1.	Record Nr.	UNINA9910823745903321
	Autore	Manoel Edson
	Titolo	Problem determination using self-managing autonomic technology / / Edson Manoel et al
	Pubbl/distr/stampa	San Jose, CA, : IBM International Technical Support Organization, 2005
	Descrizione fisica	1 online resource (412 p.)
	Collana	IBM redbooks
	Altri autori (Persone)	NielsonMorten Jul SalahshourAbdi Sampath K. V. LSai SudarshananSanjeev
	Soggetti	Information technology Information resources management
	Lingua di pubblicazione	Inglese
	Formato	Materiale a stampa
	Livello bibliografico	Monografia
	Note generali	Bibliographic Level Mode of Issuance: Monograph
	Nota di bibliografia	Includes bibliographical references (p. 381-384) and index.
	Nota di contenuto	Front cover -- Contents -- Notices -- Trademarks -- Preface -- The team that wrote this redbook -- Become a published author -- Comments welcome -- Part 1 Introduction -- Chapter 1. Autonomic computing overview -- 1.1 What is autonomic computing? -- 1.2 Autonomic computing concepts -- 1.2.1 Autonomic manager -- 1.2.2 Managed resources -- 1.2.3 Autonomic manager collaboration -- 1.2.4 Autonomic manager knowledge -- 1.2.5 Policies for autonomic managers -- 1.3 Evolving to autonomic computing -- 1.4 Value of autonomic computing -- 1.4.1 Software vendors -- 1.4.2 Enterprises -- 1.5 Core capabilities -- 1.5.1 Solution installation -- 1.5.2 Common systems administration -- 1.5.3 Problem determination -- 1.5.4 Autonomic monitoring -- 1.5.5 Complex analysis -- 1.5.6 Policy-based management -- 1.5.7 Heterogeneous workload management -- 1.5.8 Summary -- Part 2 Self-Managing Autonomic Technology -- Chapter 2. IBM Autonomic Computing Toolkit overview -- 2.1 Key component areas -- 2.2 IBM Autonomic Computing Toolkit tools and technologies -- 2.2.1 Autonomic Management Engine -- 2.2.2 Resource Model Builder -- 2.2.3 Generic Log Adapter -- 2.2.4 Log and Trace Analyzer -- 2.2.5 Agent Controller -- 2.2.6 Integrated Solutions Console --

2.2.7 Eclipse tooling package -- 2.3 IBM Autonomic Computing Toolkit scenarios -- 2.3.1 Problem Determination scenario -- 2.3.2 Solution Install and Deployment scenario -- 2.4 Supporting the autonomic computing architecture -- 2.5 Summary -- Chapter 3. Common Base Events -- 3.1 What is a Common Base Event? -- 3.2 Common Base Event structure -- 3.3 Common Base Event properties -- 3.4 Common Base Event required elements -- 3.4.1 The sourceComponentId and reporterComponentId elements -- 3.4.2 The creationTime element -- 3.4.3 The situation element -- 3.5 Common Base Event optional elements -- 3.5.1 The version element.

3.5.2 The localInstanceId element -- 3.5.3 The globalInstanceId element -- 3.5.4 The severity element -- 3.5.5 The priority element -- 3.5.6 The msg element -- 3.5.7 The extensionName element -- 3.5.8 The repeatCount element -- 3.5.9 The elapsedTime element -- 3.5.10 The sequenceNumber element -- 3.5.11 The contextDataElement element -- 3.5.12 The ExtendedDataElement element -- 3.5.13 The msgDataElement element -- 3.5.14 The associatedEvent element -- 3.5.15 The associationEngine element -- 3.6 Generating Common Base Events -- 3.7 Generating Common Base Events using APIs -- 3.7.1 Creating the Common Base Event -- 3.7.2 Filling in the Common Base Event -- Chapter 4. Generic Log Adapter -- 4.1 Generic Log Adapter overview -- 4.2 Creating a simple adapter -- 4.2.1 Creating the adapter configuration file -- 4.2.2 Configuring the Common Base Event elements -- 4.2.3 Configuring the adapter outputter -- 4.2.4 Testing the adapter configuration file -- 4.3 Creating a multiple contexts adapter -- 4.3.1 Adding a new context instance to the adapter file -- 4.3.2 Adding a second context to the adapter file -- 4.3.3 Associating the new context components -- 4.3.4 Testing the multiple context instance adapter file -- 4.4 Deploying and running the adapter configuration file -- 4.4.1 Starting the Generic Log Adapter Runtime from a Java class -- 4.4.2 Starting the Generic Log Adapter Runtime from CLI -- 4.5 Writing a custom outputter -- 4.5.1 Creating a custom outputter named ITSOSHIMOutputter -- 4.5.2 Testing the custom outputter -- Chapter 5. Log and Trace Analyzer -- 5.1 Log and Trace Analyzer overview -- 5.1.1 Viewing the log files -- 5.1.2 Analyzing the log files using the symptom database -- 5.1.3 Importing a symptom database -- 5.1.4 Working with multiple symptom databases -- 5.1.5 Correlating the log files.

5.2 Using Agent Controller for monitoring remote logs -- 5.3 Creating a custom log parser for the Log and Trace Analyzer -- 5.3.1 Creating the log parser plug-in -- 5.3.2 Creating the adapter configuration file -- 5.3.3 Testing the log parser -- 5.3.4 Deploying the custom parser -- 5.4 Creating a custom correlator for the Log and Trace Analyzer -- 5.4.1 Creating the custom correlation plug-in -- 5.4.2 Testing the custom correlation engine -- 5.4.3 Deploying the custom correlation -- Chapter 6. Autonomic Management Engine and resource models -- 6.1 Autonomic Management Engine -- 6.1.1 Introduction to Autonomic Management Engine -- 6.1.2 Autonomic Management Engine structure -- 6.2 Resource models -- 6.2.1 Resource Model Builder -- 6.2.2 Common Information Model classes -- 6.2.3 Classic probes -- 6.2.4 Events -- 6.2.5 Thresholds -- 6.2.6 Parameters -- 6.2.7 Logging -- 6.2.8 Dependencies -- 6.2.9 Source (decision tree script) -- 6.3 Simple Agent Reference Application (SARA) -- 6.4 How to debug a resource model -- 6.4.1 Debugging JavaScript -- 6.4.2 Debugging CIM classes with SARA -- Chapter 7. Advanced Autonomic Management Engine topics -- 7.1 Embedding the Autonomic Management Engine into applications -- 7.1.1 Designing the embedding application -- 7.1.2 Building and launching the embedding application -- 7.2 Structure of resource model -- 7.2.1 Directory tree structure -- 7.2.2 The rme. config file -- 7.2.3 The m12_mode.properties file -- 7.2.4 Structure of the JavaScript file -- 7.2.5 Structure of the resource model descriptor file -- 7.2.6 Structure of the MOF file -- Part 3 Case study scenarios -- Chapter 8. Problem Determination scenario one -- 8.1 Scenario description -- 8.2 Resource model definition -- 8.2.1 Creating the Eclipse project -- 8.2.2 General resource model settings -- 8.3 Creating the Managed Object Format file.

8.4 Creating the Common Information Model classes -- 8.5 Implementing the utility class -- 8.6 Implementing the Instrumentation Library Type class -- 8.7 Creating the installation package -- 8.7.1 Creating a JAR file with the ILT class and utility class -- 8.7.2 Registering custom CIM class using Resource Model Builder -- 8.7.3 Creating the resource model package -- 8.8 Scenario execution -- 8.8.1 Starting the scenario -- 8.8.2 Stopping the scenario execution -- Chapter 9. Problem Determination scenario two -- 9.1 Scenario description -- 9.2 Scenario development and implementation -- 9.2.1 Database node module one -- 9.2.2 Database node module two -- 9.2.3 Database node module three -- 9.2.4 Application node module one -- 9.2.5 Application node module two -- 9.3 Scenario execution -- 9.3.1 Starting the database node module one -- 9.3.2 Starting the database node module two -- 9.3.3 Starting the database node module three -- 9.3.4 Starting the application node module one -- 9.3.5 Starting the application node module two -- 9.3.6 The self-healing environment -- Chapter 10. Problem Determination scenario three -- 10.1 Scenario description -- 10.2 Scenario development and implementation -- 10.2.1 Database node module -- 10.2.2 Application node module -- 10.2.3 Administration node module -- Appendix A. Additional material -- Locating the Web material -- Using the Web material -- System requirements for downloading the Web material -- How to use the Web material -- Related publications -- IBM Redbooks -- Other publications -- Online resources -- How to get IBM Redbooks -- Help from IBM -- Index -- Back cover.

Sommario/riassunto

This IBM Redbooks publication provides practical information related to problem determination using IBM Self-Managing Autonomic Technology, with an emphasis on the components provided by the IBM Autonomic Computing Toolkit. The primary objective of this book is to empower software developers to provide facilities to enable applications or other system components to participate in an autonomic environment and focuses on the problem determination capabilities provided by the IBM Autonomic Computing Toolkit and other Self-Managing Autonomic Technology. The tools, components, and techniques used and demonstrated in this book provide the base for automating problem determination and self-healing of applications and components using Self-Managing Autonomic Technology. As with the technologies themselves, components of the IBM Autonomic Computing Toolkit will evolve as new capabilities and tools become available. Software developers are encouraged to check the latest releases of both IBM Autonomic Computing Toolkit and other tools provided by IBM at http://www.ibm.com/developerworks/autonomic. Please note that the additional material referenced in the text is not available from IBM.