



Feature List

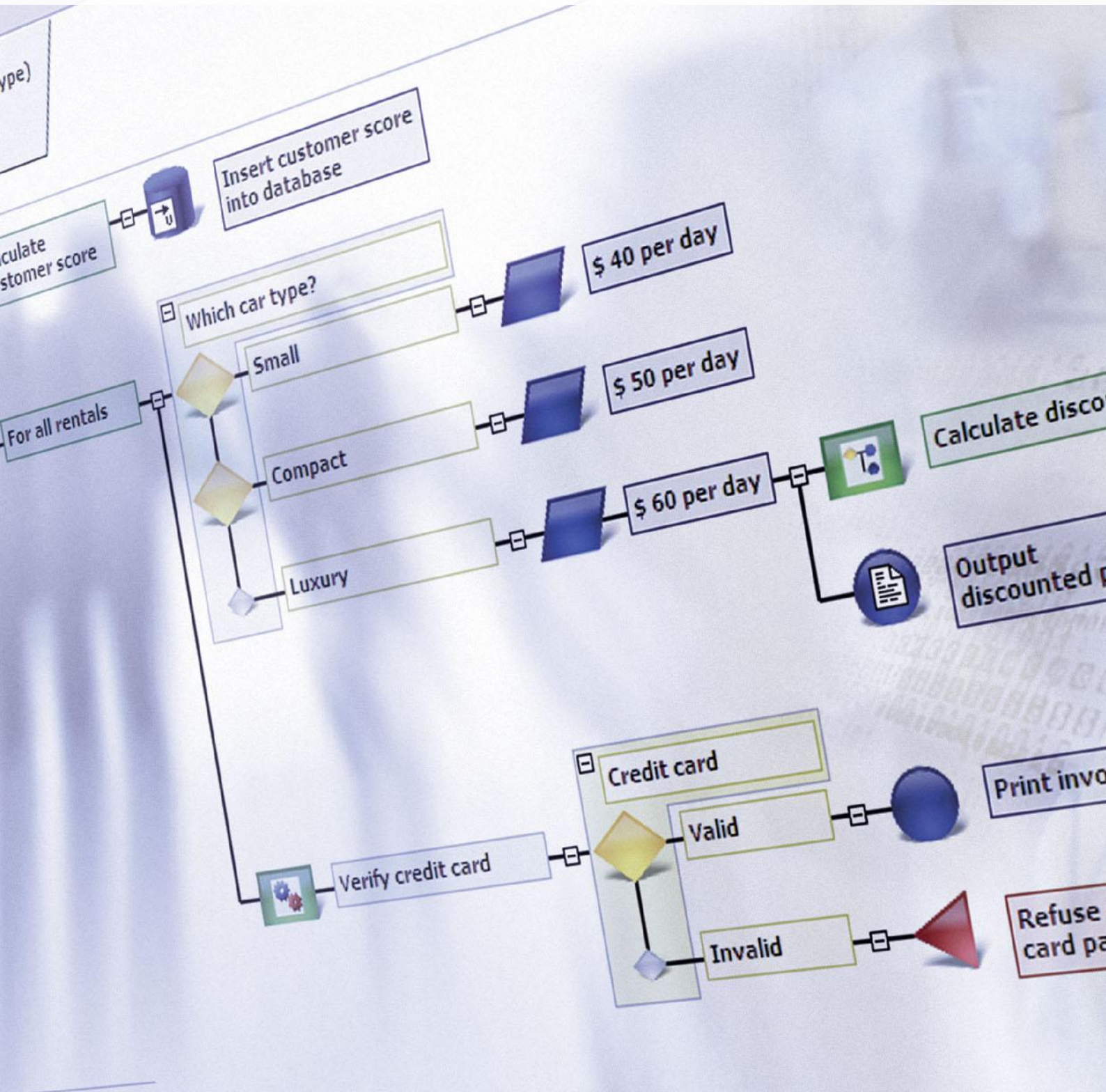
Visual Rules 6.1

Bosch Software Innovations



BOSCH

Invented for life



Contents

Page

3	Visual Rules Modeler
6	Visual Rules Team Platform
7	Visual Rules Execution Platform
8	Visual Rules Batch Platform
9	Visual Rules DatabaseIntegrator
9	Visual Rules Builder
10	Technical Limits
12	Contact

Visual Rules Modeler

Graphical Rule Editor

- ▶ Graphical display and editing of flow rules and decision tables with drag&drop
- ▶ Rule context palette for quick access to rule-specific data and tests
- ▶ Hover palette for quick insertion of rule elements
- ▶ In-place editing of rule expressions and descriptions
- ▶ Live syntax check and syntax highlighting
- ▶ Code assist to quickly access available data, operators and functions
- ▶ Automatic validation and consistency checks for rules
- ▶ Rule drafting mode (edit descriptions and/or notes only)
- ▶ Display of statistics (number of visits, percentages, timings) in the rule editor
- ▶ Dynamic layout and zoom
- ▶ Undo/redo of changes
- ▶ Hyperlinks and tooltips for expressions, data, functions etc.
- ▶ Automatic creation of data elements, types, actions etc. when first used
- ▶ Automatic refactoring of rules after renaming elements
- ▶ Editorial notes attachable to rule elements
- ▶ Comments and links to external documents for every rule element
- ▶ Outline view of rule for quick navigation
- ▶ Collapsible/expandable rule branches
- ▶ Multi page printouts of flow rules and decision tables
- ▶ Export of rules as JPG, PNG or BMP
- ▶ Support for rule artifacts (packaged JARs)
- ▶ Automatic extraction of flow rule branches into new flow rules
- ▶ Automatic conversion of decision tables to flow rules and back
- ▶ Creation of flow rules based on Java interfaces

Rule Management

- ▶ Rules and rule models can be hierarchically structured using rule packages
- ▶ Multiple rule models can be opened simultaneously
- ▶ Rule packages can import other rule packages and models for reuse of rules, data types, functions, action types and services
- ▶ Rule browser to quickly find and open any rule
- ▶ Contextual rule model search in names, descriptions, notes or expressions
- ▶ Interactive visualizations of rule and rule package dependencies
- ▶ Search for usages of every rule, data element, action
- ▶ Graphical compare and merge views for rules and rule packages
- ▶ Editorial notes attachable to every element of a rule model
- ▶ Comments and links to external documents for every model element
- ▶ Export of rule model documentation as multi-page HTML, PDF or MS Word document
- ▶ Rule models can be managed in a version control system or Visual Rules Team Server
- ▶ Any Eclipse integrated version control system can be used
- ▶ Full Maven support to resolve dependencies between rule projects and other artifacts
- ▶ Versioning scheme to manage different versions of the same rule project, each with its own dependencies.
- ▶ Visualizing the usage of rules and their elements in a matrix

Visual Rules Modeler

Flow Rule Elements

- ▶ Decision with two exits (if – then – else)
- ▶ Decision with more than two exits (switch – case)
- ▶ Fire Action to define the results of decisions
- ▶ Assignment to define computations
- ▶ Call Rule and Call Decision Table to reuse other rules
- ▶ Call Service to invoke external services
- ▶ Repeat and End Repeat to apply rule parts multiple times
- ▶ Return to skip or terminate rules
- ▶ Return Exception and Handle Exception to capture exceptional situations

Decision Table Elements

- ▶ Nesting of decisions both to the left and at the top of the decision table
- ▶ Assignments to define outputs (constant or dynamic)
- ▶ Fire Action to trigger execution of specific actions
- ▶ Return Exception to indicate illegal condition combinations
- ▶ Execution of single or multiple matching conditions allows to create score cards

Actions

- ▶ Output to a file or to the screen
- ▶ Output to a CSV file (comma separated values)
- ▶ Output log message
- ▶ Send e-mails
- ▶ User defined actions to trigger any custom functionality or to connect to other systems

Functions

- ▶ Arithmetic functions
- ▶ String functions
- ▶ Date and time functions
- ▶ Functions for lists and collections (select, slice, sort, exists, filter, avg, sum, min, max etc.)
- ▶ User defined functions on any built-in or custom type

Data

- ▶ Definition of input/output data and actions for each rule, rule package or rule model
- ▶ Definition of internal and constant data for each rule, rule package or rule model
- ▶ Flexible data visibility on rule model, rule package or rule level
- ▶ Basic data types (Float, Integer, Boolean, String, Date, Time, Timestamp)
- ▶ User defined data types
- ▶ Structured data types
- ▶ Enumerations
- ▶ Type aliases
- ▶ Support for inheritance
- ▶ Automatic import and synchronization of Java object models
- ▶ Support for Collections (Collection, List, Set)
- ▶ Support for Maps
- ▶ Support for Java 5 Enums and Generics
- ▶ Definition of meta data on several rule model components

Rule Execution

- ▶ Rules are immediately executable from within the modeling environment
- ▶ Execution statistics and –protocols (rule request or session level)
- ▶ Graphical display of execution statistics including execution paths and execution times (profiling)
- ▶ Execution via API for individual processing

Visual Rules Modeler

Rule Test

- ▶ Integrated test editor to define test cases for rules
- ▶ Editors for test data and expected results
- ▶ Configuration and execution of tests
- ▶ Graphical display of test results and deviations from the expected results
- ▶ Test suite editor to define test suites and to execute all the tests in the test suite
- ▶ Service mocking during tests

Debugging

- ▶ Graphical debugging of locally or remotely executed rules
- ▶ Breakpoints in flow rules and decision tables
- ▶ Display of current data during rule execution

Java Code Generator

- ▶ Java code generator turns rules into Java source code
- ▶ Java code is lightweight, stateless and multithreading capable
- ▶ Rule code has very high performance, ideal for mass data handling
- ▶ Rule code can be directly integrated into applications (no server required)
- ▶ Incremental code generation ensures that code is only generated for changed rules

APIs and Extensibilities

- ▶ API to integrate and call generated rule code (stateful rule sessions and rule requests)
- ▶ API to integrate custom functions
- ▶ API to integrate custom actions
- ▶ API to integrate custom user interfaces into the Modeler for custom actions and functions
- ▶ API to integrate custom services
- ▶ API for statistics and profiling (request scope and session scope)
- ▶ API to import and merge external data models

Online Help and Documentation

- ▶ User interface, online help and documentation available in English
- ▶ Tutorial and sample projects included
- ▶ Included wizard for support requests and bug reports

Import

- ▶ Wizard for importing Excel sheets as decision tables
- ▶ CSV file import for static data
- ▶ Test data import for rule tests

State Flows

- ▶ Graphical editor for modeling State Flows with drag&drop
- ▶ Rules on enter and leave of states
- ▶ Simple State Flow example showing the basic modeling concepts
- ▶ Automatic validation and consistency checks for State Flows
- ▶ Java code generator turning State Flows into Java source code
- ▶ Execution API for executing State Flow models

Visual Rules Team Platform

Rule Repository

- ▶ Centralized, tenant-aware rule repository supporting tenant relations (like cooperation, SaaS models and “act-on-behalf-of)
- ▶ Accessing detailed information about the managed resources via web console
- ▶ Strict separation of tenant data
- ▶ Simultaneous rule authoring by multiple users without conflicts
- ▶ Easy publishing and synchronization of changes from different users
- ▶ User-, team- and tenant-based permissions on every level down to individual rules and build jobs
- ▶ Full version history and audit log
- ▶ Support of trunk, branch and tag versions
- ▶ Visualization, compare and merge of any rule model revisions
- ▶ Regular file support (e.g. Word documents can also be stored and versioned)
- ▶ Searching for rules or any other resources based on specific criterias
- ▶ Rule repository stored in databases
- ▶ Configurable audit logging in databases

Build Management

- ▶ Intuitive handling and graphical overview via webconsole
- ▶ Time-based building and testing of shared rule projects
- ▶ Publishing of rule services to Execution Server
- ▶ Publishing of rule libraries to Maven Repositories
- ▶ User, team and tenant based permissions for build processes
- ▶ Graphical build statistics like Test Coverage
- ▶ Monitoring of build processes
- ▶ Providing rule libraries to the Visual Rules Modeler

Maintenance

- ▶ Overview of licenses
- ▶ Installation Wizard
- ▶ Dynamic logging configuration
- ▶ RESTful services API for:
 - Repository Services
 - Build Services
 - Coverage Services

Please note:

APIs are subject to change without notice. Changes are documented in the respective documentations.

Web Modeler

- ▶ Creation of Flow Rules and Decision Tables
- ▶ Creation of Rule Models and Rule Packages
- ▶ Project Explorer to browse the contents of a Rule Project
- ▶ Viewing of rules
- ▶ Editing of rules directly in browser
- ▶ Viewing of tests
- ▶ Editing of tests directly in browser
- ▶ Generation and display of statistics
- ▶ Validation on the level of rule projects
- ▶ Automatic creation of missing elements (Magic Dialog)
- ▶ Administration of test suites
- ▶ Automatic completion of expressions
- ▶ Data handling for data elements, types, constants and groups
- ▶ Import of Excel files as decision tables

User Administration

- ▶ Central separate component for the administration of
 - Tenants
 - Users, teams and roles
 - Permissions
 - Domains and applications
- ▶ Intuitive user guidance
- ▶ Usage of external User Management systems e.g. LDAP
- ▶ More information documented here:
 - **Identity Management Specification**

Visual Rules Execution Platform

Execution Server

- ▶ Tenant-aware deployment of rules as web services (WSDL) to be used in SOA, e.g. for decision services in BPM
- ▶ Support of tenant relations
- ▶ Fine grained Access Rights for viewing, updating, downloading and executing Rule Services
- ▶ Integration of rules into .NET applications
- ▶ Simple rule deployment directly from Visual Rules Modeler
- ▶ Hot deployment of new and existing rules without any downtime or server restart
- ▶ Multiple versions of the same rules can be provided simultaneously (e.g. with different effective dates or for different organizational units or for champion/challenger scenarios)
- ▶ Specific interfaces for each rule including data types
- ▶ Versioned dependency management between rule libraries and any other Java libraries
- ▶ Metadata support and dynamic rule request routing
- ▶ Generation and retrieval of execution traces (statistics) to visualize in Visual Rules Modeler
- ▶ Support of external databases to store the deployed rule artifacts and statistics (see system requirements)
- ▶ Dynamic logging configuration
- ▶ Configurable audit logging in databases
- ▶ Integration with central Identity Management (incl. LDAP support), more information documented here:
→ **Identity Management Specification**

Execution Core

- ▶ Tenant-aware integration of rules as services into Java applications
- ▶ Hot deployment of new and existing rules
- ▶ Multiple versions of rule services can be used simultaneously
- ▶ Rule services can be executed using Visual Rules Rule Execution API
- ▶ Versioned dependency management between rule libraries and Java libraries
- ▶ Support of external databases for storing and accessing libraries (see system requirements)
- ▶ Optional integration with central Identity Management (incl. LDAP support), more information documented here:
→ **Identity Management Specification**
- ▶ Can be operated in conjunction with an Execution Server and shares the same users and permissions

Visual Rules Batch Platform

General Features

- ▶ 100% Pure Java
- ▶ Open and Modularized Architecture
- ▶ Extensible via Custom Implementations
- ▶ Uses Industry Standards (e.g. Java, JEE)
- ▶ Extensive Code Documentation (JavaDoc)
- ▶ Supports the Spring Framework
- ▶ Shipped with Preconfigured and Executable Examples
Deployment of rules as web services (WSDL) to be used in SOA, e.g. for decision services in BPM

Spring Batch Integration

- ▶ XML-based language for configuring Batch Jobs
- ▶ Parallelization mechanisms for processing data: chunking, partitioning, and splitting flows
- ▶ Restartable Batch Jobs

Visual Rules Integration

- ▶ Fully Modelable Batch Processing Logic:
 - Processing Chunks using a Item Reader, Item Processor, and Item Writer
 - Processing Partitions using a Partition Generator and Partition Processor
 - Processing simple Tasks
- ▶ Visual Rules plug-ins for easy project creation
- ▶ Supports the Execution of Rule Models using the Execution Core
- ▶ Loads rule models from a specific URL / a classpath / a classpath resource / the Visual Rules artifact storage

Cluster Integration

- ▶ Mass-Data Processing in Scalable Distributed Environments
- ▶ Synchronous and asynchronous messaging for unicast and broadcast messages
- ▶ Command line interface to control the cluster
- ▶ Cluster Monitoring Service including Load Balancer and Blacklists for deactivated nodes

Commandline Interface

- ▶ Configurable and Extendable Command Line Interface

Visual Rules DatabaseIntegrator

DatabaseIntegrator

- ▶ Direct access to RDBMS from rules
- ▶ Import of data types from relational databases
- ▶ Definition and management of JDBC and JNDI connections
- ▶ Access to data with SQL SELECT action
- ▶ Change of data with SQL INSERT, UPDATE and DELETE actions
- ▶ Transaction control with COMMIT and ROLLBACK actions
- ▶ Call of stored procedures and stored functions with CALL action
- ▶ Creation and deletion of database elements (e.g. tables) with DDL

Visual Rules Builder

Builder

- ▶ Full integration of rules into Maven-based build and deployment processes
- ▶ Support of tenant relations
- ▶ Maven plugin for generating rule code
- ▶ Maven plugin for validating rule models
- ▶ Maven plugin for executing tests and test suites
- ▶ Maven plugin for creating documentation of test results
- ▶ Maven plugin for packaging rule artifacts (JARs with rule code, including WSDL and rule model files)
- ▶ Maven plugin for deploying these artifacts to Visual Rules Execution Server
- ▶ Maven plugin to create branches on Team Server
- ▶ Maven plugin to check in and out files on Team Server, optionally via tags
- ▶ Ant task VRCheckout to fetch rule projects from a Team Server repository
- ▶ Ant task VRCheckin to check in and out files on Team Server
- ▶ Ant task VRBranch to create branches on Team Server
- ▶ Ant task VRDeploy to automate rule deployment to Visual Rules Execution Server
- ▶ Ant task VRGenerate to automate code generation
- ▶ Ant task VRTest to automate tests

Technical Limits

There are some limitations regarding the setup and usage of Visual Rules. These limits have to be considered when integrating and working with Visual Rules. The currently identified limits are documented here but are subject to change in upcoming versions without notice. Using Visual Rules without complying to or exceeding these defined limits may be possible but is not recommended and is not supported. Not all limits are explicitly checked by the product but must be adhered to by other means.

Element	Type of Limit	Limit Value
Rule projects	per workspace	200
Rule models	per rule project	20
	per build execution	200
Rule packages	per rule model	200
	depth per rule package	20
Model path (rulemodel/package/ rule)	length (in characters)	200
- Rules - Data elements (e.g. Input/Output Data, Internal Data) - Custom service types - Custom action types - Custom functions	each element type per rule model	1000
Structures	per rule project (including reused structures)	10000
	attributes per structure	1000
	depth per structure	20
State flows	per rule model	200
	States per state flow	200
	Transitions per state flow	1000
Rule elements	per flow rule	2000
Cells	per decision table	2000
Rule tests & Test suites	rule tests per rule project	1000
	test suites per rule project	200
	rule tests per rule	200
	test cases per rule test	200
	rule tests/test suites per test suite	200
	rule tests per build execution	10000
Binary code artifact (vra, jar)	physical size	256MB
Execution result artifacts (vrexecutio, vrstatistics)	physical size	64MB
Expressions	length (in characters)	10000

Element	Type of Limit	Limit Value
Tenants	per Team Server and Execution Server instance	40
Repositories	per tenant	20
Branches	per tenant	2000
Projects	per branch	200
Files	per project	2000
Build Processes (Continuous + Release)	per tenant	200
Build history	per build process	20
Number of configurable JDKs, Maven Installations, Maven Repositories, Execution Servers	each type per Team Server instance	10
Rule services	per Execution Server cluster	4000
Dependency libraries	per Execution Server cluster	4000
Rule service executions	concurrently per Execution Server instance	400 *
Distinct rule services	loaded concurrently per Execution Server instance	100 *
Rule service libraries files	physical size	40 MB
Rule service metadata	number of metadata values	20
Batch Platform: entry rule models	per job step	Max. 3 for „Chunking“ Max. 1 for „Tasking“ Max. 2 for „Partitioning“
Rules implementing Batch Platform interfaces Interface types: IItemReader, IItemProcessor, ISimpleItemProcessor, IItemListWriter, IItemWriter, IPartitionGenerator, IPartitionProcessor, ITaskProcessor	per rule model	1 for each interface type
Processing Threads	per batch job	Max. amount of CPU cores
Configured jobs	per Batch Platform	100
Concurrent running jobs	per machine	1

*** Annotations:**

- Depending on available memory/memory settings (especially 32bit/64bit)
- Depending on CPU performance
- Given a typical size/complexity of the rule model and their dependencies

Restrictions of multi-tenancy support:

- Integrating custom services and functions in rule executions may not fully support multi-tenant separation of data
- Using external Maven repositories may not fully support multi-tenant separation of data

Europe	Americas	Asia
Bosch Software Innovations GmbH	Bosch Software Innovations Corp.	Bosch Software Innovations c/o Robert Bosch (SEA) Pte Ltd
Schöneberger Ufer 89-91 10875 Berlin Germany Tel. +49 30 726112-0 Fax +49 30 726112-100 info-de@bosch-si.com www.bosch-si.de	161 N. Clark Street Suite 3550 Chicago, Illinois 60601/USA Tel. +1 312 368-2500 Fax +1 312 368-5898 info@bosch-si.com www.bosch-si.com	11 Bishan Street 21 Singapore 573943 Tel. +65 6571 2220 Fax +65 6258 4671 info-sg@bosch-si.com www.bosch-si.sg

Bosch Software Innovations GmbH, the Bosch Group's software and systems house, designs, develops, and operates innovative software and system solutions that help our customers around the world both in the traditional enterprise environment and in the Internet of Things and Services (IoT). We place particular focus in this field on the topics of mobility, energy and building, manufacturing, and financial services. Whether in its special, targeted BPM+ and IoT editions or as flexible standalone products, our software suite is the perfect foundation not only for projects relating to the Internet of Things and Services but also for projects in the fields of Business Process Management (BPM) and Business Rules Management (BRM). Bosch Software Innovations has locations in Germany (Berlin, Immenstaad, Waiblingen), the United States (Chicago, Palo Alto, and Vienna), Singapore, China (Shanghai), and Australia (Melbourne).

More information can be found at www.bosch-si.com.

The **Bosch Group** is a leading global supplier of technology and services. In fiscal 2012, its roughly 306,000 associates generated sales of \$67.5 billion. Since the beginning of 2013, its operations have been divided into four business sectors: Automotive Technology, Industrial Technology, Consumer Goods, and Energy and Building Technology. The Bosch Group comprises Robert Bosch GmbH and its roughly 360 subsidiaries and regional companies in some 50 countries. If its sales and service partners are included, then Bosch is represented in roughly 150 countries. This worldwide development, manufacturing, and sales network is the foundation for further growth. Bosch spent some \$6.1 billion for research and development in 2012, and applied for nearly 4,800 patents worldwide. The Bosch Group's products and services are designed to fascinate, and to improve the quality of life by providing solutions which are both innovative and beneficial. In this way, the company offers technology worldwide that is "Invented for life."

Additional information is available online at www.bosch.com and www.bosch-press.com.