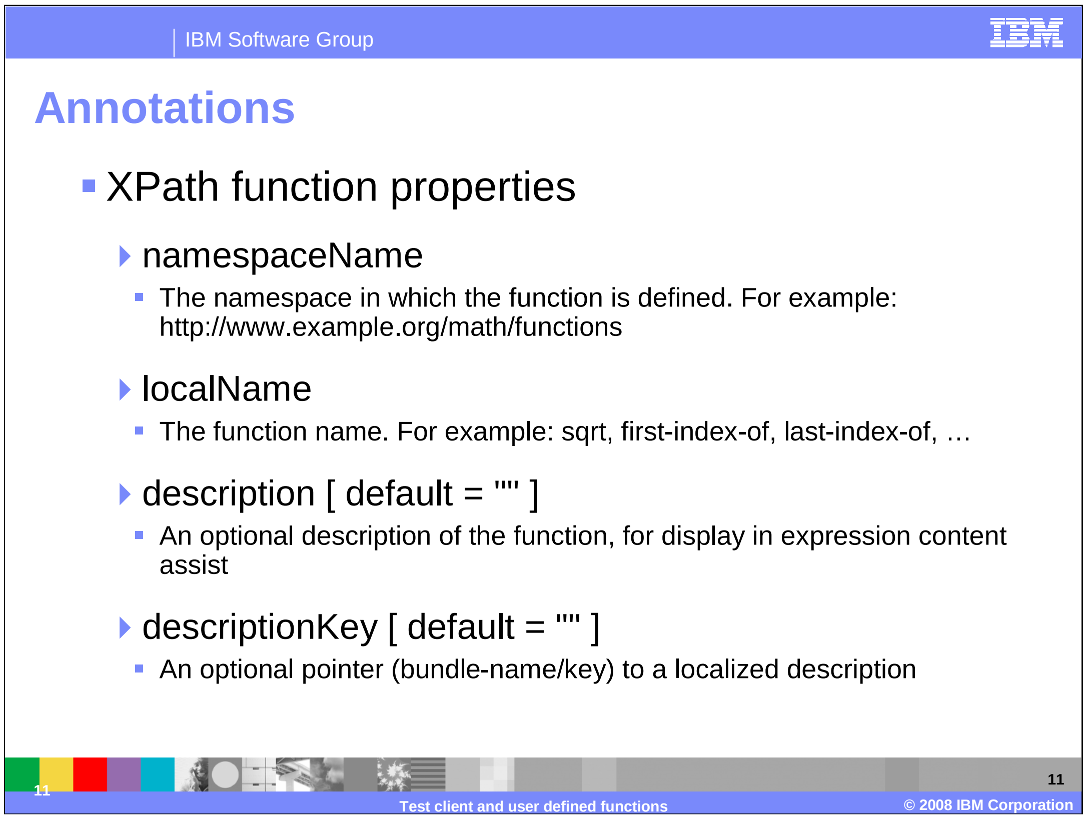
- Use Java 1.5 annotations to mark methods defining UDFs

```
import static java.lang.Math.random;
import com.ibm.wbimonitor.xml.expression.udf.XPathFunction;
import com.ibm.wbimonitor.xml.expression.udf.XPathType;
import java.math.BigInteger;
import java.math.BigDecimal;

public class CostFunctions {

    public static final String NAMESPACE = "http://cost.functions.com/expression";
    /** Calculate the cost from an external source */
    public static
    @XPathFunction(
            namespaceName = NAMESPACE,
            localName = "cost1",
            description = "calculate the cost based on task duration days",
            isDeterministic = true,
            isSelfContained = true,
            callingConvention=XPathFunction.CallingConvention.JAXB
            )
    BigDecimal cost1(BigInteger numberDays) {
        // here you could get the cost from a database,
        // but we use a random function for simplicity for this example.
        // random returns a double between 0 and 1.
        return new BigDecimal(numberDays.doubleValue() * (random() * 10));
    }
```

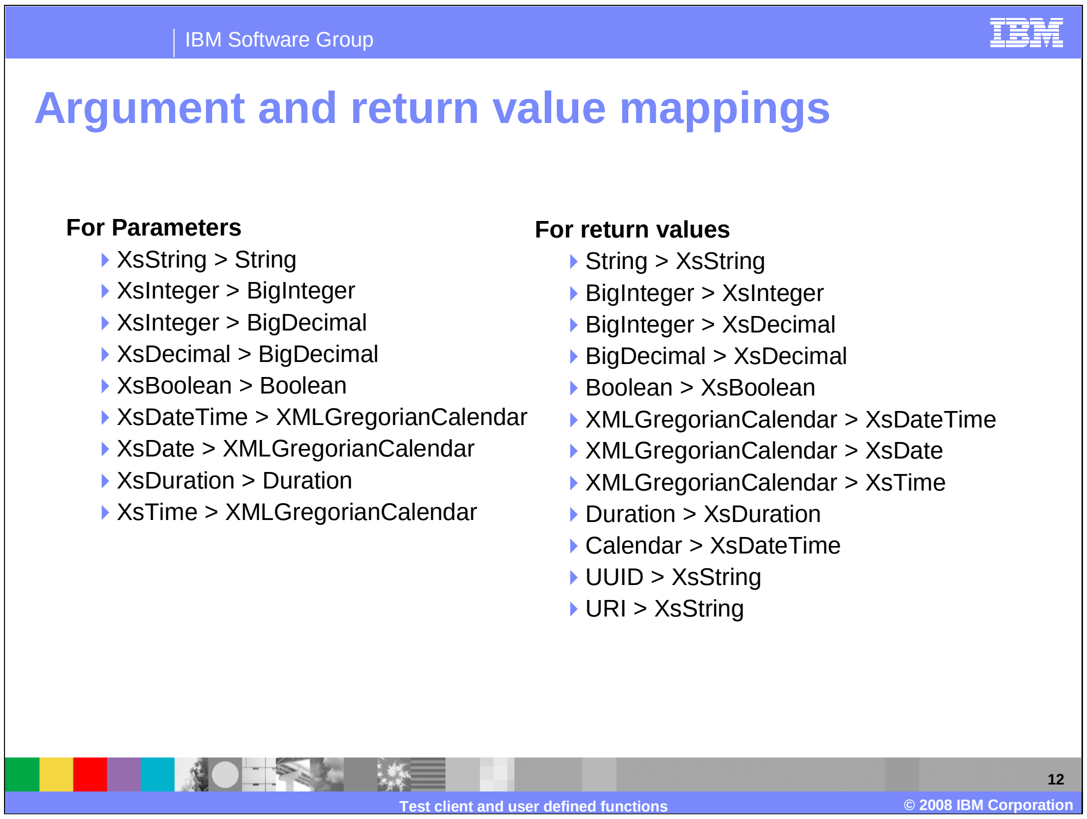
The Monitor Model Editor does not provide any special editing capabilities for writing user-defined XPath functions. This is because UDFs are Java classes and packages that can be easily developed using a Java project. The only requirement is to use Java 1.5 annotations to mark methods that are going to be used as UDFs.



Annotations

- XPath function properties
  - namespaceName
    - The namespace in which the function is defined. For example: http://www.example.org/math/functions
  - localName
    - The function name. For example: sqrt, first-index-of, last-index-of, …
  - description [default = ""]
    - An optional description of the function, for display in expression content assist
  - descriptionKey [default = ""]
    - An optional pointer (bundle-name/key) to a localized description

These are the XPath function properties which are annotated in the Java class. You can define the namespace, the function name, a description and an optional pointer to a localized description. Check the information center for the full list of annotations.



Argument and return value mappings

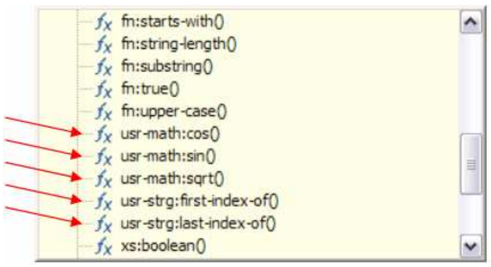
For Parameters	For return values
XsString > String	String > XsString
XsInteger > BigInteger	BigInteger > XsInteger
XsInteger > BigDecimal	BigInteger > XsDecimal
XsDecimal > BigDecimal	BigDecimal > XsDecimal
XsBoolean > Boolean	Boolean > XsBoolean
XsDateTime > XMLGregorianCalendar	XMLGregorianCalendar > XsDateTime
XsDate > XMLGregorianCalendar	XMLGregorianCalendar > XsDate
XsDuration > Duration	XMLGregorianCalendar > XsTime
XsTime > XMLGregorianCalendar	Duration > XsDuration
	Calendar > XsDateTime
	UUID > XsString
	URI > XsString

Since you are implementing a Java method that will be used as an XPath function, you need to be aware of the mappings from Java to XML data types.

Monitor currently supports eight primitive XML Schema data types which are listed in this table.

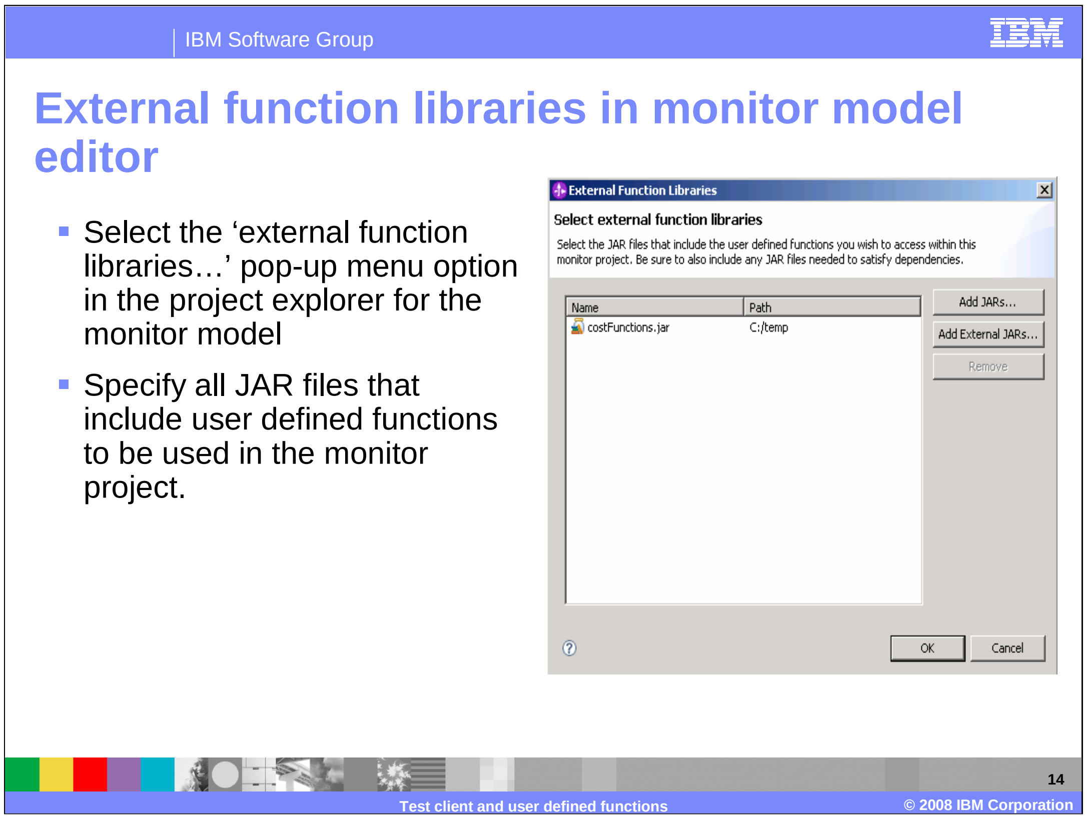
Monitor model editor expression support

- Expression content assist will offer UDFs together with built-in functions, using the model-defined namespace prefixes
  - fn functions are predefined XPath 2.0 functions
  - xs functions are predefined XML schema functions
- Validator will validate argument types and return types



fn:starts-with()
fn:string-length()
fn:substring()
fn:true()
fn:upper-case()
usr-math:cos()
usr-math:sin()
usr-math:sqrt()
usr-strg:first-index-of()
usr-strg:last-index-of()
xs:boolean()

Expression content assist shows you the user defined functions together with built-in functions, using the model-defined namespace prefixes. fn functions are predefined XPath 2.0 functions. xs functions are predefined XML schema functions. The monitor model editor validator will validate both the argument types and return types for your user defined functions.



External function libraries in monitor model editor

- Select the ‘external function libraries…’ pop-up menu option in the project explorer for the monitor model
- Specify all JAR files that include user defined functions to be used in the monitor project.

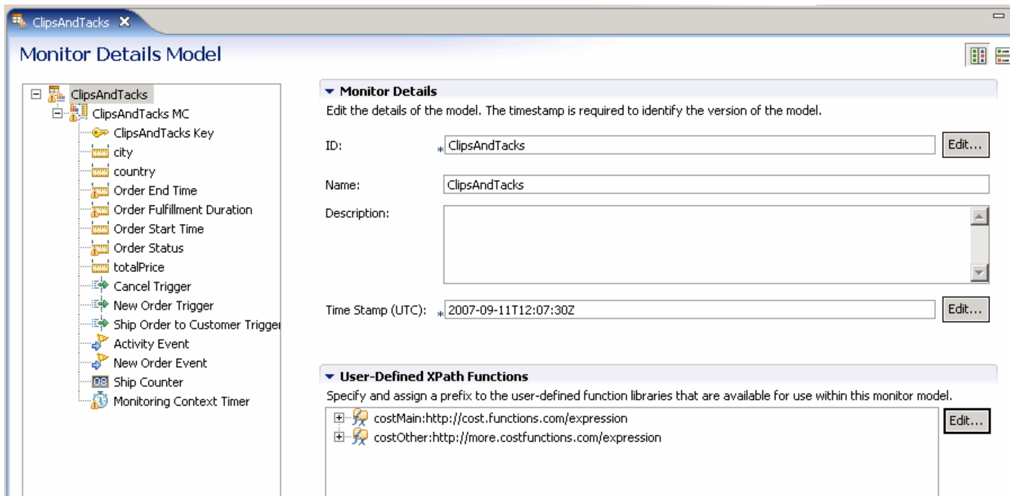
Select external function libraries

Select the JAR files that include the user defined functions you wish to access within this monitor project. Be sure to also include any JAR files needed to satisfy dependencies.

Name	Path
costFunctions.jar	C:/temp

To use the user defined functions in the monitor model editor, you need to define the JAR file in the external function libraries dialog. First right click on the monitor model in question, and then select ‘External function libraries’. Here you specify all JAR files that include user defined functions to be used in the model, and all JARs that make up the dependency tree for those functions.

User defined functions in monitor model editor


Monitor Details Model

ClipsAndTacks
ClipsAndTacks MC
ClipsAndTacks Key
city
country
Order End Time
Order Fulfillment Duration
Order Start Time
Order Status
totalPrice
Cancel Trigger
New Order Trigger
Ship Order to Customer Trigger
Activity Event
New Order Event
Ship Counter
Monitoring Context Timer

Monitor Details

Edit the details of the model. The timestamp is required to identify the version of the model.

ID: ClipsAndTacks

Name: ClipsAndTacks

Description:

Time Stamp (UTC): 2007-09-11T12:07:30Z

User-Defined XPath Functions

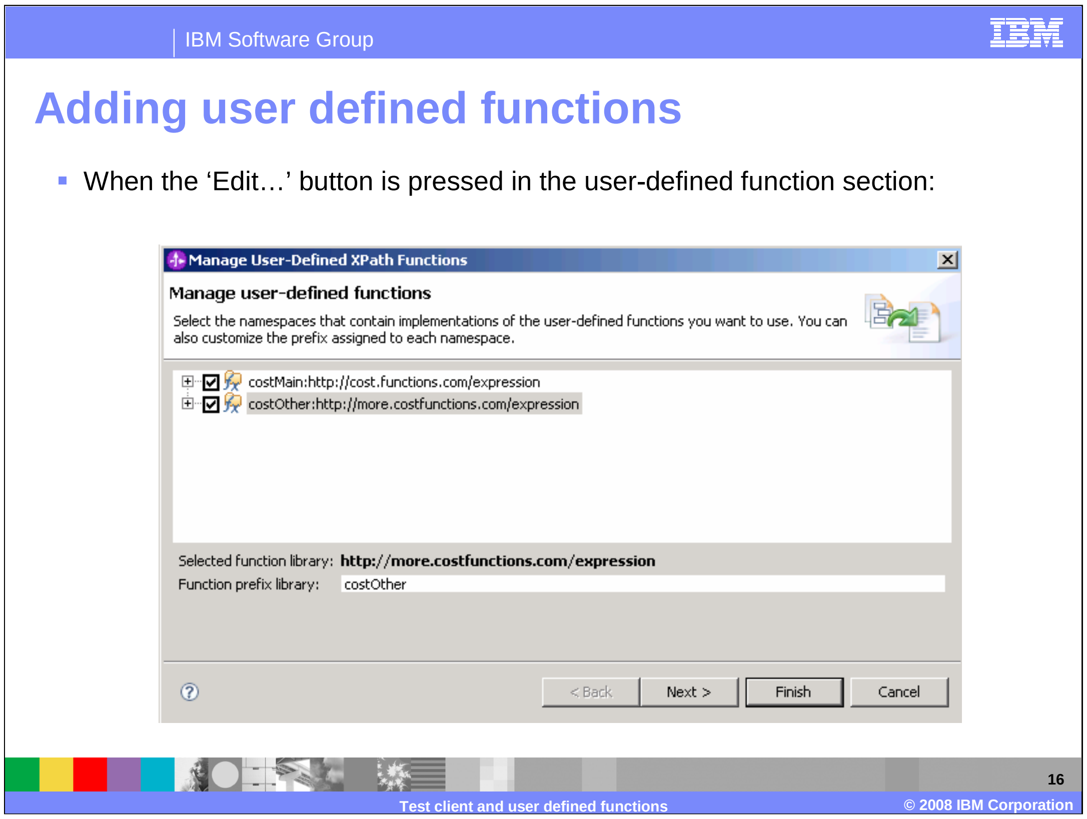
Specify and assign a prefix to the user-defined function libraries that are available for use within this monitor model.

costMain:http://cost.functions.com/expression
costOther:http://more.costfunctions.com/expression

For a given monitor model, the available user-defined functions are shown in a tree at the root model level, with the prefix and namespace listed for them.

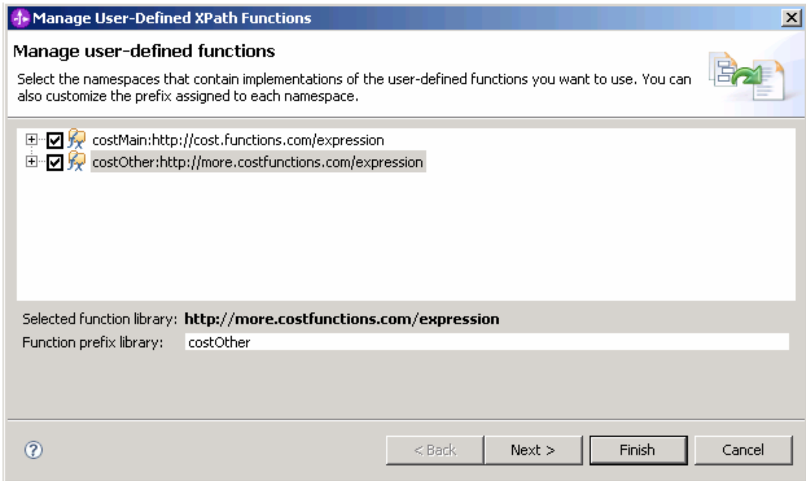
If you have just added a external function library, the functions will not show in this list until you click the edit button for this list. Adding new functions and editing existing namespace prefixes is done using the ‘Edit…’ button.

If you update a namespace prefix, the refactoring wizard will automatically update existing expressions that are using the current prefix.



Adding user defined functions

- When the 'Edit…' button is pressed in the user-defined function section:


Manage User-Defined XPath Functions
Manage user-defined functions

Select the namespaces that contain implementations of the user-defined functions you want to use. You can also customize the prefix assigned to each namespace.

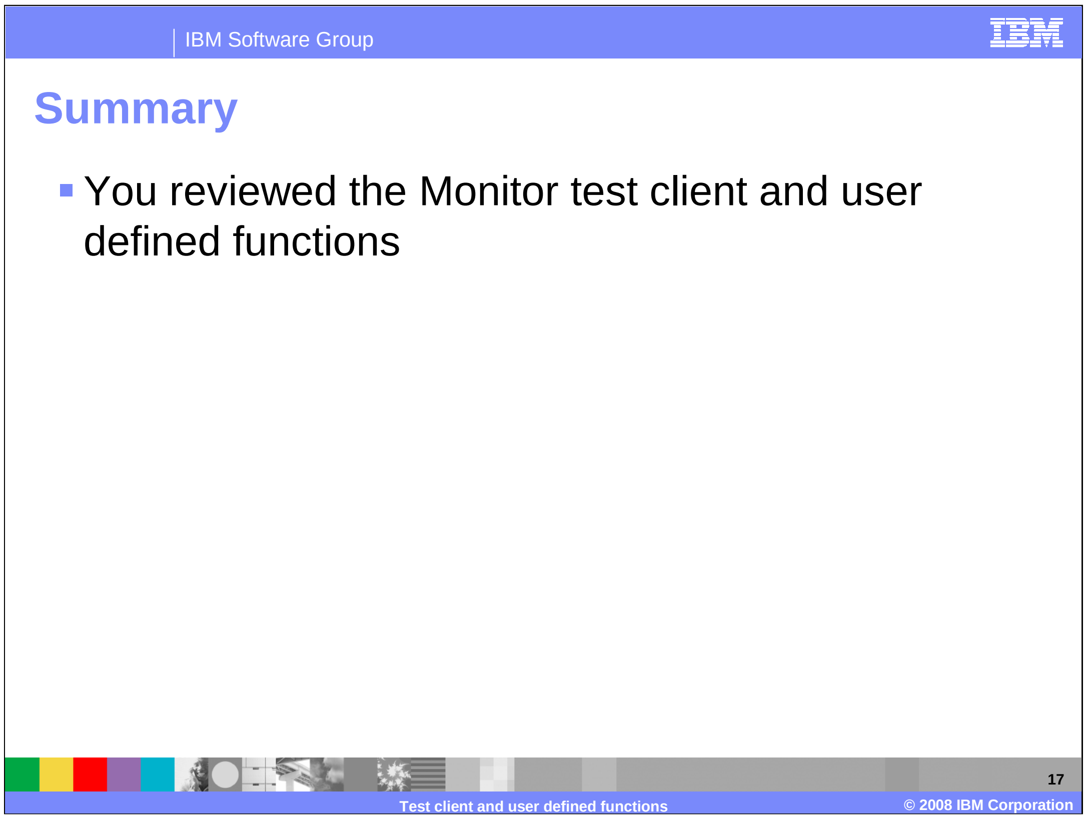
- [✓] costMain:http://cost.functions.com/expression
- [✓] costOther:http://more.costfunctions.com/expression

Selected function library: **http://more.costfunctions.com/expression**

Function prefix library: costOther

When you click to edit the user defined functions in the model, a tree shows all Java classes in the external functions libraries for the model. Namespaces require a prefix. This can be accomplished by selecting the tree node and updating the namespace in the field below.

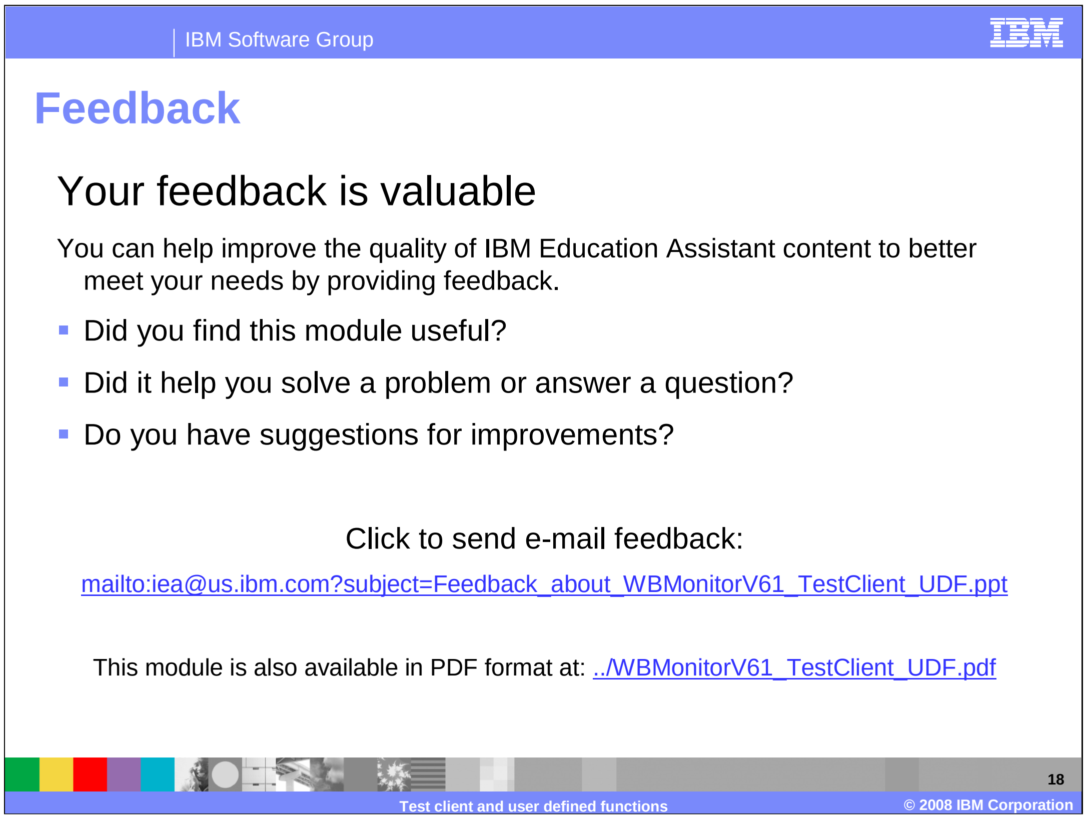
You can choose which classes to make available to the model by using the check boxes.



Summary

- You reviewed the Monitor test client and user defined functions

In this presentation you have reviewed the integrated test client and user defined functions in WebSphere Business Monitor version 6.1.



Feedback

Your feedback is valuable

You can help improve the quality of IBM Education Assistant content to better meet your needs by providing feedback.

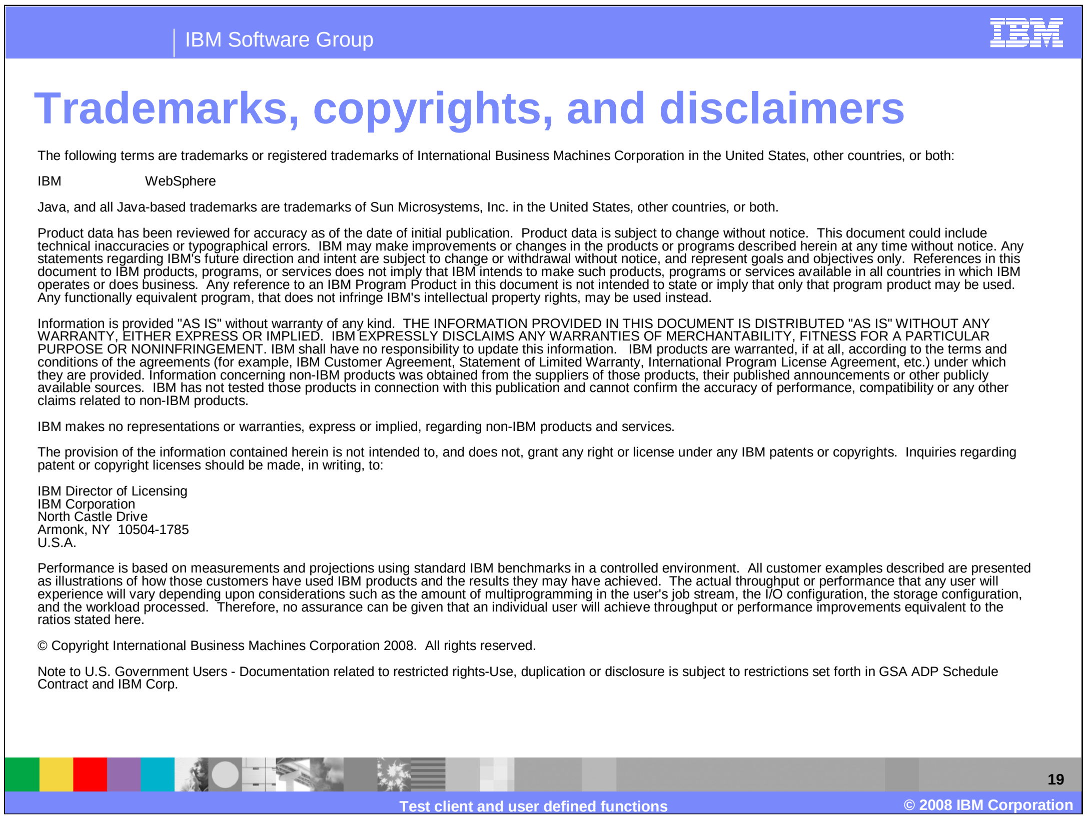
- Did you find this module useful?
- Did it help you solve a problem or answer a question?
- Do you have suggestions for improvements?

Click to send e-mail feedback:

mailto:iea@us.ibm.com?subject=Feedback_about_WBMonitorV61_TestClient_UDF.ppt

This module is also available in PDF format at: ../WBMonitorV61_TestClient_UDF.pdf

You can help improve the quality of IBM Education Assistant content by providing feedback.



Trademarks, copyrights, and disclaimers