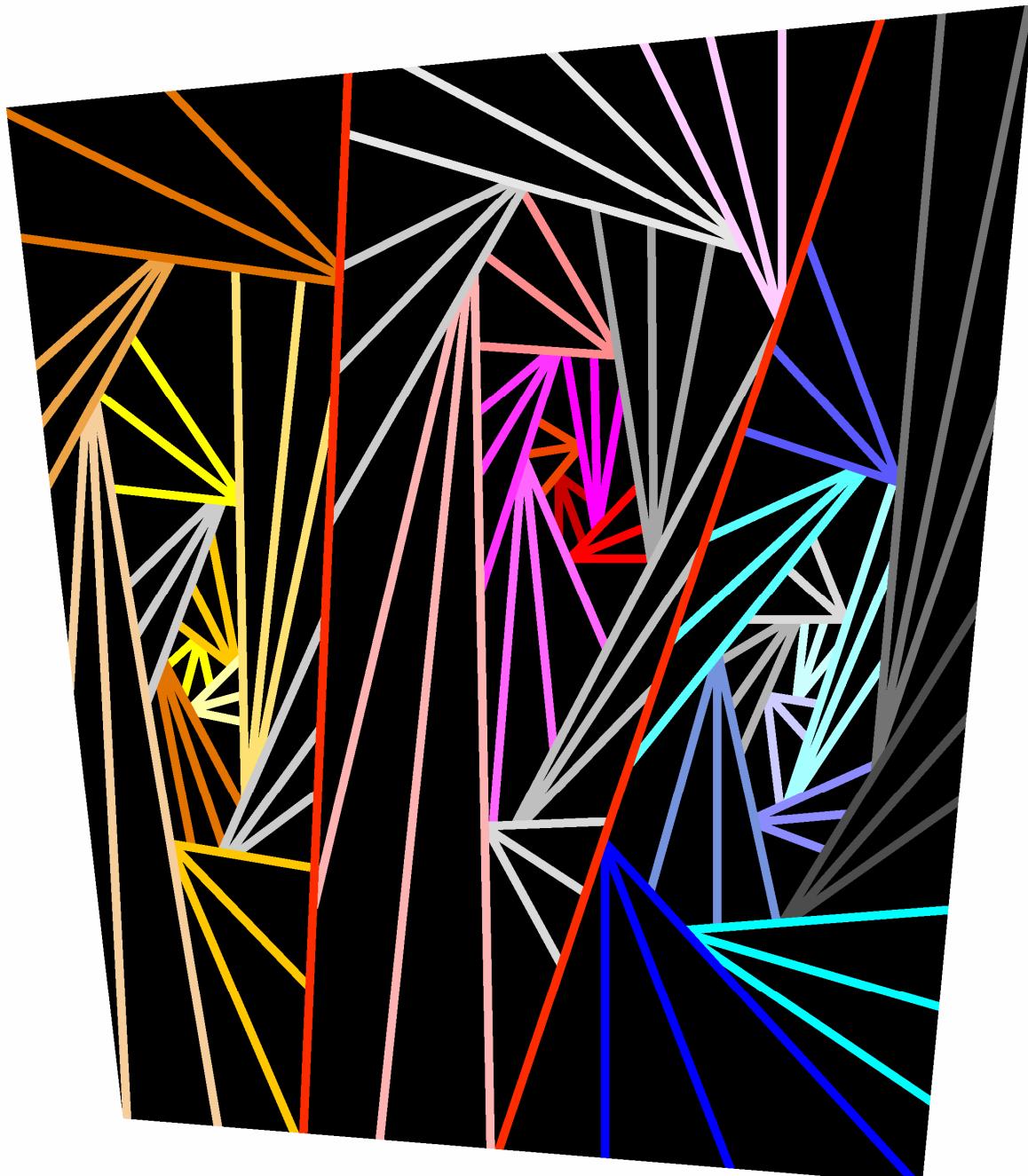


**RPS**

# Radiowave Propagation Simulator

radioplan



User Manual – Version 5.3

Enterprise Version

Professional Version

**Documentation Version:** UG-v5.3

**Software Version:** RPS-v5.3

**Authors:** Dr. J. Deißner, J. Hübner, D. Hunold, Dr. J. Voigt

Copyright © 1997–2005 by Radioplan GmbH.

All rights reserved.

*The products described in this user guide are continuously being developed and improved. For the sake of product innovation, the customer may apologize that some described features in this document might not exactly represent the current status of the product at the customers' desk.*

#### **Trademark Notice**

Radioplan is a registered trademark of Radioplan GmbH in the European Union. Microsoft, Windows, Windows NT, Windows 2000, MS Office, MS Excel, MS Outlook, Visual Basic, Visual C++, and ActiveX are trademarks or registered trademarks of the Microsoft Corporation. OpenGL is a registered trademark of Silicon Graphics, Inc. InstallShield is a registered trademark of the InstallShield Software Corporation. Unix is a trademark of X/Open Company Ltd. Linux is a registered trademark of Linus Torvalds. Matlab is a trademark or registered trademark of The Mathworks Inc. Corel is a registered trademark of the Corel Corporation. CORBA is a registered trademark of the Object Management Group, Inc. Stingray Studio is a trademark of Rogue Wave Software, Inc. AutoCAD is a registered trademark of Autodesk, Inc. VectorDraw is a trademark of VectorDraw Software Corporation. ESRI is a trademark or registered trademark of the Environmental Systems Research Institute (ESRI), Inc.

All other product or brand names are trademarks or registered trademarks of their respective holders.

#### **Contact:**

Radioplan GmbH  
Altmarkt 10  
D-01067 Dresden  
Germany  
tel.: +49 (0) 351 404 29 – 0  
fax: +49 (0) 351 404 29 – 50  
e-mail: [info@radioplan.com](mailto:info@radioplan.com)  
[www.radioplan.com](http://www.radioplan.com)

# Table of Contents

|         |   |    |
|---------|---|----|
| 1       | Overview .....                                | 15 |
| 1.1     | Introduction .....                            | 15 |
| 1.2     | Workflow Model.....                           | 16 |
| 1.3     | Summary of Main Functions and Modules.....    | 17 |
| 1.3.1   | Field Strength Prediction Algorithms.....     | 17 |
| 1.3.2   | Graphical User Interface.....                 | 17 |
| 1.3.3   | Open Simulation System.....                   | 18 |
| 1.4     | New Features in Version 5.3 .....             | 18 |
| 1.4.1   | Functionality .....                           | 18 |
| 1.4.2   | Convenience of Usage .....                    | 18 |
| 1.4.3   | Bugfixes.....                                 | 19 |
| 1.5     | New Features in Version 5.2 .....             | 19 |
| 1.5.1   | Functionality .....                           | 19 |
| 1.5.2   | Convenience of Usage .....                    | 19 |
| 1.5.3   | Bugfixes.....                                 | 20 |
| 1.6     | Upgrade .....                                 | 20 |
| 1.7     | System Requirements .....                     | 20 |
| 1.8     | Installation.....                             | 21 |
| 1.8.1   | RPS Software Package .....                    | 21 |
| 1.8.2   | RPS Installation Process .....                | 22 |
| 1.8.3   | Directory Structure .....                     | 25 |
| 1.8.4   | RPS Customization.....                        | 26 |
| 1.8.4.1 | Global RPS Settings .....                     | 26 |
| 1.8.4.2 | License Management Configuration.....         | 29 |
| 1.8.4.3 | Radio System Default Settings .....           | 30 |
| 1.8.4.4 | Transmitter Default Settings.....             | 30 |
| 1.8.4.5 | Material Database.....                        | 30 |
| 1.8.4.6 | MATLAB Syntax Highlighting Conventions .....  | 30 |
| 1.8.4.7 | Geographical Coordinate System Settings ..... | 30 |
| 1.8.5   | RPS Server Installation.....                  | 30 |
| 1.9     | Software Registration.....                    | 31 |
| 1.9.1   | Hardware Dongles .....                        | 31 |

|         |  |    |
|---------|--|----|
| 1.9.1.1 | Single Computer License .....                        | 31 |
| 1.9.1.2 | Floating License Management .....                    | 32 |
| 1.9.1.3 | Feature Updates for Hardware Dongles.....            | 33 |
| 1.9.2   | Software License Files .....                         | 34 |
| 1.9.2.1 | Registering RPS.....                                 | 35 |
| 1.9.2.2 | License Transfer.....                                | 35 |
| 2       | Getting Started.....                                 | 37 |
| 2.1     | Preparing the Environment Database .....             | 37 |
| 2.1.1   | Starting RPS.....                                    | 37 |
| 2.1.2   | Adding a new Building in the Environment .....       | 38 |
| 2.1.3   | Adding a Base Station (Transmitter) .....            | 39 |
| 2.1.4   | Adding Mobile Stations (Receivers) .....             | 40 |
| 2.2     | Field Strength Predictions .....                     | 42 |
| 2.2.1   | Preparing the Simulation.....                        | 42 |
| 2.2.2   | Performing a 2.5D Ray Tracing Simulation .....       | 42 |
| 2.3     | Prediction Results.....                              | 43 |
| 2.3.1   | 2D and 3D Views .....                                | 43 |
| 2.3.2   | Surface Plots .....                                  | 44 |
| 2.3.3   | Further Analysis.....                                | 45 |
| 2.3.4   | Assigning Radiation Patterns to Transmitters.....    | 46 |
| 2.4     | Incremental Recalculation of the Radio Channel ..... | 47 |
| 3       | Working with RPS — Overview .....                    | 49 |
| 3.1     | Components of the Working Environment.....           | 49 |
| 3.1.1   | Menu Bar .....                                       | 50 |
| 3.1.2   | Toolbars .....                                       | 51 |
| 3.1.3   | Tree Window .....                                    | 51 |
| 3.1.4   | Viewing Area .....                                   | 52 |
| 3.1.5   | Result Windows.....                                  | 52 |
| 3.1.6   | Message Window and Status Bar .....                  | 52 |
| 3.1.7   | Showing and Hiding Graphical Components.....         | 52 |
| 3.1.8   | Full Screen View .....                               | 53 |
| 3.1.9   | Clipboard Functions .....                            | 53 |
| 3.1.10  | Undo Buffer .....                                    | 53 |
| 3.2     | Coordinate Systems.....                              | 53 |
| 3.2.1   | Geographical and Cartesian Coordinate Systems .....  | 53 |

|         |  |    |
|---------|--|----|
| 3.2.2   | Euclidian Cartesian Coordinate Systems used in RPS .....                     | 55 |
| 3.3     | Layer Concept.....   | 57 |
| 3.3.1   | Material Layers.....   | 57 |
| 3.3.2   | Result Layers .....  | 59 |
| 3.4     | Personal Workspace.....  | 61 |
| 3.4.1   | Workspace Handling.....  | 61 |
| 3.4.1.1 | Handling of Legend Presets.....  | 62 |
| 3.4.2   | Display Settings.....  | 63 |
| 3.4.3   | Layer Settings .....   | 63 |
| 3.4.4   | Color Palette.....   | 66 |
| 3.4.4.1 | Inspecting the Color Palette of a Layer.....                                 | 66 |
| 3.4.4.2 | Using Legend Presets .....   | 68 |
| 3.4.4.3 | Import and Export of Legend Presets.....                                     | 69 |
| 3.5     | RPS Project Components and Principles.....                                   | 69 |
| 3.5.1   | Environment Objects .....  | 69 |
| 3.5.2   | Diffraction Edges.....   | 70 |
| 3.5.3   | Network Structure .....  | 71 |
| 3.5.3.1 | Antennas .....   | 72 |
| 3.5.3.2 | Transmitters .....   | 72 |
| 3.5.3.3 | Receivers .....  | 73 |
| 3.6     | Working with 2D and 3D Views.....  | 73 |
| 3.6.1   | Adjusting Surface Plots using the Components Toolbar .....                   | 73 |
| 3.6.2   | Zoom Functions .....   | 74 |
| 3.6.3   | Moving Graphics (Pan Mode) .....   | 75 |
| 3.6.4   | Navigation in 3D Views.....  | 75 |
| 3.6.4.1 | Rotation Mode.....   | 75 |
| 3.6.4.2 | Fly-by Mode .....  | 75 |
| 3.6.4.3 | Key Navigation.....  | 77 |
| 3.7     | Foundations of Geometrical Optics for Electromagnetic Wave Propagation ..... | 78 |
| 3.7.1   | Reflection.....  | 78 |
| 3.7.2   | Penetration .....  | 80 |
| 3.7.3   | Diffraction.....   | 80 |
| 3.7.4   | Polarization.....  | 81 |
| 3.7.5   | Ray Launching Approach .....   | 81 |
| 3.8     | Plug-in Support.....   | 82 |
| 3.8.1   | Plug-in Directories.....   | 83 |

|         |   |     |
|---------|---|-----|
| 3.8.2   | Attaching and Detaching the MATLAB Engine .....     | 84  |
| 3.8.3   | Working with Plug-ins.....                          | 84  |
| 3.8.3.1 | Plug-in Dialog .....                                | 84  |
| 3.8.3.2 | Selecting Plug-ins.....                             | 85  |
| 3.8.3.3 | Editing Custom Parameters of DLL Plug-ins.....      | 85  |
| 3.8.3.4 | Editing MATLAB Scripts .....                        | 85  |
| 3.8.3.5 | Matrix Processing and MATLAB User Library Path..... | 86  |
| 4       | Working with RPS — Configuration.....               | 87  |
| 4.1     | Environment Configuration .....                     | 87  |
| 4.1.1   | Functionality Overview .....                        | 88  |
| 4.1.1.1 | General Command Usage .....                         | 88  |
| 4.1.2   | Coordinate Systems.....                             | 89  |
| 4.1.3   | Views.....  | 89  |
| 4.1.3.1 | 2D Views.....                                       | 90  |
| 4.1.3.2 | Perspective Views.....                              | 90  |
| 4.1.3.3 | Shading.....  | 90  |
| 4.1.3.4 | View Rotation .....                                 | 90  |
| 4.1.3.5 | Zoom.....   | 91  |
| 4.1.4   | Background Image .....                              | 91  |
| 4.1.5   | Object Snap.....                                    | 92  |
| 4.1.6   | Material Layer Definition .....                     | 93  |
| 4.1.6.1 | Choosing a Material Layer.....                      | 93  |
| 4.1.6.2 | Creating a New Material Layer.....                  | 94  |
| 4.1.6.3 | Inspecting Material Properties .....                | 94  |
| 4.1.6.4 | Defining a DEM Layer.....                           | 96  |
| 4.1.6.5 | Material Database .....                             | 98  |
| 4.1.7   | Constructing Environment Objects.....               | 99  |
| 4.1.7.1 | Material Layer Selection .....                      | 99  |
| 4.1.7.2 | Constructing Typical Building Components.....       | 99  |
| 4.1.7.3 | Placing Vertices with Relative Positions .....      | 103 |
| 4.1.7.4 | Constructing the Ground.....                        | 103 |
| 4.1.7.5 | Other Environment Objects .....                     | 104 |
| 4.1.8   | Environment Import and Export.....                  | 104 |
| 4.1.9   | Working with Blocks .....                           | 104 |
| 4.1.9.1 | Creating a Block .....                              | 105 |
| 4.1.9.2 | Exploding a Block.....                              | 105 |
| 4.1.9.3 | Stencil Library .....                               | 105 |

|          |  |     |
|----------|--|-----|
| 4.1.9.4  | Block Import/Export .....                          | 107 |
| 4.1.10   | Modifying Environment Objects .....                | 108 |
| 4.1.10.1 | Inserting Additional Obstacles.....                | 108 |
| 4.1.10.2 | Moving Object Vertices.....                        | 116 |
| 4.1.10.3 | Changing Object Properties.....                    | 116 |
| 4.1.10.4 | Removing Objects .....                             | 117 |
| 4.1.10.5 | Cleaning up the Environment Database .....         | 117 |
| 4.1.11   | Supplementary Functions.....                       | 117 |
| 4.1.11.1 | Environment Editor Preferences.....                | 117 |
| 4.1.11.2 | Distance Measurement .....                         | 118 |
| 4.1.11.3 | Refreshing the View.....                           | 119 |
| 4.1.11.4 | Adding Flat Roofs .....                            | 119 |
| 4.1.11.5 | Detecting Diffraction Edges.....                   | 119 |
| 4.1.12   | Printing from the Environment Editor.....          | 119 |
| 4.2      | Network Configuration .....                        | 120 |
| 4.2.1    | Antenna .....                                      | 120 |
| 4.2.1.1  | Importing Antennas.....                            | 121 |
| 4.2.1.2  | Assigning an Antenna to the Receivers.....         | 122 |
| 4.2.1.3  | Inspecting the Antenna Configuration Data.....     | 122 |
| 4.2.1.4  | Viewing the Antenna Diagram .....                  | 123 |
| 4.2.1.5  | Deleting an Antenna .....                          | 124 |
| 4.2.2    | Transmitter .....                                  | 124 |
| 4.2.2.1  | Creating a Transmitter .....                       | 124 |
| 4.2.2.2  | Moving a Transmitter .....                         | 125 |
| 4.2.2.3  | Inspecting Transmitter Data .....                  | 126 |
| 4.2.2.4  | Viewing the Antenna Diagram of a Transmitter ..... | 131 |
| 4.2.2.5  | Viewing an LOS Preview Plot .....                  | 131 |
| 4.2.2.6  | Changing Transmitter States .....                  | 132 |
| 4.2.2.7  | Deleting a Transmitter .....                       | 133 |
| 4.2.3    | Receiver.....                                      | 133 |
| 4.2.3.1  | Receiver Groups.....                               | 133 |
| 4.2.3.2  | Placing Receivers .....                            | 134 |
| 4.2.3.3  | Duplicating a Receiver Group .....                 | 137 |
| 4.2.3.4  | Inspecting Receiver Data .....                     | 137 |
| 4.2.3.5  | Viewing the Antenna Diagram of the Receivers ..... | 138 |
| 4.2.3.6  | Deleting Receivers .....                           | 138 |
| 5        | Working with RPS — Simulation .....                | 141 |

|         |  |     |
|---------|--|-----|
| 5.1     | Simulation Algorithms .....                              | 141 |
| 5.1.1   | Ray Tracing Algorithms.....                              | 141 |
| 5.1.1.1 | 3D Ray Tracing Algorithms.....                           | 141 |
| 5.1.1.2 | Satellite Mode.....                                      | 142 |
| 5.1.1.3 | 2.5D Ray Tracing Algorithms .....                        | 142 |
| 5.1.1.4 | Best Fitting Algorithm for Ray Tracing Simulations.....  | 143 |
| 5.1.2   | Empirical and Semi-Empirical Prediction Algorithms ..... | 143 |
| 5.1.2.1 | Empirical Model for Outdoor Environments.....            | 143 |
| 5.1.2.2 | Empirical Model for Indoor Environments.....             | 144 |
| 5.2     | Simulation Setup .....                                   | 144 |
| 5.2.1   | General Simulation Workflow .....                        | 144 |
| 5.2.2   | Increasing Prediction Efficiency.....                    | 145 |
| 5.2.2.1 | When is a Channel Recalculation Necessary? .....         | 145 |
| 5.2.2.2 | Incremental Simulations .....                            | 146 |
| 5.2.2.3 | Hybrid Simulation Mode.....                              | 147 |
| 5.2.2.4 | Active and Inactive Transmitters .....                   | 147 |
| 5.2.3   | Configuring a Simulation.....                            | 147 |
| 5.2.3.1 | Angular Settings.....                                    | 151 |
| 5.2.3.2 | Channel Parameters.....                                  | 151 |
| 5.2.3.3 | Propagation Phenomena and Options .....                  | 152 |
| 5.2.3.4 | 2.5D and 3D Ray Tracing Options .....                    | 152 |
| 5.2.3.5 | Ray Splitting.....                                       | 153 |
| 5.2.3.6 | Miscellaneous.....                                       | 153 |
| 5.3     | Simulator Invocation.....                                | 154 |
| 5.3.1   | Choosing a Propagation Algorithm .....                   | 154 |
| 5.3.2   | Start of Simulation .....                                | 155 |
| 5.3.3   | Retrieving Results .....                                 | 156 |
| 5.3.4   | Stopping a Simulation .....                              | 156 |
| 5.4     | Parallelized Ray Tracing .....                           | 156 |
| 5.4.1   | Simulation Setup .....                                   | 157 |
| 5.4.1.1 | Name Server Setup .....                                  | 157 |
| 5.4.1.2 | Starting RPS Servers.....                                | 158 |
| 5.4.1.3 | Selecting RPS Servers .....                              | 159 |
| 5.4.1.4 | Using a Server Pool .....                                | 160 |
| 5.4.2   | Performing Parallelized Simulations.....                 | 161 |
| 5.4.2.1 | Simulator Invocation.....                                | 161 |
| 5.4.2.2 | Load Balancing.....                                      | 162 |

|         |   |     |
|---------|---|-----|
| 5.5     | Propagation Model Plug-ins .....                                  | 163 |
| 5.5.1   | Using Custom Propagation Models .....                             | 163 |
| 5.5.2   | Developing User-Defined Empirical Propagation Model Plug-ins..... | 164 |
| 5.5.2.1 | DLL Plug-in SDK for Custom Propagation Models.....                | 166 |
| 5.5.2.2 | MATLAB Plug-in SDK for Custom Propagation Models.....             | 167 |
| 5.6     | Dynamic Simulations .....   | 169 |
| 5.6.1   | Mobility Model .....  | 169 |
| 5.6.2   | Configuring a Dynamic Simulation.....                             | 171 |
| 5.6.3   | Performing a Dynamic Simulation.....                              | 173 |
| 5.6.4   | Storing Dynamic Channel Data.....                                 | 173 |
| 6       | Working with RPS — Result Analysis .....                          | 175 |
| 6.1     | Simulation Statistics .....                                       | 175 |
| 6.2     | Concept of Result Analysis.....                                   | 176 |
| 6.3     | Spatial Analysis .....  | 177 |
| 6.3.1   | Result Surface Plot Layers .....                                  | 178 |
| 6.3.2   | Received Power (Coverage) Surface Plots.....                      | 178 |
| 6.3.3   | Best Serving Cell Surface Plots .....                             | 179 |
| 6.3.4   | Signal-to-Interference Ratio (SIR) Surface Plots.....             | 180 |
| 6.3.5   | Delay Spread Surface Plots .....                                  | 180 |
| 6.3.6   | Angular Spread Surface Plots .....                                | 181 |
| 6.3.7   | Custom Surface Plots .....  | 182 |
| 6.3.8   | Contour Plots.....  | 182 |
| 6.3.9   | Coinciding Surface Plot Layers .....                              | 182 |
| 6.3.10  | Deriving Histograms from Surface Plots.....                       | 185 |
| 6.3.11  | Exporting Surface Plots Layers.....                               | 185 |
| 6.3.12  | Printing Surface Plots .....                                      | 186 |
| 6.4     | Surface Plot Plug-ins.....  | 186 |
| 6.4.1   | Using Custom Surface Plots .....                                  | 186 |
| 6.4.2   | Developing User-Defined Surface Plot Plug-ins .....               | 187 |
| 6.4.2.1 | DLL Plug-in SDK for Custom Surface Plots .....                    | 188 |
| 6.4.2.2 | MATLAB Plug-in SDK for Custom Surface Plots.....                  | 191 |
| 6.4.2.3 | Support Functions for MATLAB Custom Surface Plot Plug-ins .....   | 192 |
| 6.5     | Point Analysis .....  | 192 |
| 6.5.1   | Site-specific Channel Impulse Response .....                      | 193 |
| 6.5.2   | Site-specific Directions of Arrival .....                         | 194 |

|          |   |     |
|----------|---|-----|
| 6.6      | Path Analysis .....                                       | 195 |
| 6.6.1    | Received Power along a Path .....                         | 196 |
| 6.6.2    | Delay Spread along a Path.....                            | 197 |
| 6.7      | Transmitter Analysis.....                                 | 197 |
| 6.7.1    | LOS Preview Plot .....                                    | 198 |
| 6.7.2    | Statistical Analysis.....                                 | 198 |
| 6.7.2.1  | Antenna Tilt Analysis.....                                | 198 |
| 6.7.2.2  | Antenna Azimuth Angle Analysis .....                      | 200 |
| 6.7.2.3  | Transmit Power Analysis .....                             | 202 |
| 6.7.3    | DoA Analysis .....  | 203 |
| 6.8      | Post-Processing Plug-ins as Stand-Alone Applications..... | 203 |
| 6.8.1    | Using Custom Post-Processing Plug-ins .....               | 204 |
| 6.8.2    | Developing User-Defined Post-Processing Plug-ins.....     | 205 |
| 6.9      | Ray Path Analysis .....                                   | 206 |
| 6.9.1    | Inspecting Ray Paths .....                                | 206 |
| 6.9.2    | Inspecting Launched Rays .....                            | 207 |
| 6.9.3    | Inspecting Incident Rays.....                             | 208 |
| 6.10     | Handling of Chart Graphs, Histograms, and Tables .....    | 209 |
| 6.10.1   | Using Chart Graphs .....                                  | 209 |
| 6.10.1.1 | Chart Components.....                                     | 209 |
| 6.10.1.2 | Scaling and Zooming .....                                 | 210 |
| 6.10.1.3 | Creating a Table from a Graph .....                       | 210 |
| 6.10.1.4 | Exporting Chart Data .....                                | 211 |
| 6.10.1.5 | Printing a Chart Graph.....                               | 211 |
| 6.10.2   | Using Histograms.....                                     | 211 |
| 6.10.2.1 | Creating a Table from a Histogram.....                    | 212 |
| 6.10.2.2 | Customization of Histograms .....                         | 212 |
| 6.10.2.3 | Printing a Histogram .....                                | 213 |
| 6.10.3   | Using Tables.....   | 213 |
| 6.10.3.1 | Exporting Table Data .....                                | 214 |
| 6.10.3.2 | Statistical Data Evaluation in Tables.....                | 214 |
| 6.10.3.3 | Transforming Table Data into Graphical Presentations..... | 215 |
| 6.10.3.4 | Customization of Tables .....                             | 216 |
| 6.11     | Data Import and Export.....                               | 216 |
| 6.11.1   | Import of Antenna Diagrams.....                           | 216 |
| 6.11.2   | Import and Export of Environment Data .....               | 217 |

|          |  |     |
|----------|--|-----|
| 6.11.2.1 | Importing DXF/DWG Files.....                             | 217 |
| 6.11.2.2 | Import Environment Data from OBL Files .....             | 218 |
| 6.11.2.3 | Exporting Objects to DXF or DWG Files .....              | 219 |
| 6.11.2.4 | Exporting Objects to OBL Files.....                      | 219 |
| 6.11.3   | Exporting Simulation Results .....                       | 219 |
| 6.11.3.1 | Exporting Received Power and Delay Spread.....           | 219 |
| 6.11.3.2 | Exporting Data as an ASCII File .....                    | 220 |
| 6.11.3.3 | Exporting DoA as an ASCII File.....                      | 220 |
| 6.11.3.4 | Exporting Path Analysis Results into a MATLAB file ..... | 221 |
| 6.11.4   | Exporting Static Channel Data.....                       | 222 |
| 6.11.4.1 | Static Channel Data Export into ASCII Matrices.....      | 223 |
| 6.11.4.2 | Static Channel Data Export into a MATLAB File .....      | 224 |
| 6.11.4.3 | Data Post-Processing in MATLAB.....                      | 225 |
| 6.11.5   | Exporting Dynamic Channel Data.....                      | 226 |
| 6.11.5.1 | Dynamic Channel Data Export as ASCII Matrices .....      | 226 |
| 6.11.5.2 | Dynamic Channel Data Export into a MATLAB File .....     | 227 |
| 6.11.5.3 | Data Post-Processing in MATLAB.....                      | 228 |
| 7        | RPS COM Automation Interface .....                       | 229 |
| 7.1      | Capabilities using an Automation Interface .....         | 229 |
| 7.2      | COM Interface Specification.....                         | 230 |
| 7.3      | Automation Example.....                                  | 241 |
| 8        | Examples .....   | 245 |
| 8.1      | Antenna Diagrams .....                                   | 245 |
| 8.2      | Simple Indoor Environment .....                          | 245 |
| 8.3      | Pico Cellular Indoor Environment.....                    | 246 |
| 8.4      | Urban Outdoor Environment .....                          | 246 |
| 8.5      | Dresden City Center Environment .....                    | 247 |
| 8.6      | Suburban Outdoor Environment .....                       | 248 |
| 8.7      | Dynamic Outdoor Example .....                            | 248 |
| 8.8      | Dynamic Indoor Example .....                             | 249 |
| 9        | Support.....   | 251 |
| 9.1      | Electronic Support.....                                  | 251 |

|          |  |     |
|----------|--|-----|
| 9.2      | Telephone Support.....                     | 251 |
| 9.3      | Support Center .....                       | 251 |
| 9.4      | Feedback, Comments, Suggestions .....      | 251 |
| 9.5      | Contact Information.....                   | 252 |
| 9.6      | Third Party Components.....                | 252 |
| 10       | Appendix A.....                            | 253 |
| 10.1     | Window Controls Reference .....            | 253 |
| 10.1.1   | Menus .....                                | 253 |
| 10.1.2   | Toolbars.....                              | 257 |
| 10.1.2.1 | Standard Toolbar .....                     | 257 |
| 10.1.2.2 | Component Toolbar .....                    | 257 |
| 10.1.2.3 | Paint Toolbar .....                        | 258 |
| 10.2     | Environment Editor Reference .....         | 258 |
| 10.2.1   | String Command Reference.....              | 259 |
| 10.2.2   | Environment Editor Toolbars .....          | 260 |
| 10.3     | Stencil Library Overview .....             | 262 |
| 10.4     | RPS File Format.....                       | 263 |
| 10.5     | OBL File Format .....                      | 264 |
| 10.6     | Antenna File Format.....                   | 264 |
| 10.7     | Supported Raster Data File Formats .....   | 265 |
| 10.7.1   | TIFF / Geo TIFF Format .....               | 266 |
| 10.7.1.1 | ESRI World File Format.....                | 266 |
| 10.7.2   | BIL File Format.....                       | 266 |
| 10.7.3   | Arc/Info Raster Formats.....               | 266 |
| 10.7.3.1 | Arc/Info ASCII Interchange Format.....     | 266 |
| 10.7.3.2 | Arc/Info Binary Grid Format.....           | 267 |
| 10.7.4   | EOSAT Fast Format .....                    | 267 |
| 10.7.5   | Erdas Imagine Format .....                 | 267 |
| 10.7.6   | GIF File Format .....                      | 267 |
| 10.7.7   | Grid eXchange File Format .....            | 267 |
| 10.7.8   | Hierarchical Data Format (Release 4) ..... | 267 |
| 10.7.9   | Japanese DEM Format.....                   | 267 |
| 10.7.10  | JPEG File Format.....                      | 268 |
| 10.7.11  | Atlantis MFF Raster File Format .....      | 268 |

|         |   |     |
|---------|---|-----|
| 10.7.12 | PCI Labeled Raw Format.....                       | 268 |
| 10.7.13 | Portable Network Graphics .....                   | 268 |
| 10.7.14 | USGS DOQ Format.....                              | 268 |
| 10.7.15 | USGS SDTS DEM Format.....                         | 269 |
| 10.7.16 | X11 Pixmap Format.....                            | 269 |
| 11      | Appendix B.....                                   | 271 |
| 11.1    | Dielectric Properties of Selected Materials ..... | 271 |
| 11.2    | Notational Conventions.....                       | 272 |
| 11.3    | Abbreviations and Acronyms.....                   | 272 |
| 11.4    | References .....                                  | 273 |
| 12      | Index.....  | 275 |



## 1 Overview

### 1.1 Introduction

RPS, the Radiowave Propagation Simulator, is a radio coverage/performance planning system for a variety of radio systems. Outstanding characteristics of the RPS platform are:

- state-of-the-art graphical user interface (GUI) with extensive analysis and presentation functions
- very fast and accurate 3D ray tracing as well as empirical propagation algorithms
- highly parallelized ray tracing engine with sophisticated load balancing and hybrid prediction mode for unprecedented simulation performance
- open system architecture due to
  - various data import/export interfaces for network configuration, environment, and network performance data
  - propagation/post-processing plug-in technology for user-defined algorithms
  - COM interface for remote application control and planning system integration

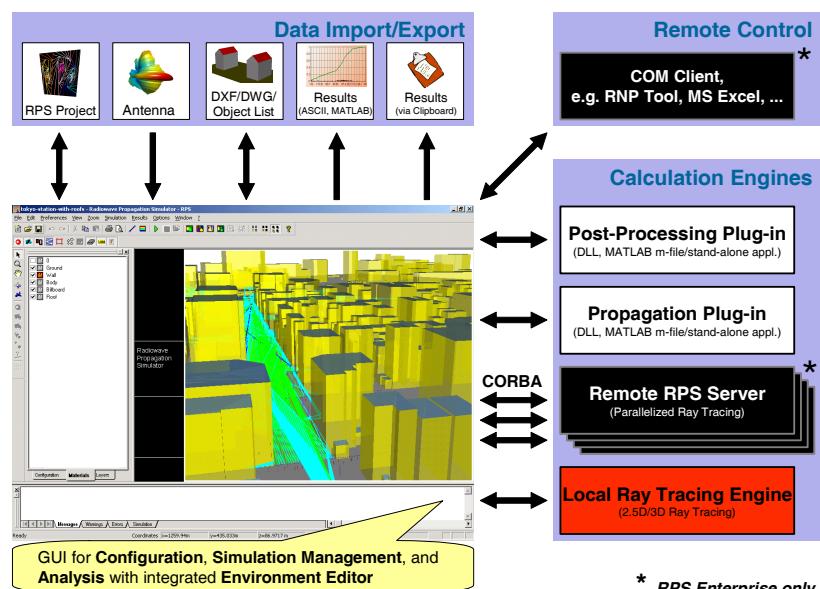


Fig. 1-1 RPS overview

RPS is delivered in two versions:

- **RPS Enterprise:** This version is mainly suited for larger organizations, such as network operators, service providers, or system/equipment manufacturers, with high requirements on performance and accuracy of planning data and extensive computational resources available. RPS Enterprise represents a very efficient planning system for a variety of wireless technologies due to its unprecedented performance, highly parallelized ray tracing engine as well as easy integration possibilities by means of its COM interface.
- **RPS Professional:** This version is suited for smaller organizations, such as local carriers, network maintenance companies, consultancies, as well as academia. RPS Professional embraces the same functional range for radio network planning and coverage prediction; however, as a stand-alone application, it does not offer parallelized simulation and planning system integration support.

## 1.2 Workflow Model

Fig. 1-2 shows the general RPS workflow model:

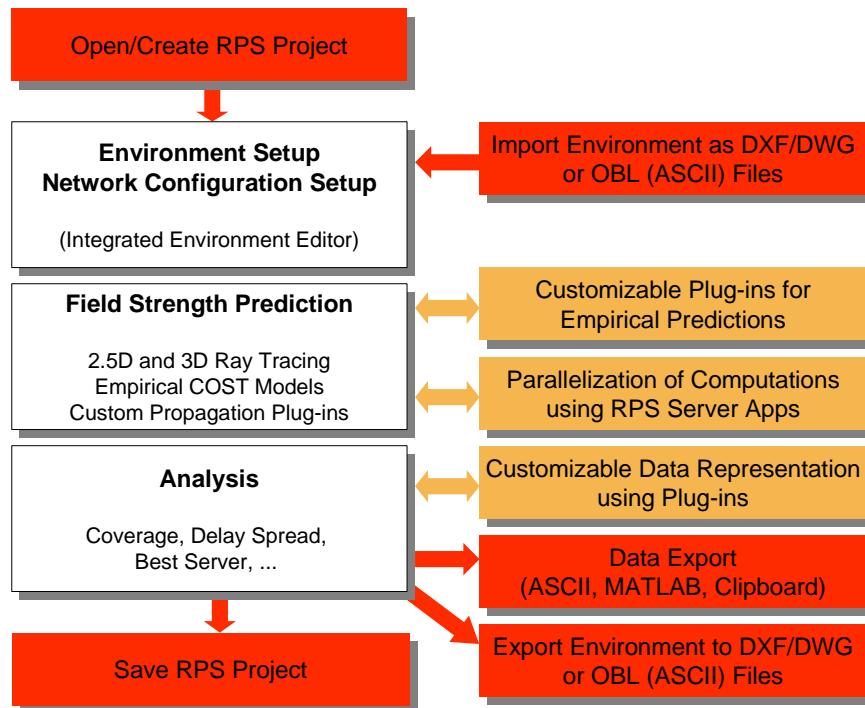


Fig. 1-2 RPS workflow model

First, and before a simulation can be performed, an environment has to be generated using the integrated environment editor. Likewise, an existing environment database can be opened, or can be imported from DXF/DWG or tab-delimited ASCII files (object list).

Next, the radio network is configured by placing transmitters at selected base station positions, whereas receivers are usually set in a matrix or along a line at places where channel data shall be obtained. Then, a network simulation can be performed. The user can choose between built-in ray tracing algorithms and empirical propagation models. In addition, custom propagation models can be applied using the RPS plug-in technology. The simulations can be performed on a local machine, or distributed using a theoretically unlimited number of workstations connected via a TCP/IP network.

The channel characteristics can be analyzed and post-processed in a following step. Customizable analysis functions can be implemented using plug-ins. The simulation results can be stored or exported in tab-delimited ASCII files or MATLAB files for further use.

The plug-in technology enables the customization and extension of RPS towards a comprehensive network planning platform with flexible and unlimited analysis opportunities. Network-specific performance indicators can easily be derived from the radio channel data. At the same time, RPS provides the framework to conveniently present and analyze such secondary results.

### **1.3 Summary of Main Functions and Modules**

#### **1.3.1 Field Strength Prediction Algorithms**

- Field strength prediction and channel determination in the micro- and mm-wave range (300 MHz to 300 GHz) using built-in ray tracing and empirical prediction models
- The required computational time can be reduced by up to a factor of 50 compared with common 3D ray tracing propagation models using 2.5D ray tracing algorithms.
- Simulations of unprecedented performance can be conducted by distributing them on numerous RPS Servers running under Windows NT or Linux workstations connected to the Windows client via a TCP/IP network and a CORBA interface.
- Specific simulation algorithms supporting multi-processor systems under Windows NT 4.0, Linux 2.x or higher.
- Consideration of complex radiation patterns  $C_{E,S}(\theta, \varphi)$  and polarization  $\vec{p}$  (linear and circular) from transmitting and receiving antennas
- Investigation of depolarization effects due to full polarimetric calculation of wave propagation using the built-in ray tracing algorithms
- Storage of outgoing angles of rays transmitted from a base station and angles of arrival (AOA) at receivers for the analysis of Space Division Multiple Access Systems (SDMA Systems)
- Smart database algorithms: RPS can help to perform wave propagation calculations in a short time also in environments with thousands of buildings and receiver positions.

#### **1.3.2 Graphical User Interface**

- An integrated environment editor with AutoCAD-like usage allows for an effective construction and modification of the geometrical environment. Moreover, the network configuration (transmitters and receivers) can be set up easily.
- Enhanced 3D view with zoom, rotation, “fly-by” mode, and intuitive key navigation enables a detailed observation of local effects.
- As a background image e.g. a map of a city or a building construction plan can be displayed which simplifies the construction of building structures considerably.
- Sophisticated analysis and presentation functions enable an efficient and flexible performance assessment of a variety of radio technologies.
- Graphical surface plots allow for a true-to-scale representation of results.

- Changing of transmitter power, radiation pattern, polarization of receivers etc. is possible without any recalculation of the radio channel.

### 1.3.3 Open Simulation System

- Plug-in technology allows user-defined extensions for empirical and semi-empirical models and the presentation of customized data directly within the RPS framework.
- Interaction with the MATLAB Engine allows the usage of MATLAB scripts as plug-ins; scripts can be modified directly in the RPS framework.
- Dynamic extensions are incorporated including a mobility model for generation of dynamic channel data.
- A COM automation interface supports the remote control of RPS from any Windows application that supports COM.
- DXF/DWG import and export filters including layer support to exchange building data to and from RPS with other CAD based applications
- The MATLAB interface allows export of channel data including angular information for data analysis and system simulations.

## 1.4 New Features in Version 5.3

### 1.4.1 Functionality

A new *satellite mode* was introduced for modeling transmitters with very high altitudes.

The *diffraction edge detection* was *revised*. Edges where more than two polygons coincide are no longer detected as diffraction edges.

In multi-story buildings, the *altitude adjustment* for transmitters and receivers can be done *relative to the floor heights*. Thus these network elements can e.g. be easily placed on different floors at the same x/y positions.

The *coincide layers function* was *extended*. Now it is possible to apply a threshold check between two layers.

The *CORBA communication* in parallel simulations was further *improved*.

### 1.4.2 Convenience of Usage

The concept of a *user-defined workspace* was introduced. The workspace saves all personal display and layer presentation settings of the user. Furthermore, global and personal *legend presets* (defining color palettes, value ranges, transparency, etc.) can be handled in order to be applied for any kind of layer.

*Material properties* from the material database can now be *directly assigned to material layers* in the material layer overview dialog.

Each *transmitter* now has a *visibility flag*. It can e.g. be used to hide transmitters from other floors in a 2D view of a multi-story building. Furthermore, each transmitter can be assigned a *dedicated color* which is used for the Best Server Plot.

### 1.4.3 Bugfixes

When applying strong downtilt angles to an antenna, the pattern could be slightly distorted. Now antenna patterns can be freely rotated without distortion.

The OBL parser could only parse up to 4095 characters per line. So for very long lines (e.g. polygons with many vertices), the file could not be properly loaded. This restriction was relaxed.

If RPS file was opened with a double click, a possible raster DEM was not loaded. Now this case is handled in the same way as if the project was opened from within RPS.

RPS could crash when a block was created in the Environment Editor with a name that was already used for another block. This case is caught now correctly.

If only a raster DEM, but no polygons were defined in an environment, RPS could crash during ray tracing. This case is now handled properly.

## 1.5 New Features in Version 5.2

### 1.5.1 Functionality

The usage of *floating licenses* is enabled now in conjunction with hardware dongles.

For surface plots, the *legend settings* are now *persistently stored* in a `legend.ini` file. Moreover, a new *angular spread* standard surface plot was introduced. In the *LOS preview plot* the configured noise floor is additionally considered.

Each *transmitter* now has an *individual carrier frequency* which allows for investigations with several concurrent systems. The Tx overview dialog was enhanced by functions for *duplicating* and *deleting* a transmitter. Transmitters can now also be *placed above buildings* (i.e. above DTM) in addition to the DEM layer.

The *height setting of receivers* based on DEM has been *improved* especially for rough DEM layers. Receivers below non-vertical objects (e.g. roofs) can be automatically removed.

The *Environment Editor* was enhanced by several *new functions*:

- A new command `3DPOLYGON` was added.
- Material layers have an additional description field.
- The Environment Editor provides printing support.

Results produced by the *path analysis* can now directly be *exported to a MATLAB file*.

The *COM interface* was *extended*.

### 1.5.2 Convenience of Usage

The propagation algorithm can now be *selected* in a *list field* in the standard toolbar. So the currently chosen algorithm is always visible in the GUI. Additionally, the choice of the propagation algorithm is now *project specific*.

*Ray splitting* can now be configured by a *maximal ray tube size* rather than a splitting interval in order to ease the understanding of this parameter.

The 3D view shows a *navigation compass* in order to ease the orientation while moving through the 3D environment.

The *<Shift>* key is now generally supported to create a *new separate layer* of the same type.

### 1.5.3 Bugfixes

RPS now works generally with real powers – the unit selection between dB/dBm is not necessary any more.

The antenna interpolation has been improved to more realistically imitate a 3-dimensional antenna pattern from the two, horizontal and vertical, cuts.

When a simulation was cancelled, all active transmitters were set obsolete which could lead to undesired erasure of results. Now only the those transmitters are set obsolete on simulation interruption that were not finished yet. The propagation results from completely calculated transmitters is thus correctly kept.

In the path analysis, deactivated transmitters could cause an error message “An even number of data items required”. Now this inapplicable message is suppressed.

The OBL import was corrected to allow vertical polygons being correctly decomposed into triangles and to always assign the right material layers to all polygons.

## 1.6 Upgrade

Since the RPS prediction algorithms were updated, it is necessary to use the latest release for all RPS Server applications running on the network. RPS 5.3 works only with RPS Servers Version 5.3 (RPS Enterprise only).

The RPS file format has been enhanced to support the new features of RPS 5 and higher. Older files will be automatically converted.

## 1.7 System Requirements

For the work with RPS the following system parameters are necessary or recommended:

- Windows NT/2000/XP
- Pentium-class processor
- 256 MB RAM (512 MB – 1 GB for large environments)
- 100 MB hard disk space (1 GB recommended)
- A graphics resolution with at least 65,536 colors and OpenGL support
- CD-ROM drive (for installation)
- USB port (in case of hardware dongle software protection)
- 3.5" floppy disk drive (in case of license file software protection)

- TCP/IP network (RPS Enterprise only)
- The MATLAB R11, R12, R12.1, or R13 software package is required for m-file based plug-ins or MATLAB stand-alone application plug-ins. MATLAB must be purchased separately from The Mathworks Inc. ([www.mathworks.com](http://www.mathworks.com)) or through a local vendor.

The size of required memory depends significantly on the problems to be solved with RPS. For the calculation of the electromagnetic field distribution in micro cells with thousands of obstacles the memory mentioned above should be sufficient. Hints to save the memory resources are given in the corresponding sections in this user guide. Also the usage of multi-processor systems (MPS) is supported by RPS under Windows NT/2000/XP environments due to parallelized ray tracing algorithms.

RPS Enterprise contains RPS Servers for the Windows and the Intel based Linux operating system (Kernel 2.x or higher) for distributed computing in a TCP/IP network. RPS communicates internally with the RPS Servers via a CORBA interface.

## 1.8 Installation

### 1.8.1 RPS Software Package

The RPS software comprises:

- the RPS graphical user interface with integrated simulation engine, a multi-threaded application that manages all configuration data, results, and the simulations
- the RPS Server applications, that allow a distributed computing across a TCP/IP network (RPS Enterprise only)
- a User Manual
- several example projects incl. a COM example (the latter one in RPS Enterprise only)
- several plug-in examples for customized propagation models and user-defined surface plots
- a library with antenna diagrams
- a stencil library of frequently used building blocks for the environment
- a material database

The file structure on the installation CD-ROM is given in Table 1-1.

Table 1-1. Directories on the installation CD-ROM

| <i>Directory</i>     | <i>Description</i>  |
|----------------------|---|
| corba\OmniNamesWIN32 | The name server of the CORBA implementation omninames (RPS Enterprise only)           |
| doc                  | This documentation in Portable Document Format (PDF)                                  |
| drivers              | Diverse hardware drivers  |
| examples             | The examples delivered with RPS, including background images for the outdoor examples |

| <i>Directory</i>    | <i>Description</i>   |
|---------------------|--|
| examples\comexample | An example that demonstrates the COM interface using MS Excel as client (RPS Enterprise only)                                |
| matlab_scripts      | MATLAB scripts as examples for data post-processing using MATLAB   |
| rpsserver\LINUX2    | The RPS-Server application for Linux from Kernel 2.x (RPS Enterprise only)   |
| rpsserver\WIN32     | The RPS-Server application for WIN32 platforms (RPS Enterprise only)   |
| rupdate             | Feature update application for hardware dongles  |
| sdk                 | Templates and sample projects for the development of user-defined plug-ins using shared libraries compiled with Visual C++ 6 |

### 1.8.2 RPS Installation Process

The software must be installed comparable to any other Windows installation. The setup program performs all necessary operations concerning the installation and registration of the software package. The required files are stored on the CD-ROM.

#### Important Notice for Hardware Dongle Usage:

If your RPS installation is delivered with a hardware dongle, make sure that the dongle is *not* inserted before the entire installation procedure has been finished! This will ensure that the USB port driver for the hardware dongle is installed correctly *before* the dongle is inserted for the first time.

The installation of the driver for the hardware dongle requires *administrator rights* on your computer. Make sure that you have those rights when you install RPS with hardware dongle support.

To install the software on your computer (Windows NT/2000/XP platform), insert the CD-ROM into the CD-ROM drive. The setup program is automatically started. The Windows Installer is configured and the setup procedure is started.

If a RPS of the same version is already installed on this computer, a dialog appears that allows to

- modify the current installation, for instance to add/remove features
- repair the current installation, or
- remove (uninstall) the current installation.



Fig. 1-3 Program maintenance dialog

In the following, the steps required to accomplish a new installation are described. First, the license agreement is shown. Please, read it carefully to get familiar with the license terms. If you do not accept it, you cannot install the software on your computer. To accept the license conditions, select the radio button I accept the terms in the license agreement and click Next.

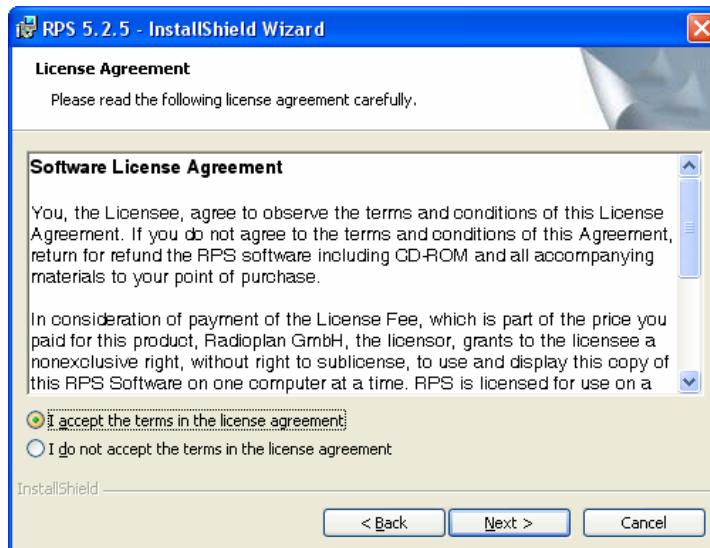


Fig. 1-4 License agreement dialog

In the customer information dialog one can choose whether the application shall be installed available for all users or only for the current user. It is recommended to use the default settings.

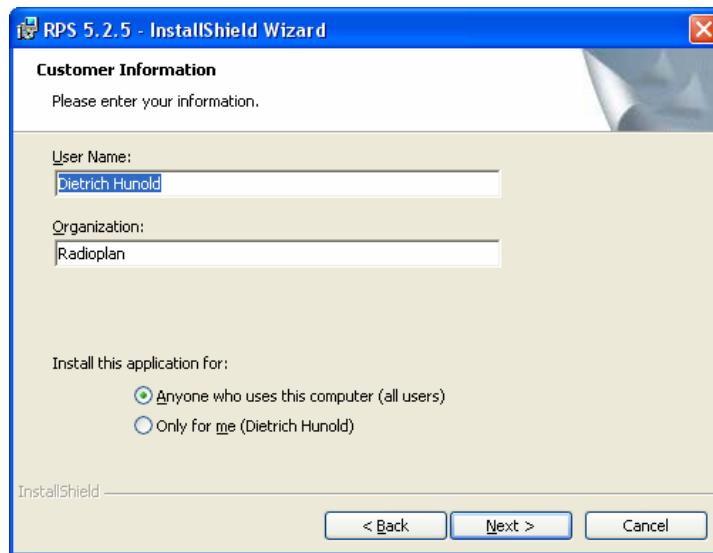


Fig. 1-5 Customer information dialog

In the next dialog one can decide whether to install the complete software package or a customized version. Using the latter option, the available features are shown in a hierarchical structure. Some of the RPS core components are required and must be installed. It is always possible to install additional features at a later time by running the setup procedure again. For the features to be installed a custom setup directory can be chosen using the Change... button.

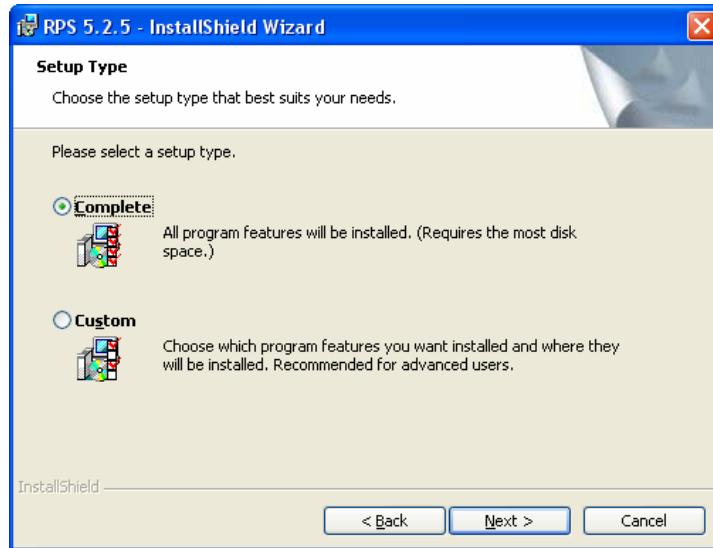


Fig. 1-6 Custom setup dialog

Next, the setup procedure is ready to install. After confirmation, optionally old RPS installations will be uninstalled, and the selected features are installed.

Furthermore, the USB port driver for the hardware dongle support is automatically installed in the background. Make sure that the hardware dongle is not inserted during the entire installation procedure!

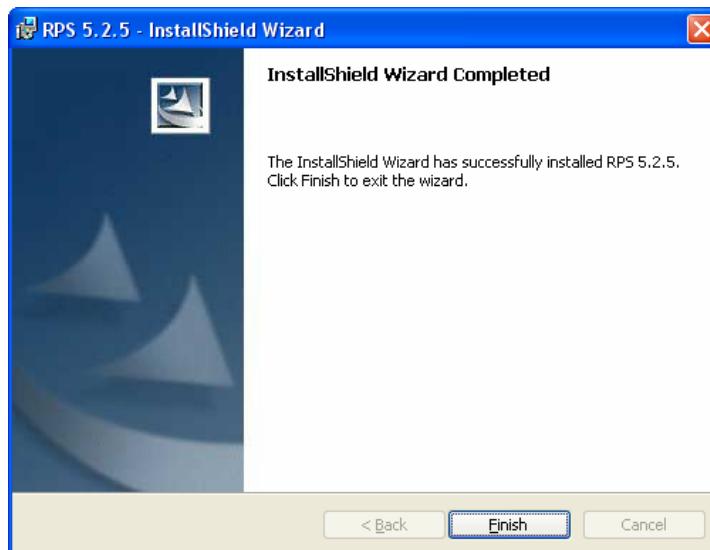


Fig. 1-7 Installation completion dialog

A new menu folder RPS is created in the Windows start menu that contains some icons to start the RPS 5 application or to open one of the documentations installed with RPS.

In order to uninstall the RPS software, to install any other feature, or to repair the installation, just run the setup program again.

If you use an RPS installation with hardware dongle software protection, the according driver has been installed silently during the installation process. This could only be done if you have administrator rights on your computer. You can insert the dongle into the USB port now which will activate the driver for that port. In case your operating system is Windows XP, a message window similar to the one in Fig. 1-8 could appear. You can ignore this message and continue with the hardware setup as usual.



Fig. 1-8 Windows XP message during hardware dongle driver setup

### 1.8.3 Directory Structure

During the installation procedure, the complete RPS package has been installed into the %ApplicationPath% (commonly c:\program files\RPS). The directory structure under this directory is shown in Table 1-2.

Table 1-2. Structure of the RPS root directory

| <i>Subdirectory</i>  | <i>Description</i>   |
|----------------------|--|
| CORBA\OmniNamesWIN32 | The name server of the CORBA implementation omninames (RPS Enterprise only)  |
| doc                  | This documentation in Portable Document Format (PDF)   |
| drivers              | Diverse hardware drivers   |
| examples             | The examples delivered with RPS, including background images for the outdoor examples  |
| examples\comexample  | An example that demonstrates the COM interface using MS Excel as client (RPS Enterprise only)                                |
| matlab_scripts       | MATLAB scripts as examples for data post-processing using MATLAB   |
| RPS                  | RPS application directory  |
| RPS\library          | All library items (antennas, stencils) are located here.   |
| RPS\MatlabRxx        | MATLAB specific RPS DLLs for different versions of MATLAB  |
| RPS\plugins          | The plug-in directory for customized propagation models and user-defined post-processing functions                           |
| rpsserver\LINUX2     | The RPS-Server application for Linux from Kernel 2.x (RPS Enterprise only)   |
| rpsserver\WIN32      | The RPS-Server application for WIN32 platforms (RPS Enterprise only)   |
| rupdate              | Feature update application for hardware dongles  |
| sdk                  | Templates and sample projects for the development of user-defined plug-ins using shared libraries compiled with Visual C++ 6 |

#### 1.8.4 RPS Customization

In the %ApplicationPath% directory there are several \* .ini files that contain application parameters and, thus, enable a customization of RPS. Normally such settings are configured only during the installation process. However, if necessary they could also be adjusted at any given time.

All \* .ini files usually consist of several sections that are prefaced by a specifier enclosed in brackets. Then in each line a parameter with associated value is given. Comment lines can also be inserted, starting with a semicolon. The following subsections give explanations on the most important of the initialization files.

##### 1.8.4.1 Global RPS Settings

The rps .ini file is used to adjust global settings for RPS. It contains two sections for RPS itself and MATLAB plug-ins as demonstrated in this example:

```
[RPS]
EnableBmpMenus = 1
;LanguageDLL = "RPSJapanese.dll"

; --giopMaxMsgSize
:
; maximum message size in bytes of one CORBA block, default value is
2097152
giopMaxMsgSize = 20000000
```

```

;clientCallTimeOutPeriod = 10000
; --AdvancedSchedulingInterval in ms:
; --sleep time between two cycles of load balancing
AdvancedSchedulingInterval = 1000

; --AdvancedSchedulingMode :
; --0: In each load balancing cycle highest loaded server send rays to
lowest loaded server,
; -- second highest to second lowest until difference between high and
lowly loaded server is < 10 rays
; --1: Only idle (empty) servers will receive rays from highly loaded
servers
; --2: Only servers loaded with max. 50 rays will receive rays from highly
loaded servers
AdvancedSchedulingMode = 0

; --AdvancedSchedulingTransferredRayCountMin :
; --Minimum number of required rays to be transferred that load balancing
is accepted
AdvancedSchedulingTransferredRayCountMin = 10

; --AdvancedSchedulingTransferredRayCountMax :
; --Max number of rays that will be transferred from high to low loaded
server
AdvancedSchedulingTransferredRayCountMax = 250

; --SendEnvironmentDataServerCountMax :
; --Max. number of servers that can receive environment data in parallel
; --Transferring data to too many servers can lead to bad CORBA performance
SendEnvironmentDataServerCountMax = 5

; --ReceiveResultsServerCountMax :
; --Max. number of servers that can send results data in parallel
; --Transferring data to too many servers can lead to bad CORBA performance
ReceiveResultsServerCountMax = 5

WorkerThreadPriority = -1

DefaultFont = "Arial"
;DefaultFont = "MS Mincho"
DefaultMessageWindowFont = "Courier New"
;DefaultMessageWindowFont = "MS UI Gothic"
DefaultCharSet = 0

DataFormat = "VDF"

[Matlab]
Release = 13
LibPath = "%ApplicationPath%\matlab_library"

```

In some entries the string “%ApplicationPath%” can be used as a placeholder for the root directory where RPS was installed in. Table 1-3 gives an overview of the application parameters in the rps.ini file.

Table 1-3 Application parameters of rps.ini

| <i>Parameter</i>                         | <i>Description</i>   |
|--|--|
| <b>[RPS]</b>                             |  |
| EnableBmpMenus                           | This flag (0/1) enables bitmap menus. On some Windows NT 4.0 Terminal Server Editions, this flag must be 0 to avoid incorrectly appearing menus (this problem was reported to the provider of the respective software development library) until this problem will be solved.  |
| LanguageDLL                              | This variable defines the name of an optionally used DLL with language specific string tables. By defining a certain language DLL, RPS can appear in different languages.  |
| giopMaxMsgSize                           | The maximum message size in bytes of one CORBA block, default value is 2,097,152 (corresponding to 2MB). It is recommended to choose a larger value in order to avoid some loss of data during communication between client and server.  |
| AdvancedSchedulingInterval               | The sleep time between two cycles for the advanced scheduling algorithm in ms. Please do not modify this value!  |
| clientCallTimeOutPeriod                  | The maximal server response time. If this period is exceeded a communication error occurs.   |
| AdvancedSchedulingMode                   | One of three modes are possible, each of them represented by an integer:<br>0: In each load balancing cycle, the highest loaded server sends rays to the lowest loaded server, the second highest to the second lowest, etc., until the difference between highly and lowly loaded servers is less than 10 rays.<br>1: Only idle (empty) servers will receive rays from highly loaded servers.<br>2: Only servers loaded with max. 50 rays will receive rays from highly loaded servers. |
| AdvancedSchedulingTransferredRayCountMin | Minimum required number of rays to be transferred during load balancing between RPS servers.   |
| AdvancedSchedulingTransferredRayCountMax | Maximum number of rays to be transferred from a highly loaded to a lowly loaded RPS server during load balancing.  |
| SendEnvironmentDataServerCountMax        | Max. number of servers that can receive environment data in parallel. If data is transferred to too many servers at the same time, the CORBA throughput could suffer, possibly leading to communication errors.  |
| ReceiveResultsServerCountMax             | Max. number of servers that can send results data in parallel. If data is transferred to too many servers at the same time, the CORBA throughput could suffer, possibly leading to communication errors.   |
| WorkerThreadPriority                     | An integer giving the process priority for the ray tracing threads. It is recommended to assign a lower priority than normal (i.e. values below 0, typically -1) in order to not affect the performance of the GUI.  |

| Parameter                | Description  |
|--------------------------|--|
| DefaultFont              | Defines the default font used by the graphical user interface of RPS. It could be necessary to adjust this font for different languages.   |
| DefaultMessageWindowFont | Defines the default font used for the message window in the lower part of the application window. It could be necessary to adjust this font for different languages.   |
| DataFormat               | The data format for environment data to be saved with an RPS project. Possible formats are "VDF" and "DWG". It is recommended to use "VDF" in favor of "DWG" especially for large environments in order to achieve an optimum application performance. |
| <b>[Matlab]</b>          |  |
| Release                  | The MATLAB version that RPS is linked with. Possible values are 11, 12, 12.1, and 13.  |
| LibPath                  | An additional library path for MATLAB helper files. Some incorporated MATLAB m-file plug-ins require certain helper files that are located in this library path.   |

Due to significant differences and binary incompatibilities of different MATLAB versions (releases R11, R12, R12.1, and R13), RPS is linked against the libraries of all those releases separately. One can modify the MATLAB version used by editing the parameter 'Release' in the [Matlab] section of the `rps.ini` file. Please, make sure that the configured value is compatible with the MATLAB version you use.

Note that the MATLAB stand-alone plug-in examples that are delivered with RPS were compiled with the latest MATLAB edition (R13). Moreover, the MATLAB runtime library is only installed for R13 by default. If a different MATLAB version shall be used, one can copy the appropriate files from the RPS CD-ROM into the respective subdirectory.

#### 1.8.4.2 License Management Configuration

The usage of hardware dongles or license files for software protection is configured by the parameters in the `license.ini` file in the `%ApplicationPath%` directory. A sample file is given below, the explanations of the parameters follow in Table 1-4.

```

[License]
;License = 0 : HW protection
;License = 2 : SW protection
License = 0

;Port = USB : for local USB key (default setting)
;Port = TCP_IP : for remote connection via TCP/IP
Port = "USB"

;Server = USB : for local USB key (default setting)
;Server = CBNetServerDNSName : for remote connection (or IP address)
Server = "USB"

```

Table 1-4 License management parameters in license.ini

| Parameter        | Description  |
|------------------|--|
| <b>[License]</b> |  |
| License          | A discriminator to either use hardware dongles (0) or license files (2) for software protection. If license files are chosen, then the other parameters Port and Server are ignored.   |
| Port             | Specifies the port/connection where the hardware dongle is to be expected. If single computer licenses are used, then the dongle must be inserted into the USB port of the respective computer and thus the value must be "USB". For floating licenses, however, the value must be "TCP_IP" because the connection to the license server is realized via TCP/IP. |
| Server           | Specifies the license server for floating licenses. Since in the case of single computer licenses there is no license server, the value must be "USB" in this case. Otherwise, it must denote the DNS name of the license server in the network or, alternatively, its IP address.   |

#### 1.8.4.3 Radio System Default Settings

Some basic settings of certain radio systems are pre-configured in the RPSFrequencyTemplates.ini file. This file contains a line for each system, giving the basic parameters as tab-delimited values. A system template can be selected in the preferences dialog in order to determine appropriate settings for the propagation prediction. See Section 5.2.3 for details.

#### 1.8.4.4 Transmitter Default Settings

General default settings of each transmitter are pre-configured in the RPSDefaultParameters.ini file. This file contains two groups, specifying parameters for the transmitter hardware and some general network parameters. These values can later be modified in the transmitter settings dialog. See Section 4.2.2 for details.

#### 1.8.4.5 Material Database

A material database is provided in the MaterialList.ini file in the %ApplicationPath%. It contains a list of material property definitions in a tab-delimited ASCII format.

#### 1.8.4.6 MATLAB Syntax Highlighting Conventions

The MatlabEdit.ini file determines the MATLAB specific syntax highlighting of the integrated script editor. Please do not change these settings.

#### 1.8.4.7 Geographical Coordinate System Settings

The wkt.ini file contains the definitions of all available coordinate system settings. See Section 3.2 for more details on coordinate systems in RPS.

### 1.8.5 RPS Server Installation

RPS Enterprise allows for distributed coverage simulations thanks to highly parallelized ray tracing algorithms that can increase the computation speed dramatically. In order to take full advantage of an RPS Enterprise version, all intended RPS Server workstations must have access to the RPS installation directory which is usually located on the client computer.

The server startup script (batch file or shell script, respectively) must still be modified for the hostname of the computer where the CORBA name server will run. Usually this is the client computer. The corresponding startup script under the `rps\rpsserver\` subdirectory contains only a single line:

```
rpsserver.exe %COMPUTERNAME% -ORBInitRef NameService=corbaname::<hostname>
```

Please change the `<hostname>` placeholder to the actual used one of the CORBA name server. The default port number is 2809. If a different port number shall be used, then the valid format of the hostname must be:

```
<hostname>:<portnumber>
```

The hostname placeholder can also contain name spaces, delimited by dots. More details about the RPS Server setup are given in Section 5.4.

## 1.9 Software Registration

RPS has a built-in software protection function to avoid illegal copies of registered products. Two different options can be used: either hardware dongles or software license files. In the case of hardware dongles, also floating licenses can be supported. In the following, these options are explained in more detail.

### 1.9.1 Hardware Dongles

RPS can be delivered with hardware dongle protection by MARX Datentechnik GmbH. These security devices can be plugged into the USB port of a computer. They support the following two options:

- **Single License:** The RPS software can only be used on that computer where the hardware dongle is inserted. In case of parallelized simulations distributed in a LAN, the hardware dongle must be inserted in the client computer.
- **Floating License:** A single hardware dongle controls the software usage in a computer network. The RPS software can be used concurrently on any computer in the network up to the number of purchased floating licenses.

#### Important Notice for the Installation:

Make sure that the hardware dongle is *not* inserted before the entire installation procedure has been finished! This will ensure that the USB port driver for the dongle is installed correctly *before* it is inserted into your computer for the first time.

#### 1.9.1.1 Single Computer License

When using a hardware dongle for a single computer license, the dongle must be inserted into the USB port of that computer where RPS is used. In case of parallelized simulations distributed in a LAN, the hardware dongle must be inserted in the client computer. As long as RPS is running the dongle must be kept inserted; otherwise the application will be closed automatically. So in this case, the number of purchased RPS licenses is equal to the number of hardware dongles shipped.

In order to use a hardware dongle for license control on a single computer, make sure to have the following settings in the `license.ini` file which is located in the `%ApplicationPath%` directory (see also Section 1.8.4.2):

```
[License]
License = 0
Port = "USB"
Server = "USB"
```

### 1.9.1.2 Floating License Management

In order to enable the usage of floating licenses in a computer network, one computer has to be selected as the license server. All floating licenses are then controlled by a single hardware dongle that is inserted into the license server. Furthermore, the `CbNetSrv.exe` application from the `%ApplicationPath%\CbNet Server\` directory must be started on the license server. Then a icon is shown in task bar – a double click on it opens a window showing the current license log file. Additionally, the license server can be stopped and started again by pressing the according button in this window.

Along with the server application, a comfortable floating license administration tool `AdminApp.exe` is provided in the `%ApplicationPath%\CbNet Server\` directory. This tool can be used to

- administer the license server (restart, shut down);
- administer the license usage (list all users, disconnect single users).

After starting the administration tool, the main window as given in Fig. 1-9 is shown. First, all license servers (typically only one) must be searched in the network by pressing the Get/Update Servers button. All servers found are listed in the field to the right. Then the license administrator can log in to the currently selected server by pressing the other button Login To Server. The administrator password is needed for this. A new window is opened that allows all administrative actions to be done.

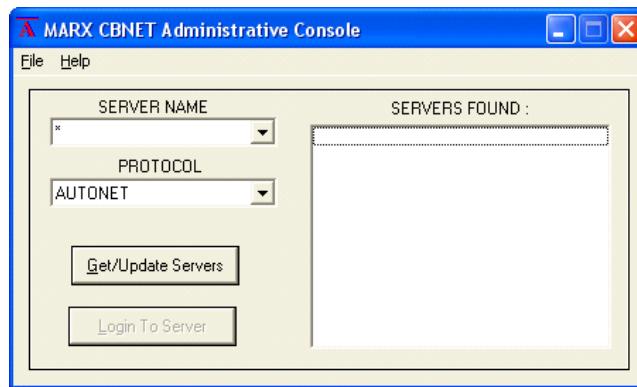


Fig. 1-9 Main license administration tool window

For administrative purposes, also a web-based frontend is available that offers the same functionality as the license administration tool. It can be accessed by selecting the URL "`http://<LicenseServerName>:8756`" in a normal web browser where `<LicenseServerName>` is the hostname of the license server or, alternatively, its IP address, and 8765 is the default TCP/IP port number. Read the

%ApplicationPath%\CbNet Server\server.htm file for more information on the floating license administration.

When using floating licenses, the RPS application must be installed on every computer where it shall be used. Additionally, it must be ensured that the following settings are valid in the license.ini file in the %ApplicationPath% directory on every computer where RPS is installed (see Section 1.8.4.2):

```
[License]
License = 0
Port = "TCP_IP"
Server = "<LicenseServerName>"
```

The string <LicenseServerName> can either be the hostname of the license server or, alternatively, its IP address.

If the hardware dongle is properly inserted into the license server and the license server application is running, RPS can be used throughout the network utilizing the floating licenses. The hardware dongle controls the number of concurrent instances of RPS that are simultaneously running.

#### 1.9.1.3 Feature Updates for Hardware Dongles

If RPS is protected by a hardware dongle, all the necessary license information is retrieved from this USB hardware security device. When a customer licenses additional software modules – also for a limited time period – these modules can be activated remotely by reprogramming the hardware dongle. This section describes how the license information can be updated at the end user's side without sending the USB dongle to Radioplan.

The customer generates a transaction file that will be generated for the currently connected hardware dongle first. This file must be sent to Radioplan, and in return an activation file is submitted to the end user. Using this activation file, the customer can update the respective hardware dongle finally as depicted in Fig. 1-10.

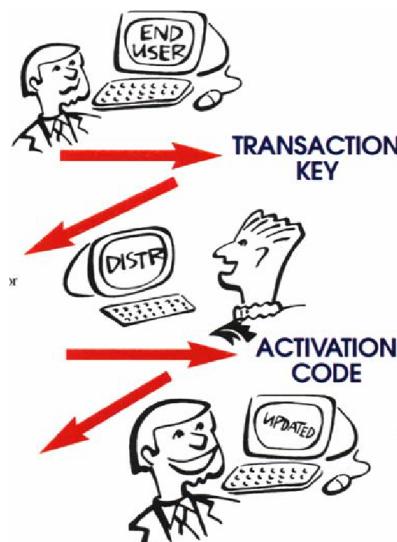


Fig. 1-10 General procedure of the hardware key update process

The necessary steps for creating a transaction key and updating the hardware security device using an activation code provided by Radioplan is described in detail now.

The transaction necessary to start a license update can be generated with the utility `rupdate.exe`. This application was installed together with RPS and it can be found in the `%ApplicationPath%\rupdate` subdirectory. The application window of the Remote Update utility is shown in Fig. 1-11.

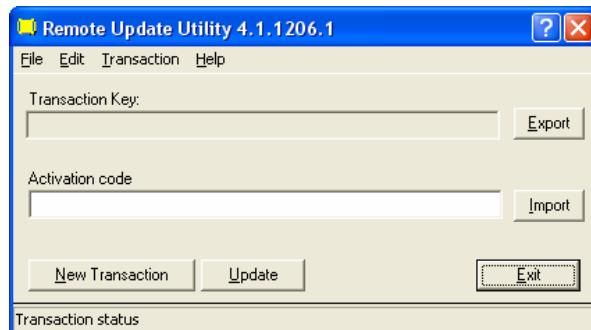


Fig. 1-11 Remote update utility used for hardware dongle reprogramming

A transaction key is created by executing the New Transaction procedure. If successful, a new transaction key is shown in the upper field. This key can be exported to a file with the Export button. This transaction key is also stored onto the hardware dongle. Thus, it is only possible to create a new transaction key, if no transaction was initialized before or if the last transaction has completed. However, it is still possible to overwrite a previously created transaction key which cancels the transaction started before. If an open transaction is detected, a warning appears and the user has the choice to cancel this open transaction.

Then, please send the following information via email to [support@radioplan.com](mailto:support@radioplan.com):

- the serial number of the dongle (as found on the little badge attached on the dongle) and
- the transaction key file `TrKey.rfp`.

In return the customer will receive a file that comprises the updated hardware dongle information, namely an activation code (default: `ActCode.rfp`). This file must be imported into the Remote Update utility by using the Import button. Finally, the update procedure executed by pressing the Update button will reprogram your hardware dongle.

Note that only that hardware dongle can be updated that the transaction key was created for. It is not possible to reprogram other hardware dongles using the submitted activation code.

### 1.9.2 Software License Files

The alternative to hardware dongles is license files. In this case, all license information is stored in a file (`rpslicense.1f`) which is delivered on a separate diskette. Please, copy the license file into the `%ApplicationPath%` directory before you start RPS for the first time.

The license file must be saved before RPS is removed, reinstalled or an update is installed.

The license file mechanism distinguishes two states of the license. Initially, the license is in a locked status where the application can be executed max. 30 times. During this period, the customer must contact our support in order to unlock the license, refer to Subsection 1.9.2.1 for further details. Once the license has been unlocked, the application is bound to the hardware of a specific workstation. It is possible to transfer a registered license from one PC to another PC.

When using a license file, it must be ensured that the following settings are valid in the `license.ini` file in the `%ApplicationPath%` directory:

```
[License]
License = 2
```

#### 1.9.2.1 Registering RPS

It is necessary to unlock the RPS license when using a license file. Without doing so, RPS can be started max. 30 times. Within this period, the customer must contact our technical support via email ([support@radioplan.com](mailto:support@radioplan.com)), and submit the computer ID number of the workstation where RPS shall be unlocked. Immediately, we return the registration codes to unlock the application.

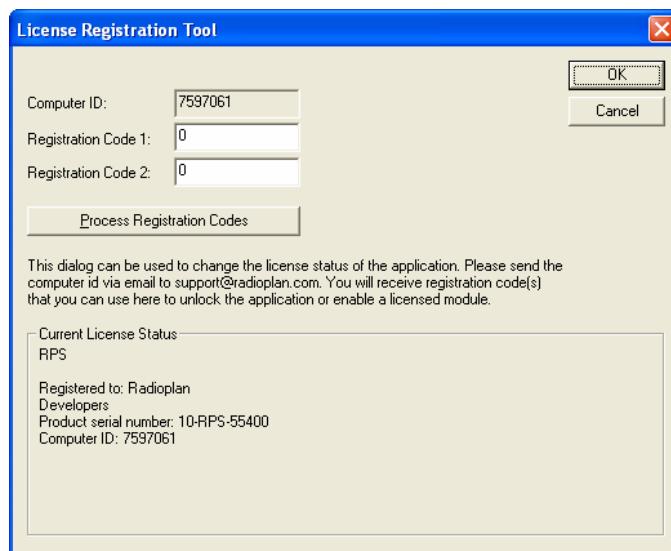


Fig. 1-12 License registration dialog

The computer ID of your workstation can be found in the license registration tool dialog. The dialog is reachable by the menu entry Tools -> License Management -> License Registration Tool. When you have received the registration code(s) from the technical support, you can enter these values in the fields registration code 1 and registration code 2. The button Process Registration Codes will change the license file. In the lower part of the dialog, the current license status is shown.

#### 1.9.2.2 License Transfer

From time to time, customers may want to move licensed software from one machine to another. However, this license transfer scheme can be accomplished without having the customer call for registration codes. In this case, a simple three-step procedure is performed utilizing a blank floppy disk. As an alternative, a shared directory can be used as transfer media as well, which contains currently no license file.

Do not use the original license file diskette that was delivered with RPS!

*Step 1:*

To transfer a license from one computer to another, RPS must be installed on the new machine. This software will be locked, so it will not run unlimited. The Tools -> License Management -> Start License Transfer menu item is chosen which creates a blank license file on the floppy identified with the computer ID number of this machine.

*Step 2:*

During the second step, the blank floppy disk is taken to the licensed (old) machine. The menu item Tools -> License Management -> Transfer this License is chosen on the old machine which manipulates the license file stored on the floppy and de-authorizes the license on the current machine. During this step, the license file on the floppy is enabled and the license file on the current computer is disabled.

*Step 3:*

The last step is to simply move the license file from the floppy disk onto the new computer to be licensed. This file has already been properly enabled for the target machine. The license file can be copied using the Tools -> License Management -> Complete License Transfer menu item again on the target machine.

## 2 Getting Started

This Section will give you a brief introduction how a channel prediction simulation is performed. RPS uses a vector-based database that contains the geometry of the environment. Before any field strength prediction can be done, positions of base stations (transmitters) and mobile stations (receivers, mobile users) must be defined. If all necessary preparations are done, simulation parameters can be set, and you can perform your first simulation. Eventually, you will learn to analyze and visualize prediction results.

### 2.1 Preparing the Environment Database

#### 2.1.1 Starting RPS

You can start RPS like any Windows application. As a second step, you can load the first project into the RPS application. Please select File -> Open, then select the following RPS project:

```
%ApplicationPath%\examples\getting started\getting started 1.rps
```

Immediately, you see a 2D top view of the environment database in the integrated Environment Editor as shown in Fig. 2-1.

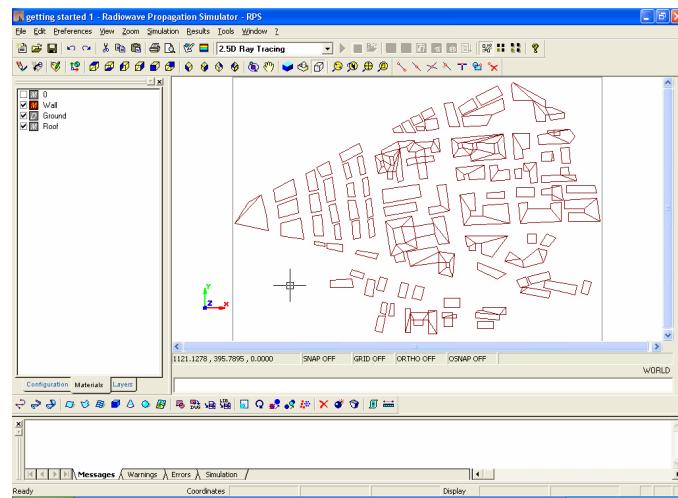


Fig. 2-1 Initial environment of an urban area

### 2.1.2 Adding a new Building in the Environment

The Environment Editor can be used to add or remove buildings, insert additional obstacles like persons or cars, or to change the material layer properties. Its usage is similar to AutoCAD, i.e. it is command driven where each command is followed by some arguments.

Here it will be explained how a new building is added. The easiest way to accomplish this is to first define the walls of the building and then add another polygon as the roof.



Before you start editing, make sure that the world coordinate system is active. Just click the appropriate icon in the upper toolbar of the Environment Editor. The currently chosen coordinate system is indicated in the lower right corner of the viewing area which must be "WORLD" in this case. Also, the proper material layer for the drawing action must be active. This is done by clicking on the "Wall" layer item in the tree window to the left. The currently active layer is indicated by the icon.



Furthermore, it is useful to use an object snap to better align the drawn objects. Select the appropriate icon for the snap to object vertices (tooltip Snap Object End) from the upper toolbar of the Environment Editor. Additionally, the snapping grid should be enabled by double-clicking the area below the viewing area where "SNAP OFF" is written. This string should turn then into "SNAP ON".



Now the walls of the new building can be drawn along a polyline. Use the command Draw Wall along Polyline by clicking the appropriate icon in the lower toolbar of the Environment Editor. First you have to enter the wall height in the command line of the Environment Editor. It should be 15m, so just type "15" in the command line and finish with the <Enter> key. Then the wall line can be drawn by setting the vertices with the left mouse button. Note that the polyline must be correctly closed. This is automatically supported by the object snap, so wait until the first vertex is highlighted by a pink frame before you set the last vertex at the position of the first one. After that the right mouse button must be clicked to stop drawing. The result of this action can be seen in Fig. 2-2.

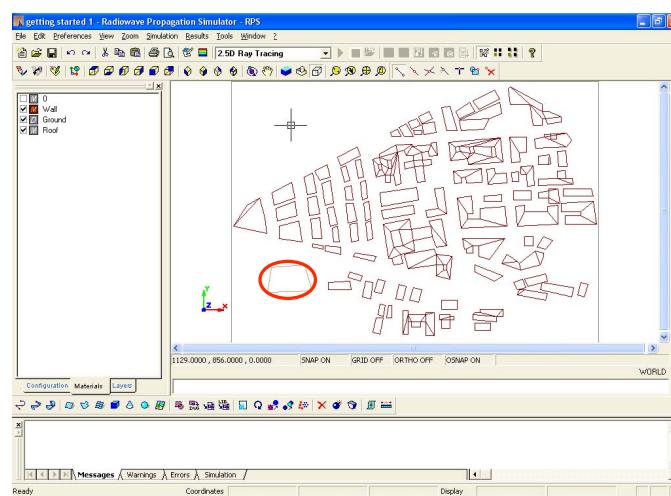


Fig. 2-2 New building wall in the urban environment



In order to be able to properly add the roof, it is useful to change to another perspective now, e.g. the SW view. Then zoom on the new building by using one of the zoom commands from the upper toolbar.





In preparation for adding the missing roof, you should first select the proper material layer, i.e. "Roof" in the tree window to the left. Then the Draw Surface (3D Face) command is used to add vertex by vertex of the roof polygon. Make sure to use the object snap to accurately position the vertices of the roof at the upper corners of the walls.

Now, we have inserted our first building into an environment database. The final status of the new building is shown in Fig. 2-3.

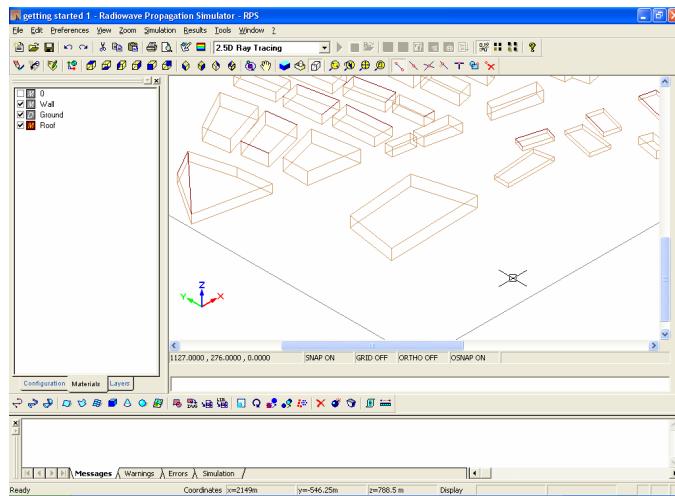


Fig. 2-3 Environment with the user-drawn building

### 2.1.3 Adding a Base Station (Transmitter)



After the modification of the environment, the network structure shall be configured in the following step. For this purpose, we have to switch to the 2D view where base stations and mobile stations can be edited. Use the appropriate icon from the standard toolbar.



Before the network structure is edited, a proper grid width should be set for the following actions. This is done by using the Grid Size icon from the paint toolbar to the left. A dialog appears where the grid size should be set to 5m, see Fig. 2-4.

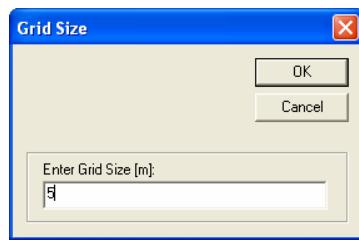


Fig. 2-4 Grid size dialog



As part of the network structure, we place a transmitter into the environment first. This can be performed by selecting the Edit Transmitters icon from the paint toolbar. We click at a position within the environment with the right mouse button, where the transmitter shall be placed. Choose Add... from the appearing context menu. Then a dialog is opened that lets you define some parameters for this transmitter, see Fig. 2-5.

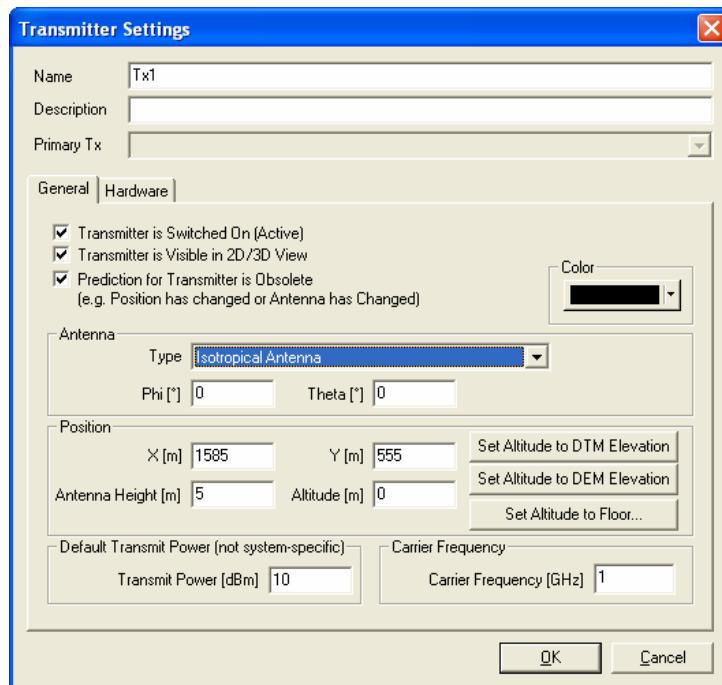


Fig. 2-5 Transmitter settings dialog

The name of the new transmitter shall be “Tx1”. Also, we have to modify the antenna height (elevation over ground) of the transmitter station. We set this to 5 meters. Furthermore, we set its Transmit Power parameter to 10 dBm, the Carrier Frequency to 1 GHz, and choose the default antenna “Isotropic Antenna”. It is possible to place a second or third transmitter in the environment at this time, but we will start with a simple prediction in our first simulation.

#### 2.1.4 Adding Mobile Stations (Receivers)

Receivers can be placed at any position, where radio channel characteristics shall be calculated by RPS. In this example, we will place several receivers along a straight line in one step.



To insert the receivers, the Line of Receivers icon must be selected from the paint toolbar. In this mode, receivers can be placed along a polyline. Each vertex of the line is set with the left mouse button. When the right mouse button is pressed, the line drawing is stopped and the dialog shown in Fig. 2-6 appears.

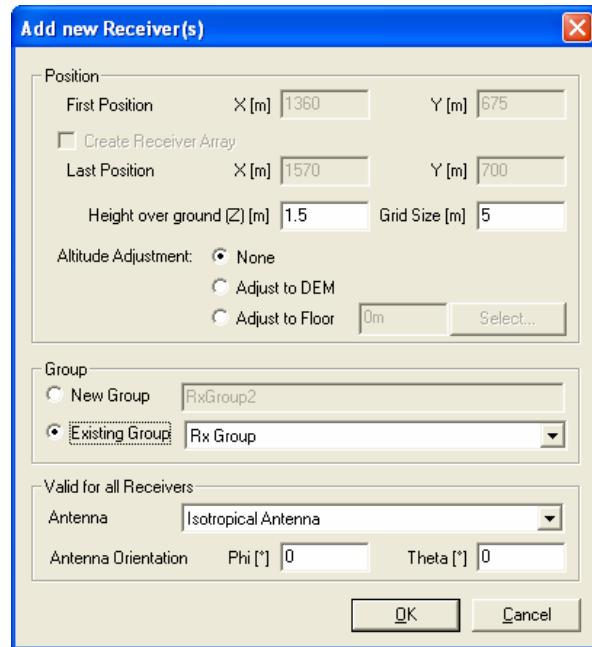


Fig. 2-6 Receiver settings dialog

The height of the receivers need not be modified – 1.5 m is a proper height. Since no other receivers have been created before, you can choose the existing group “Rx Group”. As with the transmitter, the default antenna “Isotropical Antenna” should be selected. As soon as the OK button is pressed, the receivers are added in the environment. The distance between two adjacent receiver positions is defined by the grid size, which was adjusted to 5 meters before.

Up to here, we have prepared the environment, and have placed a transmitter as well as several receivers. Now you can compare your configuration with that in the file getting started 2.rps.

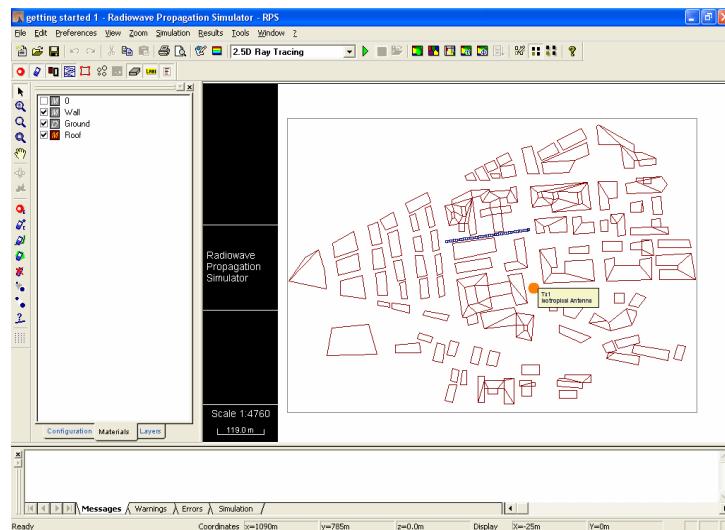


Fig. 2-7 Environment with transmitter and receiver positions

## 2.2 Field Strength Predictions

### 2.2.1 Preparing the Simulation



Before we can perform our first field strength prediction, some additional parameters have to be adjusted. This can be done using the general preferences dialog. It can be reached by using the menu entry Preferences -> General Preferences, or by selecting the corresponding icon from the standard toolbar located on the top of the RPS application. The dialog contains several tabs. The first tab summarizes all parameters for the configuration of the ray tracing algorithms as apparent from Fig. 2-8.

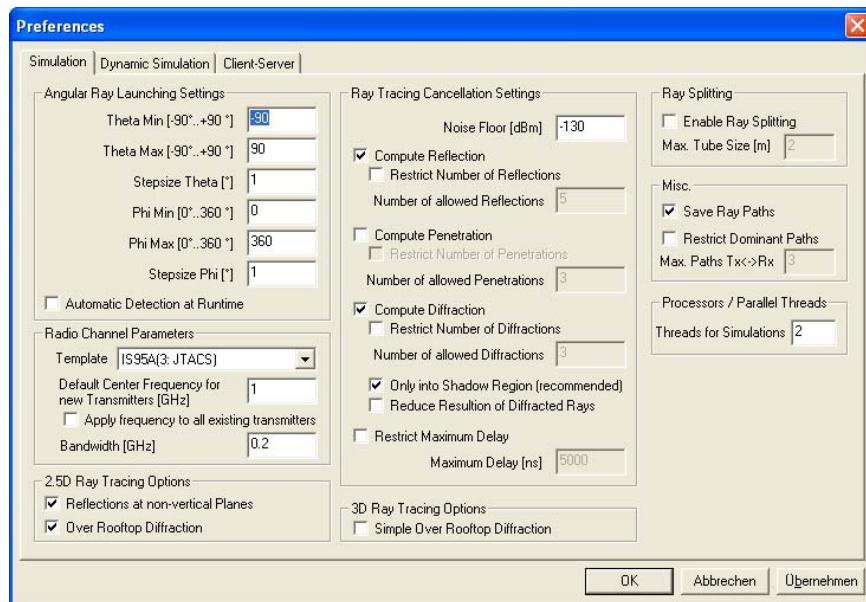


Fig. 2-8 Simulation tab in the general preferences dialog

Please, make all settings as given in the Figure. In particular, we check the Save Detected Rays checkbox located in the “Misc” group which stores the geometry of detected rays for a later investigation. Note that this option does not affect any simulation results, but it is useful for analyzing the prediction results after the simulation process. In the group “Ray Tracing Calculation Settings”, the noise floor (the criterion that determines when the tracing of rays can be stopped) is set to -130 dBm. The number of used threads for a simulation can be set to a value between 2 and 4 the latter one being particularly effective on multi-processor and pipelined processor architectures.

### 2.2.2 Performing a 2.5D Ray Tracing Simulation

Now we can perform our first simulation. First the option Preferences -> Auto Detect Diffraction Edges before each Simulation Run (active by default) should be checked to ensure that the diffraction edges are automatically detected at the beginning of the simulation.



We use the built-in 2.5D ray tracing algorithm. This simulation method can be selected from the list field in the standard toolbar as shown in Fig. 2-9. Then the simulation is started by choosing the Start button in the standard toolbar. The simulation progress is indicated by some status messages in the message window.

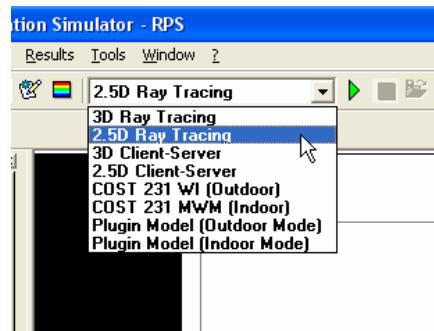


Fig. 2-9 Choosing the 2.5D ray tracing algorithm

After a few seconds, the simulation is done. Now, we can analyze the simulation results. For comparison, the environment with simulation results is stored into the file `getting started 3.rps`.

## 2.3 Prediction Results



Now it is time to get familiar with basic functions for an analysis of the prediction results. First, we can observe the ray paths that give a contribution to the channel impulse response at any receiver position. Using the icon Show Rays from the standard toolbar, the ray paths can be shown in the environment as polylines the color of which is scaled according to their particular received power.

### 2.3.1 2D and 3D Views



The environment including prediction results can be observed in 2D view (horizontal x-y plane) or in 3D view. In the latter case, the buildings, transmitters, and receivers can be rotated, you can zoom into the environment, and you can “fly” over and within the building structure using mouse or key navigation commands to get a detailed and precise overview. You can switch between 2D and 3D views using the icons from the standard toolbar. All prediction results can be shown in 2D as well as in 3D view mode.



Different results can be displayed in surface plots (Section 2.3.2) that each represent a separate result layer. Each result layer has its own presentation properties (color map, alpha blending factor, transparent colors, ranges, etc.). These properties can be modified in the layer settings dialog for the currently displayed result layer. It can either be reached by clicking into the lower part of the black legend to the left part of the viewing area, or using the icon from the standard toolbar on top of the window, see Fig. 2-10.

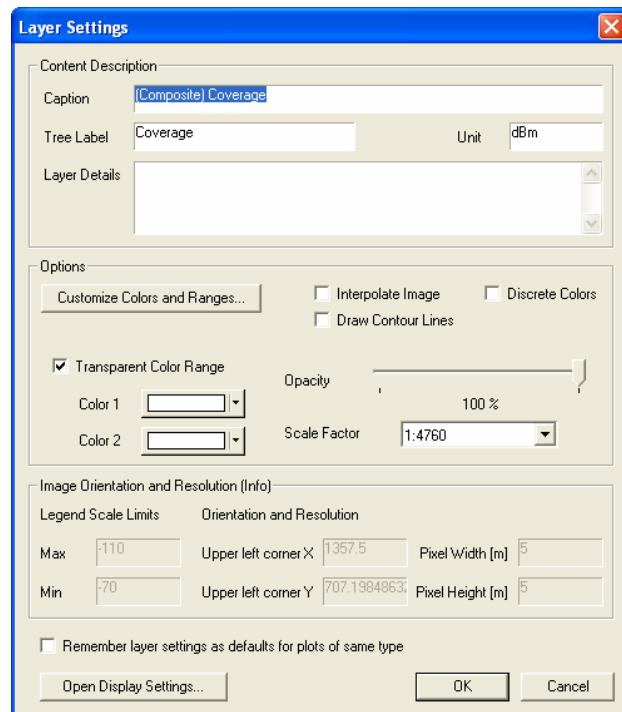


Fig. 2-10 Layer settings dialog

### 2.3.2 Surface Plots

Surface plots let you directly observe channel prediction results within the environment. RPS has functions to calculate the coverage, delay spread etc. Fig. 2-11 shows the environment, the detected ray paths, and the coverage as color surface plot in 3D view when we moved our camera near the receiver positions using the “Fly-by” mode, discussed later in Section 3.6.4.

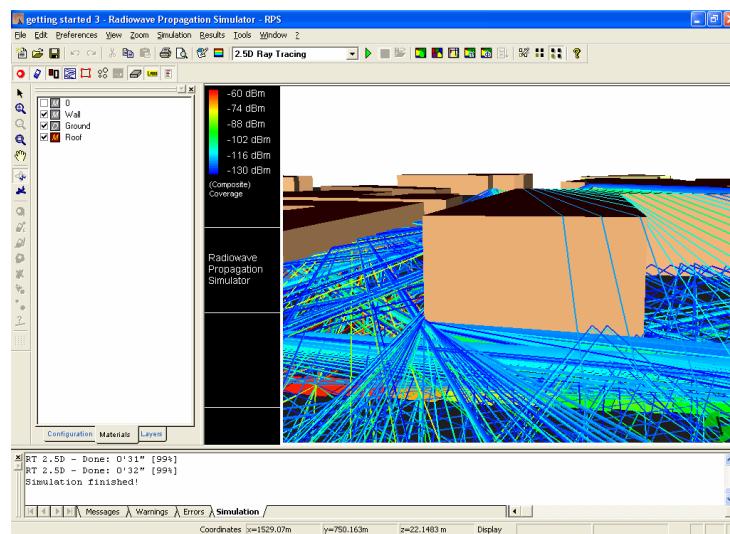


Fig. 2-11 3D view of the environment, ray paths, and coverage values

The coverage, delay spread, signal-to-interference ratio, best server, and user defined surface plots can be shown by using the corresponding functions and icons from the Results -> Surface Plots submenu bar or the standard toolbar.

### 2.3.3 Further Analysis



Besides surface plots, RPS also offers powerful post-processing modules for point, path, and Tx analysis that can be accessed in the 2D view. They allow you to observe characteristic channel data (e.g. channel impulse responses) in rectangular XY charts, polar charts e.g. of angles of arrivals (AoA, DoA), or tables. As an example, we analyze the coverage values along the receiver positions that we have placed in the environment before. This can be done in the path analysis mode that can be entered using the appropriate icon from the paint toolbar to the left.

Then a polyline can be defined with the left mouse button. The drawing is stopped when the right mouse button is pressed. You should draw a simple straight line along the receivers that were placed before. After the right mouse button was pressed, a dialog appears that asks you for a capture radius. This capture radius determines the range around the line to also include receivers in the analysis that are not exactly hit. Enter 5 meters for the capture radius. Next, the path analysis dialog is opened, see Fig. 2-12.

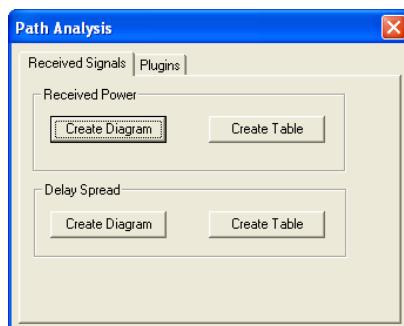


Fig. 2-12 Path analysis dialog

Use the button Create Diagram in the upper group “Received Power” to create a diagram of the coverage along the previously defined path. A separate window is opened that shows the diagram similar to that in Fig. 2-13.

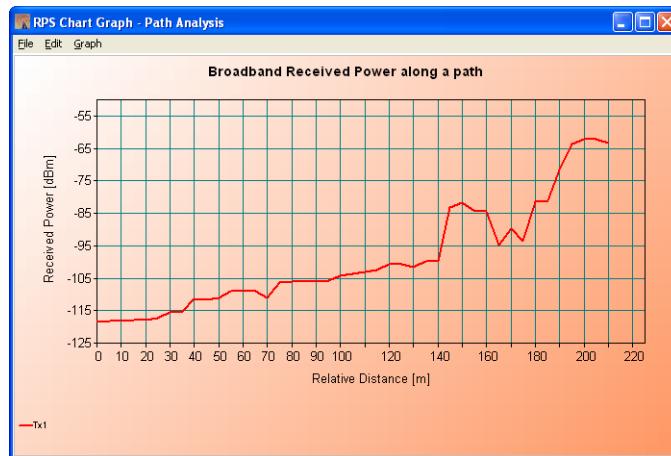


Fig. 2-13 Chart diagram of the wideband coverage along a path

It is possible to place several result windows (diagrams or tables) simultaneously on the screen which allows for a comfortable comparison between different results.

### 2.3.4 Assigning Radiation Patterns to Transmitters

Our simulation was done with isotropic antenna beam patterns. In reality, an antenna has a directed radiation pattern. In this Section you learn to apply another radiation pattern to a transmitter station.

Antennas are handled like templates in RPS, i.e. a certain antenna patterns can be used by several transmitters and/or receivers. All available antennas are listed in the Configuration tab of the tree window to the left.

A new antenna can be imported by right-clicking on the Antennas node and choosing Import Antennas... from the context menu. A file open dialog appears where you should select the Dipole Antenna. Immediately it is added in the antenna tree, see Fig. 2-14.

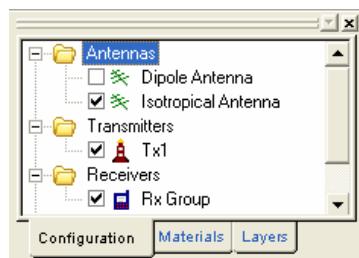


Fig. 2-14 Configuration tab with network setup

In order to assign the new dipole antenna to the transmitter, you must double-click the Tx1 item in the Configuration tab of the tree window (below the antennas) to open its settings dialog. There the dipole antenna can be selected in the appropriate list field.

The pattern of the dipole antenna can also be displayed in a polar diagram. This function can be accessed from the context menu of the corresponding antenna in the Configuration tab of the tree window by choosing the Show Antenna Diagrams option. Then, the horizontal and vertical antenna patterns should be presented in a separate window like in Fig. 2-15.

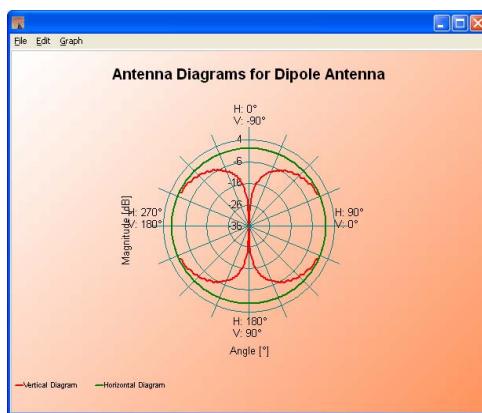


Fig. 2-15 Chart plot of the radiation patterns of the dipole antenna

As a further modification the transmit power of the base station shall be changed. It is assumed that the base station, called New Transmitter here, is sending a signal of only 3 dBm. This can also be configured in the transmitter settings dialog as described above in conjunction with the assignment of the antenna pattern.



After these changes have been confirmed, the resulting coverage with the chosen antenna and modified transmit power can be immediately presented by pressing the coverage icon again. Note that no recalculation of the radio channel is necessary!

The current state is stored in the file `getting started 4.rps`.

## 2.4 Incremental Recalculation of the Radio Channel



Now it is time to place a second transmitter (base station) into the environment. We use the corresponding icon from the paint toolbar, add a second transmitter, and assign it a height over ground of 8 meters.



We have to recalculate the radio channel, and we use the 2.5D ray tracing mode again. Before the calculation is started, RPS detects that for some transmitters the calculation was done previously, and asks the user, if the channel prediction should be performed only for "obsolete" transmitters. Since we use the same algorithm as before, we can use this incremental prediction mode. It calculates the radio channel only for the newly added transmitter station.

After the second channel prediction, each receiver position has stored the rays coming from both transmitter stations. Now, it is possible to investigate the signal-to-interference ratio (SIR), or the site-specific best serving cell ID, for instance.

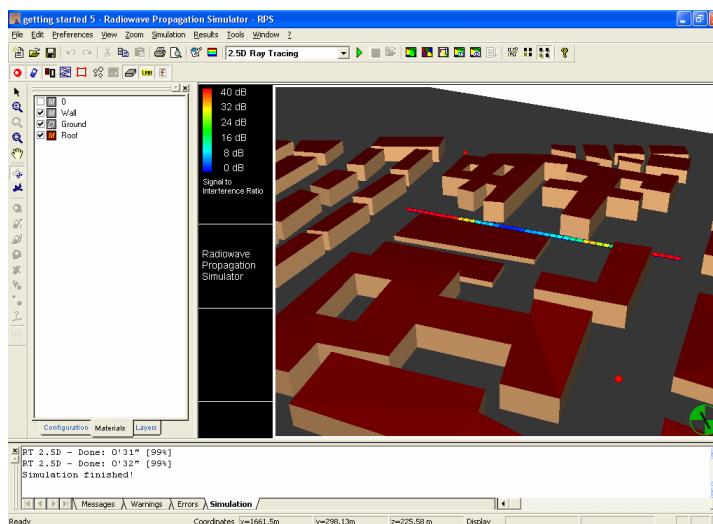


Fig. 2-16 Signal-to-interference ratio with respect to the best serving transmitter

Now, you should be able to perform your first owner-driven simulation. The functions and procedures used here are explained in more detail in the following Sections. The final database is available in the file `%ApplicationPath%\examples\getting started\getting started 5.rps`.



## 3 Working with RPS — Overview

This Chapter gives an introduction on how to conduct efficient coverage predictions and subsequent performance analysis tasks in radio network planning with RPS. Thus the reader will get familiar with the Graphical User Interface of RPS and will learn about the main configuration components for a coverage prediction. An introduction to the foundations of geometrical optics will provide insights into the basic principles of the ray tracing mechanisms applied in RPS. Last but not least, the plug-in support for user-defined propagation and analysis functions is described.

### 3.1 Components of the Working Environment

RPS provides a comfortable graphical user interface that guides the user through all tasks in the coverage planning process. In particular, it enables the user to:

- create/open/save projects
- view/configure environment data in several ways
- view/configure network data in several ways
- setup coverage prediction simulations (3D/2.5D ray tracing, empirical models, plug-ins)
- start/stop/control coverage predictions
- view and evaluate result data in various ways

RPS is invoked by selecting RPS n.n ->  RPS n.n from the Windows start menu. The application window is opened that looks similar to Fig. 3-1, however initially with an empty viewing area. As soon as a project has been opened, the viewing area is filled.

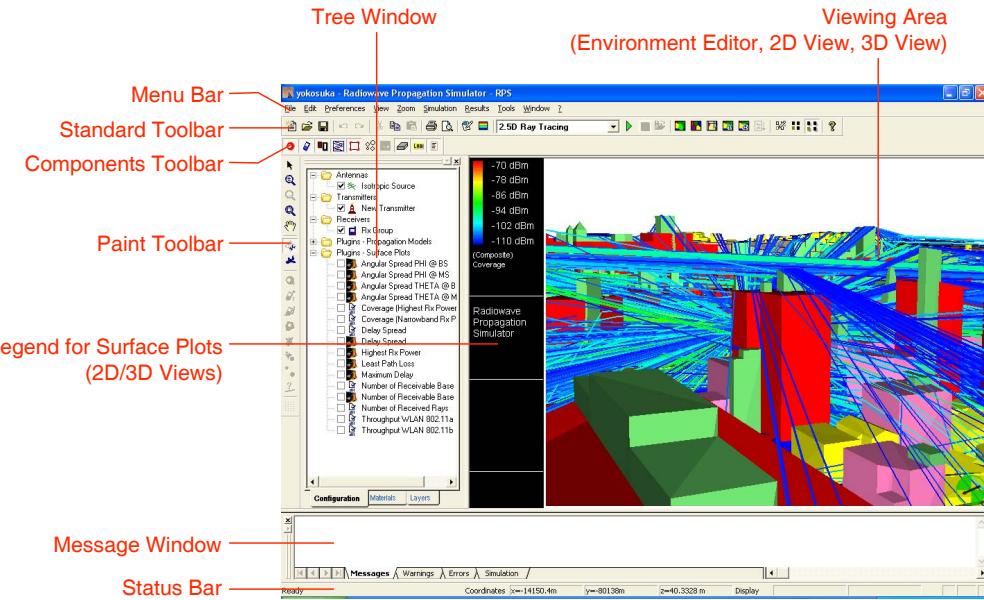


Fig. 3-1 Window controls of the RPS application

### 3.1.1 Menu Bar

Most of the functionality of RPS can be reached from the menu bar, see Fig. 3-2.

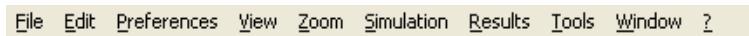


Fig. 3-2 RPS menu bar

A detailed overview of the distinct menus can be found in Appendix 10.1.1.

The framework supports the definition of user-defined shortcuts to menu entries. Any user-defined shortcut is stored persistently, so that they are also available in the next RPS session. In order to define a shortcut to a frequently used menu entry, open the shortcut keys dialog using the Tools -> Shortcuts... menu entry. A list of all possible menu entries appears in the left part of the window. In the right part, the shortcuts are listed, which are currently assigned to the corresponding menu entry, see Fig. 3-3.

When a certain item is selected in the left macro list, the corresponding description appears in the upper right corner. By clicking the button Create Shortcut a new window is opened where the user can edit the shortcut for the selected macro by typing it. After the shortcut has been defined, it is shown in the lower right corner. Additionally, the new shortcut is also shown in the corresponding menu entry to the right.

To remove a defined shortcut, just select the corresponding shortcut in the lower right corner (for a selected macro in the left list) and click the button Remove. The pre-defined shortcut settings can be restored by clicking the button Reset All.

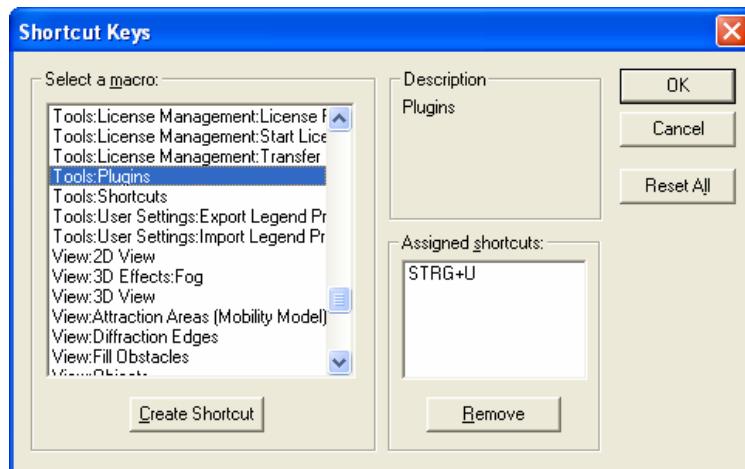


Fig. 3-3 Shortcut keys configuration dialog

The user-defined shortcuts also appear in the menu bar to the right of the corresponding menu entry.

### 3.1.2 Toolbars

Three different toolbars are provided for easy access to frequently used functions. When positioning the mouse pointer over a toolbar icon, a tooltip with a short description of the corresponding function appears. An explanation of the respective function is also shown in the status bar.

- The **standard (or default) toolbar** offers functionality that is mainly contained in the File, Edit, View, and Simulation menus. They embrace, among others, functions to open, save, and view configurations and results, and to control coverage simulations.
- The **components toolbar** has some checkboxes to show or hide the distinct configuration data items in surface plots. The same functionality is also contained in the View menu. In particular, the presentation of transmitters, receivers, environment objects, rays, diffraction edges, etc. can be toggled.
- The **paint toolbar** defines the mode that controls the actions of the mouse pointer. In the default paint mode, the window controls can be used in the normal way. In all the other modes, the mouse pointer is used to perform special actions (e.g. zoom, pan, set transmitters/receivers, etc.).

More information about the toolbars can be found in Appendix 10.1.2.

### 3.1.3 Tree Window

The tree window lets the user control all RPS project data. It consists of three tabs: Configuration tab, Materials tab, and Layers tab.

In general, each tab presents its data in a tree structure. Each tree item (leaf) has a context menu that can be accessed by right-clicking the item. In the Configuration tab, many items have a checkbox where those items can be separately activated. This way e.g. transmitters can be switched on or off.

### 3.1.4 Viewing Area

The large window is used for the presentation of the environment/network/result data in different views. Moreover, the environment database and the network configuration can be edited in the Environment Editor or the 2D view, respectively.

In particular, three different viewing modes are available. The standard toolbar provides a quick access to switch between the different viewing modes. The respective icons are listed in Table 3-1.

Table 3-1 Icons for different viewing options

| Icon | Description                                  | Tooltip            |
|------|--|--------------------|
|      | Integrated Environment Editor                | Environment Editor |
|      | 2D View (incl. Network Configuration Editor) | 2D View            |
|      | 3D View                                      | 3D View            |

### 3.1.5 Result Windows

Despite surface plots, result data of analysis and presentation functions are displayed in separate windows. These result data can be shown in two different ways, as

- chart graph (diagram, histogram), or as
- table (result parameters, summary report, etc.).

A result window has an own menu bar with functions to print, save, and copy the presentation, as well as to modify the presentation style in case of chart graphs. More details on the different result presentation options can be found in conjunction with the explanation of the result analysis in Chapter 6.

### 3.1.6 Message Window and Status Bar

The message window is used by the MATLAB based plug-ins for custom surface plots. Furthermore, warnings, temporary results, possible errors, and the simulation progress are displayed in this window.

The status bar displays the coordinate information in all three viewing modes. In 3D view, the position of the camera is displayed. Furthermore, a progress bar indicates the progress in the execution of time-consuming operations.

### 3.1.7 Showing and Hiding Graphical Components

Different window controls like the standard toolbar, the status bar, the message window, and the tree window can be shown or hidden selectively. Furthermore, the window components of them (message/tree window) can be docked within the framework (default state), or their state can be changed to floating windows.

The components can be shown or hidden by checking or unchecking the according entries in the Window menu. The docking state of the window components can be changed by double clicking on their thick border areas.

### 3.1.8 Full Screen View

RPS provides a full screen view mode. It can be activated using Zoom -> Full Screen View from the menu bar. Alternatively, the <F5> function key activates the full screen mode. In this mode, the viewing area is enlarged to the full screen size. The menu bar appears, when the mouse pointer is placed near the upper border of the screen.

The normal view mode can be restored by clicking on the button in the upper left corner of or by pressing the <ESC> key.

### 3.1.9 Clipboard Functions

Any graphics, chart graph, or table can be exported to other Windows applications via the clipboard. The format of graphics and chart graphs can be pixel based (bitmap format) or vector based (windows meta file format). Tables are copied in Windows Excel 97 format.

The normal copy function (invoked by Edit -> Copy from the menu bar or typing <Ctrl-C>, respectively) are used for bitmaps of surface plots and chart graphs, and table copies. If a surface plot shall be copied as a Windows meta file, use the menu entry Edit -> Copy Plot as Graphic instead.

### 3.1.10 Undo Buffer

RPS provides undo/redo functions in the integrated Environment Editor and in overview dialogs with table grids. So it is possible to revert the last modifications done on the environment or in overview dialogs. Note that the Environment Editor only allows a single redo action. Redo is not available in overview dialogs.

## 3.2 Coordinate Systems

RPS as a GIS-enabled platform supports both geographical and Cartesian coordinate systems. To understand how RPS handles different coordinates, first a short introduction into types of coordinate systems and their transformation is given. Next, the orientation of the supported 3D coordinate systems is explained.

### 3.2.1 Geographical and Cartesian Coordinate Systems

A geographical coordinate system describes a position in 3D space by a longitude and a latitude, often given in degrees that are measured relative to a reference ellipsoid. This type of coordinate systems is frequently called an ECEF (earth centered, earth fixed) system.

A Cartesian coordinate system defines the projection of a part of the earth's surface onto a flat plane, the projection plane. Since the earth is not an ideal sphere, the projection of its surface onto a flat plane is done part wise. Many different coordinate systems exist in order to describe a small piece out of the earth's surface by a flat plane as accurate as possible.

Some of the projections are defined relative to a set of common reference ellipsoids, given by a datum transformation. A datum transformation is the composite operation of a translation, a rotation, and a scaling of a point in 3D space. Such a transformation can be described by seven parameters in a way given by (3.1). Table 3-2 contains a description of the parameters.

$$\begin{bmatrix} x \\ y \\ z \end{bmatrix} = \begin{bmatrix} dx \\ dy \\ dz \end{bmatrix} + \begin{bmatrix} 1+m & 0 & 0 \\ 0 & 1+m & 0 \\ 0 & 0 & 1+m \end{bmatrix} \cdot \begin{bmatrix} 1 & r_z & -r_y \\ -r_z & 1 & r_x \\ r_y & -r_x & 1 \end{bmatrix} \cdot \begin{bmatrix} x_0 \\ y_0 \\ z_0 \end{bmatrix} \quad (3.1)$$

Table 3-2 Datum transformation parameters

| Parameter | Description                  | Unit                    |
|-----------|------------------------------|-------------------------|
| dx        | translation dx to WGS84      | m                       |
| dy        | translation dy to WGS84      | m                       |
| dz        | translation dz to WGS84      | m                       |
| m         | scaling factor to WGS84      | ppm (parts per million) |
| rx        | rotation ro (rho) to WGS84   | radian                  |
| ry        | rotation rf (phi) to WGS84   | radian                  |
| rz        | rotation rk (kappa) to WGS84 | radian                  |

In order to represent a position in both, Cartesian coordinates and geographical coordinates, a projection coordinate system is required. It defines the transformation of Cartesian coordinates into geographical coordinates and vice versa. Note, that RPS uses this projection coordinate system for all coordinates stored internally.

In some cases it may be helpful to represent coordinates in a different coordinate system compared to the projection system. This can be done by choosing a second display coordinate system. This display coordinate system can be a Cartesian or a geographical coordinate system. Therefore, it is possible to represent display coordinates in a second projection or in a second geographical coordinate system. Fig. 3-4 explains the coordinate system transformation between a projection and a display coordinate system schematically.

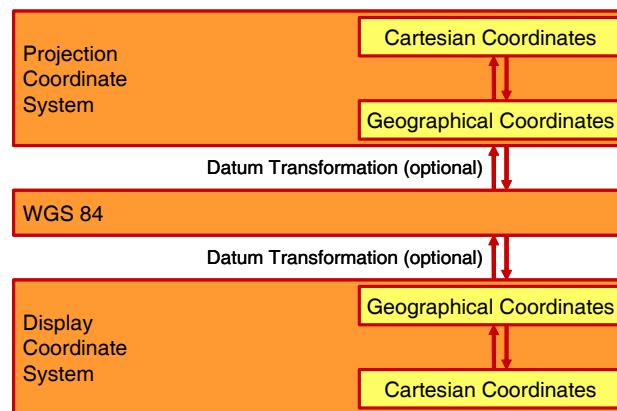


Fig. 3-4 Principle of coordinate system transformation between projection and display coordinate systems

The coordinate systems can be chosen in the projection dialog, accessible via [View -> Projection...](#) from the menu bar, see Fig. 3-5.

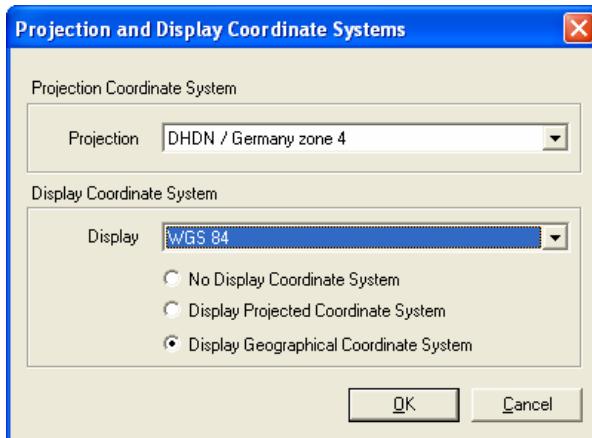


Fig. 3-5 Projection settings dialog

From the upper list field in the coordinate settings dialog one of the coordinate systems for the projection can be selected. In the lower part, optionally a display coordinate system can be chosen. Depending on the selection buttons, the display coordinate system can be a Cartesian or geographical system.

The coordinate system list itself can be modified by editing the `wkt.ini` file, located in the application folder. Each coordinate system is defined by at least two lines. An optional third line can be used for a user-defined datum transformation. The format used in the ini-file is called Well Known Text (WKT) format. It can be handled by several GIS tools. The snapshot below shows a definition of a user-defined projection coordinate system with a special datum transformation to and from WGS84:

```
EPSG=64261
```

```
PROJCS[ "DHDN MMO / Germany zone
3", GEOGCS[ "DHDN", DATUM[ "Deutsche_Hauptdreiecksnetz", SPHEROID[ "Bessel
1841", 6377397.16, 299.15281]], PRIMEM[ "Greenwich", 0], UNIT[ "degree", 0.01745329251994
33]], PROJECTION[ "Transverse_Mercator"], PARAMETER[ "latitude_of_origin", 0], PARAMETER[ "central_meridian", 9], PARAMETER[ "scale_factor", 1], PARAMETER[ "false_easting", 350
0000], PARAMETER[ "false_northing", 0], UNIT[ "metre", 1]]
```

```
TOWGS84=583.00 68.0 399.50 0 0 -13.6e-6 11.3e-6
```

Internally, the EPSG identifiers will be stored into the RPS project file for the projection and optionally the display coordinate system. Therefore, it is necessary that all users that use the same database work with equal coordinate system definitions which can be simply achieved by using one identical `wkt.ini` file.

### 3.2.2 Euclidian Cartesian Coordinate Systems used in RPS

Internally, RPS uses a 3-dimensional Euclidian coordinate system as depicted in Fig. 3-6. It consists of three orthogonal axes (x, y, z) where the x- and y-axes form a horizontal plane, and the z-axis is vertical.

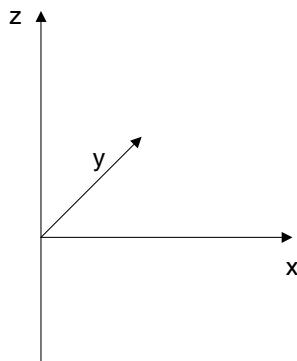


Fig. 3-6 Coordinate system used in RPS

Often the coordinates of the x-y-plane are denoted by means of the orientations North – South – West – East. They have the following meanings:

|       |                      |
|-------|----------------------|
| North | positive y-direction |
| South | negative y-direction |
| West  | negative x-direction |
| East  | positive x-direction |

These relations are depicted in Fig. 3-7.

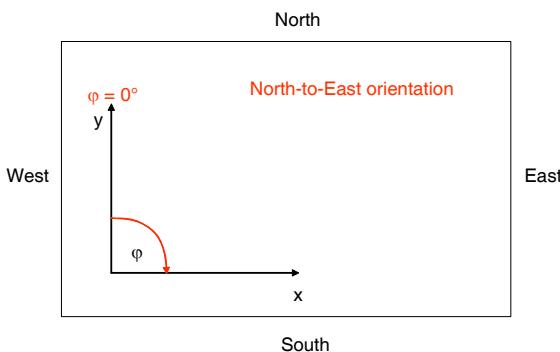


Fig. 3-7 Horizontal x-y-plane of the coordinate system

Especially when handling antenna patterns, directions in the 3-dimensional coordinate system are given in terms of the angles  $\varphi$  and  $\theta$ . The orientation of these angles is presented below.

The angle  $\varphi$  in the horizontal x-y plane is defined in the range  $[0^\circ \dots 360^\circ]$  with the so-called North-to-East orientation (refer to Fig. 3-7). This means that  $\varphi$  starts at the positive y-axis (North), and then is counted clockwise towards the position x-axis (East).

The vertical x-z-plane is shown in Fig. 3-8. The angle  $\theta$  is counted downward and is defined in the range  $[-90^\circ \dots 90^\circ]$ . An angle of  $\theta = 0^\circ$  denotes the direction of the horizontal x- or y-axis, i.e. positive angles represent a corresponding downtilt of an antenna. An angle of  $\theta = -90^\circ$  denotes the direction of the z-axis. As an example, an antenna with  $\theta = 10^\circ$  means that the antenna has  $10^\circ$  downtilt.

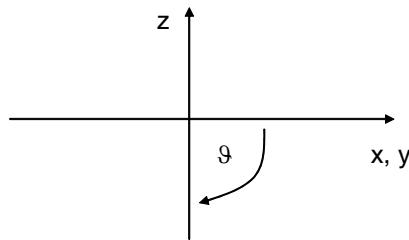


Fig. 3-8 Vertical x-z plane or y-z plane of the coordinate system

In contrast to this, vertical antenna patterns are usually defined for a range of  $\theta = [0^\circ \dots 360^\circ]$  where an angle of  $\theta = 0^\circ$  denotes the direction of the x-axis, i.e. with the same orientation as described above. The rear hemisphere of the pattern is then in the range  $\theta = [90^\circ \dots 270^\circ]$  from bottom to top.

### 3.3 Layer Concept

In the usage of RPS, the layer concept plays an important role. Basically, two different types of layers are distinguished:

- **Material Layers:** A material layer describes the material properties of environment objects. All available material layers are summarized in the Materials tab of the tree window.
- **Result Layers:** A result layer contains arbitrary result data bound to certain receivers (one/some/all receivers or receiver groups). All available result layers are summarized in the Layers tab of the tree window.

#### 3.3.1 Material Layers

Material layers are useful to combine several groups of environment objects with similar properties. RPS uses material layers to group objects with identical electrical properties relevant for wave propagation such as relative permittivity, thickness, certain propagation effect flags, etc. (see Section 4.1.6). Fig. 3-9 gives an example of the presentation of available material layers in the tree window; Table 3-3 presents an overview of all material properties used in RPS.

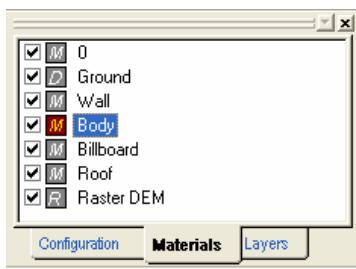


Fig. 3-9 Example of a material layer list

Only one material layer can be active at the same time. This one is denoted by a colored icon (e.g. ). All other material layers are shown as grey icons (e.g. , denoting inactive layers). For all actions in the Environment Editor, always the active material layer is applied. The active layer can be chosen by a single mouse click on the layer item in the tree window; a double click opens the settings dialog of its material properties. The checkbox left of the layer item influences its visibility (sets/unsets the active flag). Read more on the material layer handling in conjunction with the Environment Editor in Section 4.1.

Table 3-3 Material layer properties

| Property                | Description   |
|-------------------------|---|
| Name                    | Name of the material layer.<br><b>Can be imported/exported in DXF/DWG format.</b>   |
| Description             | An arbitrary description for the material layer.  |
| Active Flag             | If this flag is set, the layer is visible and is considered in coverage simulations.<br><b>Can be imported/exported in DXF/DWG format.</b>  |
| Locked Flag             | If this flag is set, the layer properties cannot be changed.<br><b>Can be imported/exported in DXF/DWG format.</b>  |
| Color                   | The presentation color of the material layer.<br><b>Can be imported/exported in DXF/DWG format.</b>   |
| Thickness               | The thickness of objects of this layer that is considered for propagation effects (reflection, penetration).  |
| Relative Permittivity   | The relative permittivity $\epsilon_r$ of objects of this layer that is used for propagation effects (reflection, penetration, diffraction).  |
| Winter Flag             | With this flag an alternative setting for the relative permittivity can be activated.   |
| Penetration Flag        | If this flag is set, the layer supports penetration.  |
| Reflection Flag         | If this flag is set, the layer supports reflection.   |
| Diffraction Flag        | If this flag is set, the layer supports diffraction.  |
| Filled Obstacles Flag   | If this flag is set, objects of this layer are treated as if they belong to closed, completely filled obstacles (e.g. persons, cars, etc.).   |
| Use as Vector DEM Layer | If this flag is set, objects of this layer are treated as DEM layer defining the ground level. As a consequence, network elements (transmitters, receivers) or other environment elements can be placed in a height relative to this DEM layer.   |
| Use as Raster DEM Layer | If this flag is set, the properties of this material layer are used for a raster DEM matrix which can be loaded separately. It is not allowed to define more than one layer as raster DEM layer. Network elements or other environment elements can be placed in a height relative to this DEM layer. |

There are three different types of material layers. Besides the normal material layers for polygons, there are two additional types of DEM layers depending on the type of DEM data (vector or raster data). These layer types are summarized in Table 3-4.

Table 3-4 Material layer types

| Layer Icon | Description  |
|------------|--|
|            | A normal material layer. The polygons of the respective layer are assigned the material properties of this layer.  |
|            | A vector DEM layer. Similar to a normal material layer. Additionally, the polygons of this layer are treated as DEM layer, i.e. other elements like transmitters or receivers can be positioned with respect to the DEM layer.                   |
|            | A raster DEM layer. A raster DEM matrix can be separately loaded. This matrix inherits the material properties of the raster DEM layer. Especially for large areas, a raster DEM matrix can achieve a higher efficiency than a vector DEM layer. |

The material layer concept in RPS is derived from AutoCAD layers. This means that the layer definitions of imported DXF/DWG environment files are retained (concerning layer name, color, active/locked state, and assignment to environment objects). However, RPS needs more information on the material properties as explained above. Thus, initially this information is filled with default values. It can be changed at any time.

As another advantage of layers, e.g. the relative permittivity could be redefined more easily for all objects with identical material properties, as soon as another frequency range is investigated. Especially the imaginary part of  $\epsilon_r$  is highly frequency dependent.

A material layer must be unique in RPS. Two layers are different, when they differ in at least one of their properties. More about material layer handling is discussed in Section 4.1.6.

### 3.3.2 Result Layers

RPS supports an arbitrary number of layers for surface plots from result data. These layers can be displayed in arbitrary order and with freely definable transparency. All created layers are collected in the Layers tab of the tree window. An example set of different layers is shown in Fig. 3-10.

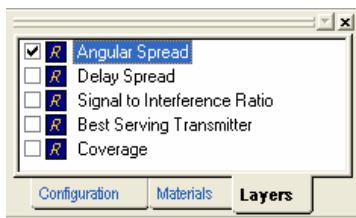


Fig. 3-10 Sample view of the Layers tab

Once a layer has been created, it is held in the RAM without any further change. Thus the presentation performance of surface plots is remarkably improved. A layer is only recalculated if its visible pixel resolution is changed e.g. due to a zoom action.

Note that a result layer is not updated automatically if the underlying data are modified (e.g. if the coverage is recalculated). Only if the corresponding result layer is created anew, the layer presentation is adjusted.

Some frequently used result layers can be created from according icons in the standard toolbar or corresponding menu entries in the Results -> Surface Plots submenu. Additionally, the plug-in technology for user-defined surface plots can be used to create arbitrary result layers. Read more about the creation of result layers in Chapter 6.

All created result layers are presented in descending order in the Layers tab of the tree window, denoted with the icon . A newly created layer will be placed *topmost*, i.e. it will appear as *first* layer in the Layers tab. A layer is visible if the checkbox left of the layer is marked. Checking a layer automatically selects that layer, too. In turn, a selected layer need not be visible, though. A double-click on a layer selects that layer and makes it the only visible one; the same function can be chosen from the layer's context menu by clicking Activate only this Layer.

In general, the order of the result layers in the Layers tab decides upon the visibility of the layers, if several layers are displayed simultaneously. Higher ordered layers are always

displayed above lower ordered layers. The order of the layers can be changed either by means of the context menu functions or by moving layers with the mouse. For the first option, see the description of the different move functions in Table 3-5 below. For the latter option, click on a layer icon and hold the left mouse button pressed down. Then move that layer to the desired position in the hierarchy of the Layers tab and drop it there. The layers are automatically reordered then.

The black legend shown in the surface plot always belongs to the currently selected layer (not immediately visible). Its properties can be set in the layer settings dialog (see Section 3.4.3) by clicking into the lower part of the legend or selecting Settings... from the context menu of the layer. It is especially useful to modify the transparency (alpha blending) of layers to visually overlay several of them.

All the functions of the result layer context menu are summarized in Table 3-5. The context menu can be reached by right-clicking the respective result layer in the Layers tab.

Table 3-5 Result layer functions

| <i>Layer Function</i>               | <i>Description</i>  |
|-------------------------------------|---|
| <u>Settings...</u>                  | Views the layer settings dialog of the result layer. Can be used to define all graphical properties of a layer as, e.g., alpha blending, scaling, color palette, etc. |
| <u>Activate only this Layer</u>     | Selects the result layer and makes it the only visible one, i.e. all other layers are unchecked.  |
| <u>Remove</u>                       | Removes this layer from the Layers tab.   |
| <u>Remove All</u>                   | Removes all layers of any type from the Layers tab.   |
| <u>Remove Layers of this Type</u>   | Removes all layers of this type from the Layers tab. If layers of different types are available, they will be retained.   |
| <u>Move to Foreground</u>           | Makes this result layer the topmost one, thus moving it to the highest position in the Layers tab.  |
| <u>Move to Background</u>           | Makes this result layer the bottommost one, thus moving it to the lowest position in the Layers tab.  |
| <u>Move one Layer Up</u>            | Increases the order of this result layer by one, thus moving it one position up in the Layers tab.  |
| <u>Move One Layer Down</u>          | Decreases the order of this result layer by one, thus moving it one position down in the Layers tab.  |
| <u>Coincide with other Layer...</u> | Opens a dialog to select a function to coincide this result layer with another one (see below).   |
| <u>View as Table</u>                | Shows the data of this result layer in a table. For each receiver position one value is given.  |
| <u>Save as TIFF Image...</u>        | Saves the current result layer presentation into a TIFF file. Shows a file save dialog to select a file name. The layer is saved together with a TFW file.            |

The graphical settings for each type of result layer are stored in a `legend.ini` file which is automatically created by RPS. As an example, whenever a coverage layer is created, the same color palette and scaling settings are used as once configured by the user. On every change of the graphical layer settings the `legend.ini` file is updated in the background such that always the latest settings for the corresponding layer are used.

An interesting option is to coincide the data of one result layer with that of another one. Choose the entry Coincide with other Layer... from a layer's context menu to show a dialog where the required settings can be made, Fig. 3-11.

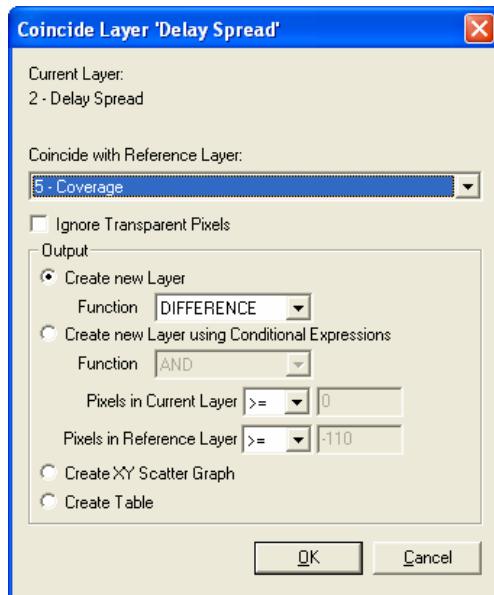


Fig. 3-11 Layer coincidence dialog

Especially for the result analysis this function can unveil relations between different parameters that otherwise would be difficult to discover. The basic principle is to relate the pixel-oriented data of two layers by means of a certain operation (e.g. DIFFERENCE, SUM, QUOTIENT, PRODUCT, etc.) to produce a new layer. It is also possible to display the position-oriented mapping of one layer to another in a chart graph (so-called scatter graph). Read more about the layer coincidence function in Chapter 6.

### 3.4 Personal Workspace

This Section introduces the options to modify the graphical presentation in the viewing area. Such presentation properties are collectively called the workspace which can be adjusted by each user individually.

The first subsection describes the workspace handling in RPS, whereas the following subsections highlight different aspects of the workspace.

#### 3.4.1 Workspace Handling

The workspace is the conglomeration of all properties that affect the appearance of the graphical elements in the viewing area. In particular, it contains:

- display settings, and
- individual layer settings (or graphical preferences).

The workspace settings are initially set to default settings which can be defined globally for all users at a customer. However, every user can individually adjust his workspace settings and store them. Such user specific settings would then override the global pre-configurations.

On start of RPS, the system tries to read workspace settings from a file `rpsworkspace.ini` at different locations as specified in Table 3-6.

Table 3-6 Order of workspace import on start of RPS

| <i>Order</i> | <i>Checked Location</i>  | <i>Comment</i>  |
|--------------|--|---|
| 1            | <code>%ApplicationPath%\rpsworkspace.ini</code>  | A central workspace is always loaded during RPS startup, if the ini file exists at this location. This global workspace can be overridden by a consecutively loaded workspace file with user-specific workspace settings.                   |
| 2            | <code>%RPSHOME%\rpsworkspace.ini</code>  | An NT environment variable “RPSHOME” can be created that defines the location of the user-specific workspace settings.  |
| 3            | <code>Documents and Settings\&lt;username&gt;\Application Data\Radioplan\RPS\rpsworkspace.ini</code> | If the RPSHOME environment variable is not defined or the ini file is not found there, RPS would try to load the workspace from this location. This option might not work if the RPS client is run on a remote server (e.g. Citrix server). |
| 4            | user's Windows registry  | If any of the previous user-specific workspaces cannot be loaded, the user-specific workspace settings would be retrieved from the registry. This option might not work if the RPS client is run on a remote server (e.g. Citrix server).   |

If one of the files does not exist or cannot be read, it automatically jumps to the next one. So first a global settings file is queried which effectively can be overridden by local settings of the user, if such user-defined settings exist.

When RPS is shut down, the current workspace settings are automatically saved in the local `Documents and Settings\<username>\Application Data\Radioplan\RPS\rpsworkspace.ini`. Thus it is ensured that these user-specific settings will be reloaded and active after a restart of RPS according to the priority given in Table 3-6.

#### 3.4.1.1 Handling of Legend Presets

Similar to the workspace, also the legend presets (defining the available color palette templates to be applied for layers) can be stored in a `legend.ini` file. As such, the legend presets are not a part of the actual workspace.

The `legend.ini` file(s) are handled in the same way as described above for the workspace file. Thus on the shutdown of RPS, a local version of the `legend.ini` file is automatically saved that would be retrieved when RPS is restarted. Generally, RPS tries to load a `legend.ini` file in the same way and order as described in Table 3-6 for the workspace file. Thus it is possible to prescribe some global legend presets for all users that can be amended by each user's individual `legend.ini` file.

Legend presets can be imported and exported separately by using the according commands in the Tools -> User Settings submenu. Read more about the handling of legend presets in Section 3.4.4.

### 3.4.2 Display Settings

The general display settings can be configured in the display settings dialog which is shown in Fig. 3-12. This dialog is opened by choosing the menu entry Tools -> Display Settings.... Its configuration is described in Table 3-7.

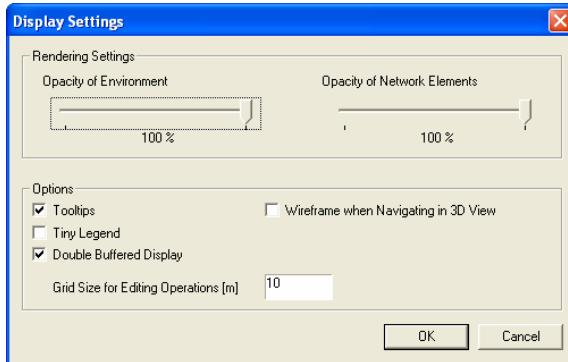


Fig. 3-12 Display settings dialog

Table 3-7 Display settings options

| Parameter                            | Description  |
|--------------------------------------|--|
| <b>Rendering Settings</b>            |  |
| Opacity of Environment               | The opacity (alpha blending factor) of the environment elements (polygons) can be smoothly changed between 0% and 100%.  |
| Opacity of Network Elements          | The opacity (alpha blending factor) of the network element symbols can be smoothly changed between 0% and 100%.  |
| <b>Options</b>                       |  |
| Toolips                              | Tooltips are displayed if this option is activated.  |
| Double Buffered Display              | This feature can be activated (which is recommended) to improve the viewing of tooltips.   |
| Wireframe when Navigating in 3D View | For larger areas it is recommended to use this feature when rotating/navigating in 3D views. Then only the wireframes of the environment objects are shown during movements. The display of rays is completely suppressed in this case.<br>The status of this flag is persistently stored in the registry. |
| Tiny Legend                          | The black legend is displayed with half the usual width.   |
| Grid Size for Editing Operations     | The grid size in meters used for all editing operations (important for e.g. creating and editing transmitters and receivers).  |

The alpha blending factor (or opacity) defines how transparent the respective items shall be drawn in front of lower layers. An alpha blending factor of 0% means total transparency (invisible items); a setting of 100% stands for full opacity – the items are displayed completely non-transparently.

### 3.4.3 Layer Settings



Besides the general display settings, each layer has its own settings for presentation. The layer settings dialog summarizes those settings. It has a subdialog for a flexible configuration of the color scale for the layer. Both dialogs can be quickly accessed by

clicking into the respective sensitive areas in the black legend of the currently active layer as apparent from Fig. 3-13.

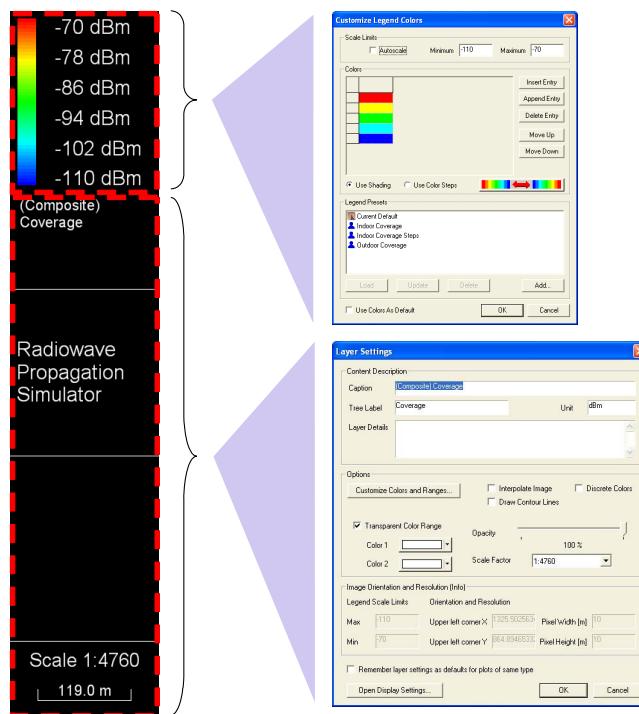


Fig. 3-13 Quick access to the layer settings

The layer settings dialog (see Fig. 3-14) can also be opened by selecting the entry Settings... from the respective layer item's context menu or by pressing the icon for the currently active layer. Its parameters are explained in Table 3-8.

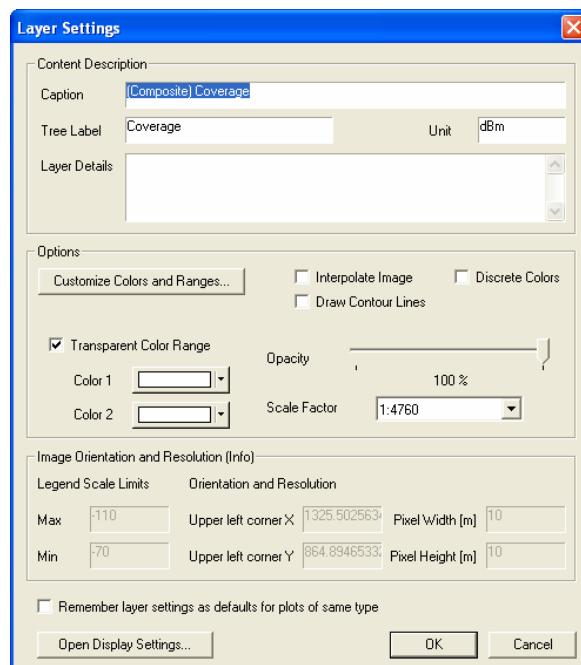


Fig. 3-14 Layer settings dialog

Table 3-8 Layer settings parameters

| Parameter  | Description   |
|--|---|
| <b>Content Description</b>                                 |   |
| Caption  | The layer caption. This name is shown in the black legend in the viewing area.  |
| Tree Label   | The label of the layer item in the tree window.   |
| Unit   | The unit applied for the layer. If this field is left empty, no color scale will be shown in the upper part of the black legend. If the color scale shall still be shown, a white space should be entered for the unit.   |
| Layer Details  | Additional comments for the layer. They are displayed in the lower part of the black legend.  |
| <b>Options</b>   |   |
| <a href="#">Customize Colors and Ranges...</a>             | A button to open the color palette dialog.  |
| Interpolate Image  | A checkbox to choose interpolation between adjacent pixels of a layer.  |
| Draw Contour Lines   | A checkbox to display contour lines instead of the actual layers. The number of contour levels and their colors are determined by the color palette of that layer (see Section 3.4.4).  |
| Discrete Colors  | A checkbox to apply a discrete color representation to the layer. This feature is useful for assigning each cell a dedicated color for mapped surface plots (plots with cell-specific values mapped onto the best server cell areas). On the contrary, for normal surface plots with specific values in each pixel, this option should not be used. |
| Transparent Color Range                                    | A checkbox to apply a transparent color range (spun between two colors) to a layer. All pixels of the current layer that belong to the specified color range according to the active color palette are presented transparently.   |
| Color 1/2  | The two colors that span the transparent color range for the layer presentation. Make sure that the colors are not too different – otherwise the color space could not be properly delimited.   |
| Opacity  | A slider to define the opacity (alpha blending factor) between 0% and 100% for the layer.   |
| Scale Factor   | The scale factor of the layer. The displayed scale factor is automatically adjusted whenever the display is rescaled by a zoom operation. It can be set to some dedicated scale factors by using the pre-defined list field entries. The latter option is recommended for true-to-scale printing of the viewing area.                               |
| <b>Image Orientation and Resolution (Info)</b>             |   |
| Legend Scale Limits  | The legend scale limits are shown here for information. They can be modified in the color palette dialog.   |
| Orientation and Resolution                                 | The position of the upper left corner and the pixel resolution of the layer are given here for information.   |
| Remember layer settings as defaults for plots of same type | If this checkbox is active when the <b>OK</b> button of the dialog is pressed, then the settings of the current layer are stored in the workspace. In the sequel, all plots of the same type will adopt these layer settings.   |
| <a href="#">Open Display Settings</a>                      | A button to open the general display settings dialog.   |

Several layers can be overlaid and displayed simultaneously. To this end, the alpha blending factor (opacity) and transparency of a color range can be controlled for each layer individually. By reducing the opacity of a layer it becomes more and more transparent such that lower layers can shine through.

Furthermore, certain colors of a layer can be set completely transparent by defining a proper transparent color range. Then lower layer get visible at the respective pixels. The transparency feature can also be used to selectively mask a certain value range of a parameter which is very helpful especially for the analysis of network related data in a surface plot.

If the checkbox Remember layer settings as defaults for plots of same type is active when the layer settings are confirmed by clicking the OK button of the dialog, these settings are stored in the workspace as template for layers of the same type. This way the user can define his own layer appearance settings and store them in his workspace. As an example, the user may define special color palette settings for a Coverage plot and remember those settings. Whenever a new Coverage plot will be created after that, the previously stored layer settings will be retrieved from the workspace and are applied for the new layer.

Note that the layer settings include the color palette definition which is described in the following Section. All user-defined layer definitions are added to the workspace, which is automatically saved on shutdown of RPS. So all user-defined layer settings are retrieved when RPS is restarted next time.

### 3.4.4 Color Palette

Data in surface plots is displayed using a color palette. A value at a certain position is shown in that color that corresponds to its scaling in the color palette between the maximum and minimum values.

#### 3.4.4.1 Inspecting the Color Palette of a Layer

The color palette of the currently active layer can be configured in the color palette dialog that is shown in a separate window when clicking the Customize Colors and Ranges... button, refer to Fig. 3-15. Alternatively, a quicker access to this dialog is provided by clicking into the upper part of the black legend area where the color scale is shown (see Fig. 3-13). The usage of this dialog is described in Table 3-9.

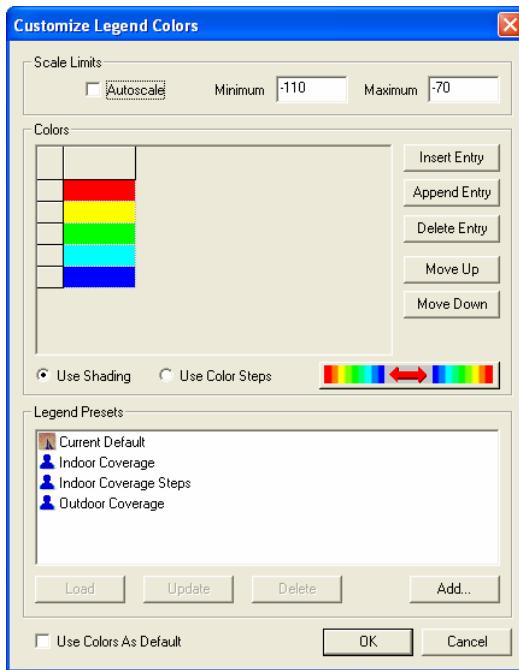


Fig. 3-15 Color palette dialog

Table 3-9 Color palette configuration

| Parameter   | Description  |
|---|--|
| <b>Scale Limits</b>   |  |
| Autoscale   | A flag to automatically retrieve the scale limits from the extreme values of the current layer.  |
| Minimum   | The minimum value of the scale limits.   |
| Maximum   | The maximum value of the scale limits.   |
| <b>Colors</b>   |  |
| <u>Insert Entry</u>   | Inserts a new color entry in the color palette below the currently marked entry.   |
| <u>Append Entry</u>   | Appends a new color entry at the end of the color palette.   |
| <u>Delete Entry</u>   | Deletes the currently marked color entry. The same function can also be performed by typing <Delete>.  |
| <u>Move Up</u>  | Moves the currently marked color entry one step up.  |
| <u>Move Down</u>  | Moves the currently marked color entry one step down.  |
| Use Shading   | A radiobutton to use a shaded color palette. The colors of the color palette are equally spaced between the scale limits. Moreover, values between the discrete interpolation values of the palette colors are smoothly shaded between the corresponding colors.                         |
| Use Color Steps   | A radiobutton to use discrete color steps. The distinct colors of the color palette each cover certain value ranges that are defined by giving a minimum value for each color. The colors are not shaded; instead, the discrete colors are used for all values in the respective ranges. |
|  A button to mirror the currently defined color palette. |  |

| Parameter                    | Description  |
|------------------------------|--|
| <b>Legend Presets</b>        |  |
| <u>Load</u>                  | Loads the selected color legend preset into the color palette table in the upper part of the dialog.   |
| <u>Update</u>                | Updates the currently selected user-defined color legend preset with the current contents of the color palette table in the upper part of the dialog.                |
| <u>Delete</u>                | Deletes the currently selected user-defined color legend preset.   |
| <u>Add...</u>                | Adds a new user-defined color legend preset. All current color palette settings are stored in the new preset.  |
| <u>Use Colors as Default</u> | If this flag is selected when the dialog is closed by clicking the <u>OK</u> button, the currently active color palette is stored as default in the user's registry. |

In case the option Use Color Steps is chosen, the spacing between the discrete color ranges can even be non-linear. In order to specify the minimum values for each color range, another column appears to the right of the colors. This situation is shown in Fig. 3-16.



Fig. 3-16 Defining an individual color scale with discrete colors

Each color of the color palette can be modified by clicking on the corresponding color field in the table and choosing the desired color from the appearing color dialog, see Fig. 3-17.

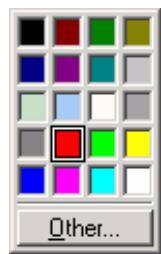


Fig. 3-17 Color dialog

#### 3.4.4.2 Using Legend Presets

In order to ease the definition of a color palette, the dialog offers some pre-defined color palettes in the Legend Presets pane. There are two types of legend presets as shown in Table 3-10. The preset "Current Default" contains the color palette settings that were latest stored in the user's registry when having the Use Colors as Defaults flag being set.

Table 3-10 Legend preset types

| <i>Symbol</i>   | <i>Legend Preset Type</i> |
|---|---------------------------|
|  | Current default           |
|  | User-defined preset       |

A legend preset can be chosen by either double-clicking on the respective entry or by selecting a preset and then press the Load button. All previous manipulations of the color palette will be overridden in that case.

The user can add an arbitrary number of user-defined color legend presets. These settings are persistently stored in the user's registry. This is very useful for often used parameters that shall always be shown with the same scaling (i.e. to compare between plots).

In order to create a new color legend preset, first the desired settings should be adjusted. Then press the Add... button and specify a legend name in the appearing dialog. This name will then be appended in the legend presets list. An existing entry can be updated by pressing the Update button; it can be deleted by pressing the Delete button.

#### 3.4.4.3 Import and Export of Legend Presets

It is possible to store the currently defined legend presets in an ini file in order to retain them for future use. This can be accomplished by using the menu entry Tools -> User Settings -> Export Legend Presets.... The default location for the legend.ini file is the path defined by the %RPSHOME% environment variable. If this variable is not defined, the legend.ini file would alternatively be stored at Documents and Settings\<username>\Application Data\Radioplan\RPS\. However, it is also possible to store a legend.ini file at any other location for special use.

Likewise, a legend.ini file can be imported into RPS by using the menu entry Tools -> User Settings -> Import Legend Presets.... Immediately, the newly loaded legend presets are available as legend templates in the color palette dialog.

When RPS is started, a legend.ini file is searched for in the same order as described in Table 3-6 for the workspace file. Thus it is possible to define some global legend presets in the %ApplicationPath% directory that would be binding for all users. In addition, a user can store his own legend presets at the local places (either %RPSHOME% or Documents and Settings\<username>\Application Data\Radioplan\RPS\).

When RPS is shut down, the current settings of the legend presets are automatically stored in the local legend.ini file.

### 3.5 RPS Project Components and Principles

#### 3.5.1 Environment Objects

While the environment can be described in urban areas by building structures, investigations of indoor wave propagation require the detailed knowledge of the structure within the building. Buildings as well as tables, cupboards, and other obstacles can be modeled by a collection of polygons.

The description of the environment is given by a collection of polygons. The arrangement of each polygon within the coordinate system can be arbitrary; there is no restriction to vertical or horizontal alignment only.

An example of a valid (though simple) environment with skew polygons (roofs) in RPS is given in Fig. 3-18.

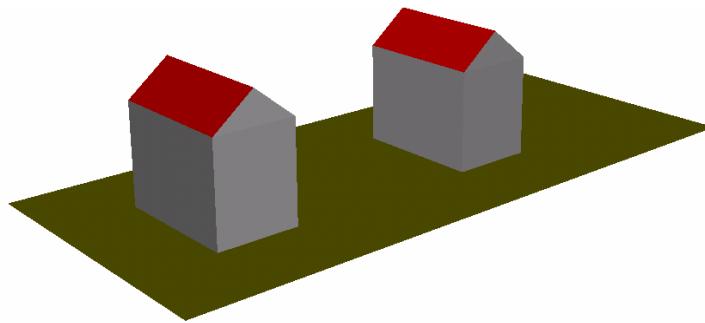


Fig. 3-18 Example of a valid (simple) environment in RPS

For the calculation of the propagation phenomena (reflection, penetration, diffraction) all the properties that are summarized in the material layer definition are necessary. Thus, by assigning a material layer to a polygon, the polygon inherits those properties. See Table 3-3 for a description of the distinct material layer properties.

For indoor environments, windows and doors can be defined by polygons that are situated within the polygons representing the surrounding walls. When the simulator detects more than one obstacle at one position, the obstacle with the smaller dimensions is taken into account. Thus, for instance an open door within a wall can be defined by a polygon with zero thickness and a relative permittivity of  $\varepsilon_r = 1 - j0$ .

### 3.5.2 Diffraction Edges

Before explaining the detection of diffraction edges, the basic influence of diffraction on the prediction results shall be discussed. In indoor environments, diffraction occurs at vertical edges in most cases. The incorporated 2.5D ray tracing algorithms fit well to this dominating behavior with sufficient accuracy. In urban outdoor areas however, it is much more likely that arbitrarily aligned edges cause diffractions. This phenomenon can be best modeled by the implemented 3D ray tracing algorithms.

At the beginning of each simulation, the diffraction edges in the environment are automatically detected. The following conditions must be met for complete and correct detection of diffraction edges:

- Only active material layers are considered for the diffraction edge detection.
- Only material layers with active diffraction flag are incorporated in the detection of diffraction edges.
- Diffraction edges can be detected not only within the same material layer but also between different material layers. The possibly different relative permittivity values of the two polygons at a diffraction edge are correctly registered.

- For the detection of a diffraction edge between two polygons, these planes must have exactly the same edge in common, i.e. the start/end points of this edge must be the same in both polygons. Note that a small shift of max. 1mm is tolerated between two polygon edges in order to still detect them as a diffraction edge. Fig. 3-19 explains this rule in an example.

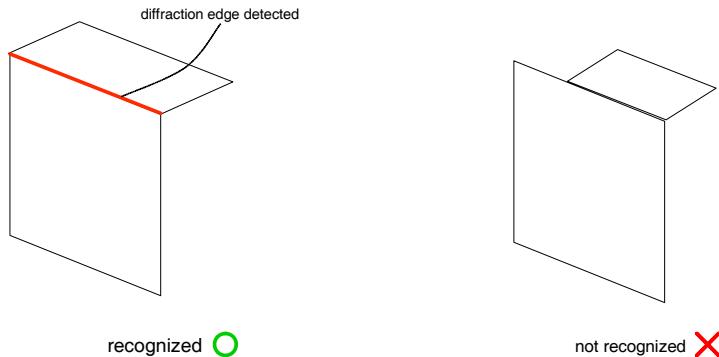


Fig. 3-19 Polygons must have a complete edge in common for a successful diffraction edge detection

- A diffraction edge between more than two polygons with a common edge will *not* be detected.

Diffraction edges are automatically detected before any simulation considering diffraction is performed. All material layers that have the diffraction flag enabled are taken into account in the detection of diffraction edges. During a coverage simulation, diffraction is only performed at such pre-determined diffraction edges.

An example for an environment with detected diffraction edges is given in Fig. 3-20.

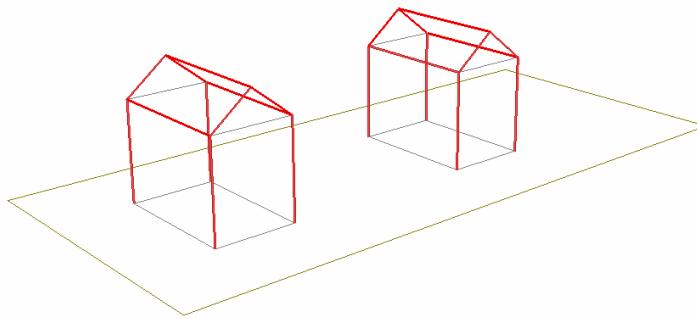


Fig. 3-20 Simple environment with automatically detected diffraction edges (red lines)

### 3.5.3 Network Structure

The configurable network structure in RPS consists mainly of the transmitters (i.e. base stations) with their locations and radio system configuration. Additionally, receivers can be placed at positions where the channel impulse response shall be measured. Thus usually, receivers are arranged along lines or in matrices.

Each transmitter or receiver is assigned an antenna. Antennas are administered by RPS like templates, i.e. one antenna can be assigned to many transmitters/receivers.

All those network elements are shown in the Configuration tab of the tree window. There they can easily be managed, loaded, activated, modified, etc. from their context menu. A sample view of a network configuration overview is given in Fig. 3-21.

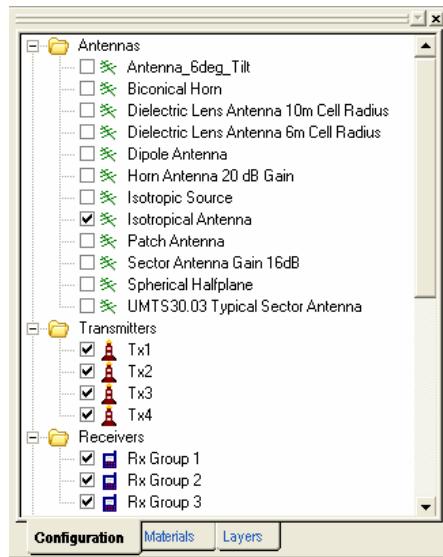


Fig. 3-21 Exemplary network configuration

### 3.5.3.1 Antennas

An antenna is defined by a horizontal and a vertical antenna diagram as well as a gain value. When using the antenna at transmitters or receivers, its 3-dimensional radiation pattern is interpolated from the horizontal and vertical diagrams. Some remarks on the radiation pattern follow in Section 4.2.1.

### 3.5.3.2 Transmitters

An important element of the network configuration is the transmitters. Each transmitter is characterized by a position in Cartesian coordinates and an assigned radiation pattern (i.e. antenna pattern)  $C_s(\vartheta, \varphi)$ . Besides that, a transmitter has some radio properties that are necessary for coverage predictions. All transmitter characteristics are summarized in Table 3-11.

Table 3-11 Transmitter characteristics

| Property          | Description  |
|-------------------|--|
| Name              | A unique identifier of the transmitter.  |
| Description       | An arbitrary description for the transmitter.  |
| Primary Tx        | The superordinate primary Tx for this transmitter.<br>Normally, this field can be left unspecified for pure radio propagation predictions and coverage analyses; thus this parameter has no relevance.<br>In a network, a secondary Tx would just function as a repeater for its primary Tx. |
| Transmit Power    | The transmit power in dB/dBm.  |
| Carrier Frequency | The carrier frequency for this transmitter.  |
| Antenna           | The assigned antenna. Only previously loaded antennas can be assigned.   |

| <i>Property</i>     | <i>Description</i>  |
|---------------------|---|
| Antenna Orientation | The orientation of the antenna with respect to azimuth and elevation.   |
| Position            | The position of the transmitter. Additionally, an altitude (ground level) can be specified.   |
| Active Flag         | If this flag is set, the transmitter is considered as active. Thus, results are shown for this transmitter, and (if the transmitter is obsolete) results are calculated in a new simulation.                        |
| Visible Flag        | If this flag is set, the transmitter is displayed in the 2D/3D views.   |
| Obsolete Flag       | If this flag is set, new results are calculated for this transmitter as soon as another simulation is started. After a complete computation of the coverage from the transmitter, this flag is automatically unset. |
| Cable Losses        | Cable losses for uplink/downlink.   |
| Noise Figure        | The noise figure of the base station receiver.  |

Each transmitter can be set active (default) or inactive. In inactive mode, the transmit power is set to 0 Watt (or  $-\infty$  dB or dBm, respectively). Furthermore, a transmitter can be declared as obsolete. In incremental prediction mode, the field strength is recalculated only for obsolete transmitters.

#### 3.5.3.3 Receivers

Receivers are always combined into Rx Groups. An Rx Group can contain an arbitrary number of receivers. When receivers are placed in the environment (single receiver, along a line, or in a matrix) they must always be assigned an Rx Group. Read more about the placing of receivers in Section 4.2.3.

All receivers use the same radiation pattern with the same orientation. It can easily be switched by checking another antenna item in the Configuration tab of the tree window (see Fig. 3-21).

### 3.6 Working with 2D and 3D Views

RPS provides graphical views for the representation of the environment including network elements in a 2D view (XY mode, top view), and in a 3D view. In both modes, channel prediction results can be presented. Moreover, in 2D view the network configuration can be edited.

For editing the environment, the integrated Environment Editor must be used. It represents a separate module besides the 2D and 3D views. A detailed description on the Environment Editor is given in Section 4.1.

The 2D view allows a true-to-scale output of the plot to any printer supported by Microsoft Windows. Parameters for both 2D and 3D surface plots can be adjusted in the layer settings dialog, discussed in Section 3.4.3.

#### 3.6.1 Adjusting Surface Plots using the Components Toolbar

The components toolbar provides functions to show or hide database components selectively. Furthermore, the fill mode of polygons (especially useful for outdoor views) and the visibility of labels and additional legends can be defined.

The 2D and 3D views can be activated using the corresponding icons from the standard toolbar. Gray icons on the components/paint toolbars indicate that these functions are not available in the active view mode. The distinct database components in the component toolbar are listed in Table 3-12. A sample plot of a 3D view is shown in Fig. 3-22.

Table 3-12 Elements of the component toolbar

| Component | Description   |
|-----------|---|
|           | Toggles the presentation of transmitters.   |
|           | Toggles the presentation of receivers.  |
|           | Toggles the presentation of environment polygons.                                 |
|           | Toggles the presentation of rays (if stored during a simulation).                 |
|           | Toggles the presentation of diffraction edges (if previously detected).           |
|           | Toggles the presentation of attraction areas along with their movement corridors. |
|           | Toggles the filled/unfilled presentation of polygons.                             |
|           | Toggles the presentation of labels at network elements.                           |
|           | Toggles the presentation of additional legends for certain surface plots.         |

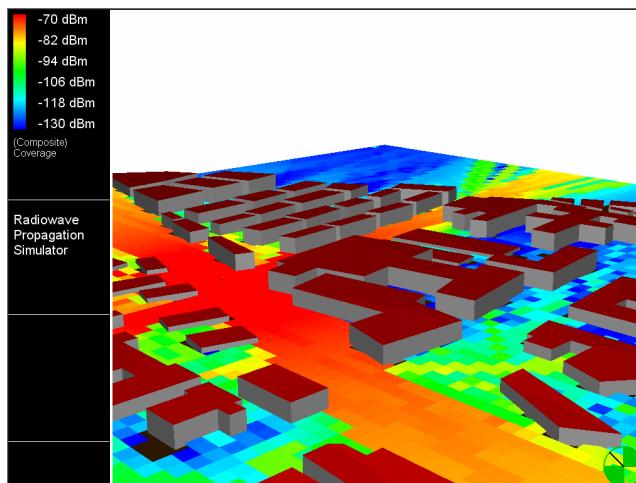


Fig. 3-22 3D surface plot with filled environment objects

### 3.6.2 Zoom Functions

RPS can print 2D and 3D plots in a configurable scale. The scale (zoom factor) can be changed with the menu entries Zoom -> Zoom In (F2) and Zoom -> Zoom Out (F3).

Alternatively, the Zoom In/Out mode can be activated from the paint toolbar. Then zooming in and out is possible by using the left and right mouse buttons. The command Zoom -> Fit to Window (F4) fits the size of the entire environment to the viewing area. This function is also available from the paint toolbar.



In 2D views, it is possible to enlarge regions using the mouse pointer. If the zoom icon in the paint toolbar is checked, the region to be zoomed can be chosen by drawing a rectangle using the mouse pointer. A first left mouse click sets one corner of the zoom rectangle, a second click the other one.

### 3.6.3 Moving Graphics (Pan Mode)



Graphics can be moved in any view using the moving (pan) icon from the paint toolbar. Therefore, the current graphic follows the mouse pointer when the left mouse key is held down.



### 3.6.4 Navigation in 3D Views

In 3D view, a navigation compass is shown in the lower right corner in order to ease the orientation in a 3D environment. The navigation compass always indicates the current perspective (alignment of the camera) in the x-y plane by the direction of the compass needle (black line), refer to Fig. 3-23.

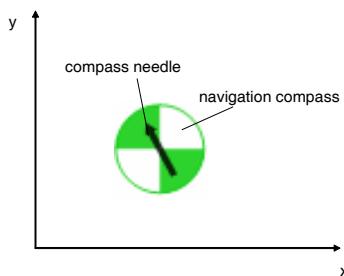


Fig. 3-23 Navigation compass in 3D view

#### 3.6.4.1 Rotation Mode



3D views can be rotated along the axis from the “camera” (the perspective) to the center point of the environment. The rotation mode can be activated by clicking the corresponding icon from the paint toolbar. The scene rotation can be initiated by holding down the left mouse button inside the viewing area and then moving the mouse to rotate the view. As soon as the mouse button is released, the rotation process is stopped.

#### 3.6.4.2 Fly-by Mode



In some cases, it may be useful to observe critical regions of the environment in detail. Therefore, a fly-by mode is integrated in RPS. The fly-by mode is activated by using the corresponding icon from the paint toolbar.

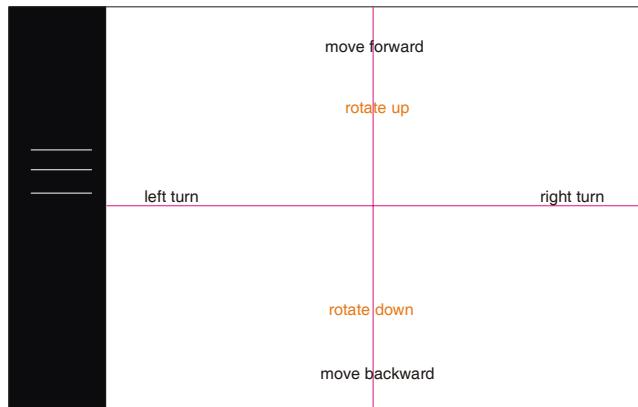


Fig. 3-24 Fly commands

The way the fly-by mode works is easy to understand, if the current view is understood as looking through a camera. This camera is moved through the environment. The commands are given by the mouse pointer in combination with holding down the mouse buttons, as apparent from Fig. 3-24.

The camera is moved forward and backward if the left mouse button is pressed. The vertical movement of the mouse pointer defines the movement direction and its speed. If the mouse is placed on the upper part of the viewing area, the camera is moved forward with a relatively high speed. Near the bottom of the viewing area the camera is moved backward.

To turn left, the mouse pointer must be placed in the left part of the screen. A right turn is done by moving the mouse into the right part of the viewing area.

A rotation can be done in fly-by mode as well. To do that, the right mouse button must be pressed. The camera is rotated up when the mouse pointer is located in the upper part of the screen, a down rotation occurs otherwise.

The speed of any movement or rotation depends on the distance from the center of the viewing area. If the mouse is placed in the middle of the screen, no rotation and no movement occurs.

An example of a detailed perspective in a 3D environment is shown in Fig. 3-25.

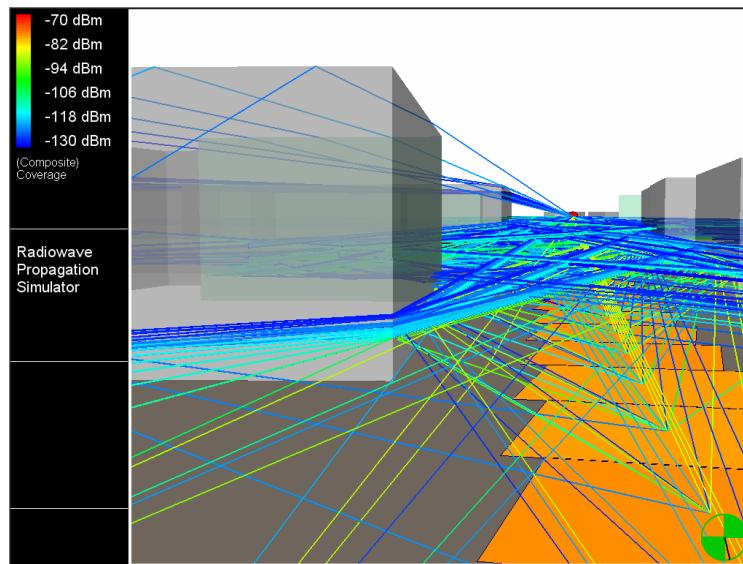


Fig. 3-25 Getting a specific perspective using the fly-by mode in 3D view

#### 3.6.4.3 Key Navigation

In addition to the fly-by mode in the 3D view, a comfortable key navigation through the 3D environment has been introduced. The functions are apparent from Table 3-13. In order to activate the key navigation mode, first click with the left mouse button into the viewing area.

Table 3-13 Key commands for 3D navigation

| <i>Key Command</i>                           | <i>Description</i>  |
|--|---|
| Cursor up <↑>                                | Moves camera position forward.  |
| Cursor down <↓>                              | Moves camera position backward.   |
| Cursor left <←>                              | Moves camera position to the left.  |
| Cursor right <→>                             | Moves camera position to the right.   |
| Page up <PgUp>                               | Moves camera position up.   |
| Page down <PgDn>                             | Moves camera position down.   |
| <Shift> + <↑>, <↓>, <←>, <→>, <PgUp>, <PgDn> | Speeds up movement of camera position (“Turbo Mode”).   |
| <Ctrl> + <↑>, <↓>                            | Turns the camera perspective up or down.  |
| <Ctrl> + <←>, <→>                            | Turns the camera perspective left or right.   |
| <Shift> + <Ctrl> + <↑>, <↓>, <←>, <→>        | Speeds up turning of camera perspective (“Turbo Mode”).   |
| S  | “Save” – Saves the current camera position and angle.   |
| B  | “Back” – Returns to the latest saved camera position and angle.   |
| T  | “Top View” – Moves camera position far above the normal building level and turns its perspective exactly down without changing its x-y position. The previous position is saved such that one can return by typing ‘B’. |

It is easy to keep in mind that the normal key navigation commands can be accelerated (switched to “Turbo Mode”) by additionally holding down the <Shift> key.

### 3.7 Foundations of Geometrical Optics for Electromagnetic Wave Propagation

The simulation algorithms are based on geometrical optics (GO) and the uniform theory of diffraction (UTD). Propagation paths from a transmitter (Tx) position to a receiver (Rx) position are described by rays. In order to ease the understanding of the parameters used in the simulation algorithms, in this Section some basic phenomena affecting wave propagation are discussed.

**Definition:** **Geometrical optics** is an approximate method to describe wave propagation for incident, reflected and refracted fields by means of rays.

Fermat's principle says, that the propagation path is shorter in time compared to adjacent paths in a homogenous medium with constant refractivity  $n$  (e.g. for air  $n = 1$ ). Thus, rays are straight lines and the wave propagation can be described by a finite number of rays, if the number of obstacles is finite too. In practice it is sufficient to determine dominant ray paths, because they contribute the major part of the energy at a specified receiver position.

Each ray is characterized by the horizontal and vertical electromagnetic field strength component assigned in relation to the propagation direction, the excess delay  $\tau$  or the distance  $d$ , the propagation direction  $\vec{r}$  itself and the polarization  $\vec{p}$ . If changes of the polarization are taken into account, the model is called fully polarimetric.

Assuming that electromagnetic waves are transmitted from a point source, the received power can be calculated using the following equation derived from Friis' formula for (undisturbed) free space loss in logarithmic scale:

$$\frac{P_r}{[dB]} = \frac{P_t}{[dB]} - 32.5 - 20 \cdot \log_{10} \left( \frac{f}{[GHz]} \right) - 20 \cdot \log_{10} \left( \frac{d}{[m]} \right) + \frac{G_t}{[dB]} + \frac{G_r}{[dB]} \quad (3.2)$$

Generally, the direction of outgoing rays or the angles of arrival at receiver positions are not identical with the main lobes of transmitting and receiving antennas. The received power is weighted additionally with the radiation patterns of the transmitting and receiving antennas  $C_t(\vartheta, \varphi)$  and  $C_r(\vartheta, \varphi)$ .

In addition to the free space loss, electromagnetic waves are reflected at obstacles, attenuated by penetrated walls or windows and diffracted at edges. The attenuation resulting from reflection and penetration can be determined through incident angle, polarization, material characteristics and the thickness of the current obstacle. Diffraction edges are comparable with line sources that define an outgoing point of new rays. The higher the carrier frequency, the lower the influence of diffraction. The handling of the propagation phenomena reflection, penetration, and diffraction as well as the related algorithms are described in the following Subsections. Incoherent scattering at rough surfaces is not considered in the current version.

#### 3.7.1 Reflection

At an obstacle, i.e. at a dielectric boundary, a part of the incoming electromagnetic wave front is reflected. The relation of the field strength of the reflected field  $\underline{E}_r$  to the incident

field  $E_i$  can be described by the reflection coefficient  $\Gamma$ . The physical boundary conditions require a separate determination of the field components that are oriented in parallel or perpendicularly to the incident plane. For skew incident angles a deformation of the polarization  $\vec{p}$  can occur. Additionally, the reflection coefficient depends on the incident angle  $\alpha$ . In Fig. 3-26 and Fig. 3-27 the reflection coefficient is shown as a function of the incident angle with the permittivity as a parameter.

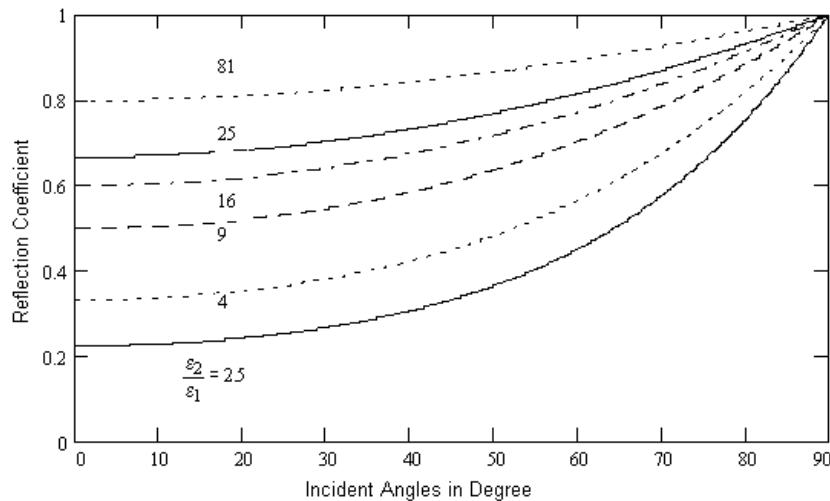


Fig. 3-26 Reflection coefficient of the perpendicular field component versus incident angle (parameter: relative permittivity)

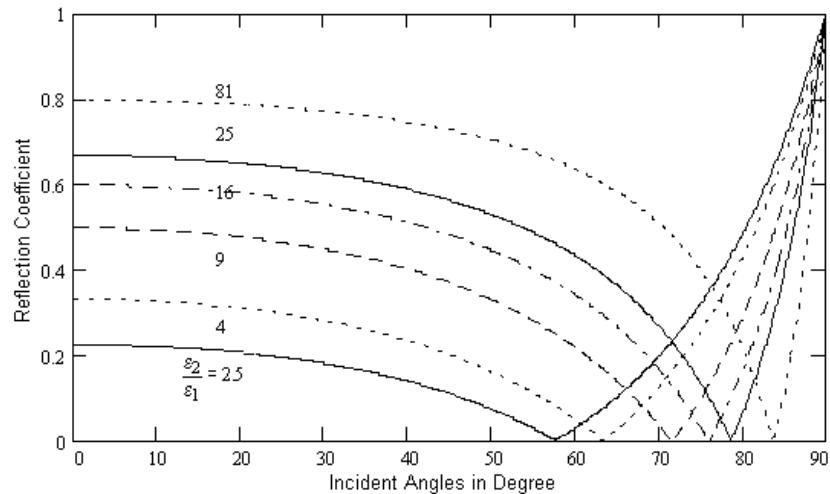


Fig. 3-27 Reflection coefficient of the parallel field component versus incident angle (parameter: relative permittivity)

Fig. 3-27 also shows angles, where the reflected parallel field component vanishes. These angles are called Brewster angles. Dielectric properties of materials can be determined for instance by measuring the Brewster angles.

If obstacles are small compared to the wavelength, the reflection coefficients depend on their thickness. The simulation algorithms implemented in RPS consider the polarization

dependency as well as the thickness of obstacles in the calculation of the reflection coefficient  $\Gamma$ . Dielectric properties of common materials are given in the appendix.

The reflection coefficient essentially depends on the real part of the permittivity.

### 3.7.2 Penetration

As for the reflection coefficients, also the penetration coefficients are determined separately for the parallel and the perpendicular field components in relation to the incident plane. The penetration can be determined by combining the loss within the obstacle and the penetration losses at the dielectric boundaries air – medium and medium – air. For walls and comparably heavy obstacles the penetration loss is dominated by the penetration loss within the medium.

The penetration loss within an obstacle mainly depends on the ratio of the imaginary part and the real part of the permittivity.

Due to the higher permittivity in denser materials the velocity of electromagnetic waves is lower than in free space. However, the resulting additional delays are so small, that they may be neglected. The local changes in the propagation direction at dielectric boundaries due to refraction are not considered.

### 3.7.3 Diffraction

Electromagnetic waves are diffracted at edges. The mathematical formulation of the diffraction phenomenon is given by the laws of the uniform theory of diffraction (UTD). Simplified, a diffraction edge can be interpreted as an outgoing point of electromagnetic waves or, in other words, it can be treated as a new transmitting source. Magnitude and phase of outgoing waves are determined through the diffraction coefficient  $D$  which can be calculated from the geometrical boundary conditions (incident angle, polarization, permittivity of edge materials). The free space loss of diffracted waves is given by  $\sqrt{d}$ , where  $d$  represents the distance from the diffraction point.

With higher frequencies  $f$ , the influence of diffracted waves is reduced.

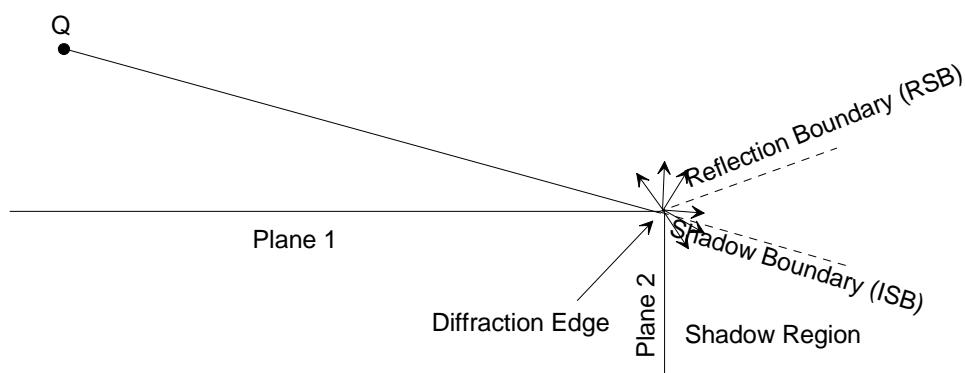


Fig. 3-28 Definition of an diffraction edge by two surfaces

### 3.7.4 Polarization

The direction of the outgoing field components (in our case represented by rays) are defined by the pointing vector  $\underline{S} = \underline{E} \times \underline{H}$ , where  $\times$  denotes the cross product. The direction of the E field vector defines the polarization. Generally, the E field components can be divided in a horizontal and vertical component in relation to the propagation direction. For a better understanding, let's assume that the antenna is located at  $P_{Tx}(x = 0, y = 0, z = 0)$  and we have an outgoing wave with the direction of the pointing vector identical to the x-axis.

$$\underline{E}(t, x) = E_y(t, x) \cos \omega t + E_z(t, x) \cos(\omega t + \delta) \quad (3.3)$$

The constant parameter  $\delta$  represents the phase difference of the E vectors. If  $\delta = 0$ , we have a linear polarization. The E field is horizontally polarized (HP), if  $E_z$  is zero.

Otherwise, we have an ideal vertical polarization (VP), if  $E_y$  is zero. If  $\delta = 90^\circ$ , we have left hand circular polarization (LHCP), and for  $\delta = -90^\circ$ , the E field right hand polarized (RHCP). Generally, every E field vector can be decomposed in a linear and a circular component. The change of the polarization state is called depolarization which can occur for instance at a reflection point. If a horizontal ray representing an incident field with ideal LHCP hits a vertical wall, the reflected field is RHCP then. Of course, the degradation of polarization depends on the permittivity.

RPS assumes that an antenna is polarized either VP, HP, LHCP or RHCP. At the receiving antenna, only those field components are considered that match to the antenna's polarization state.

### 3.7.5 Ray Launching Approach

The RPS ray tracing simulation algorithms are based on methods from geometrical optics. The task of the simulator is to determine the propagation paths that can contribute to the signal at a given receiver position. Besides the geometrical calculation, the propagation loss has to be determined for each ray separately.

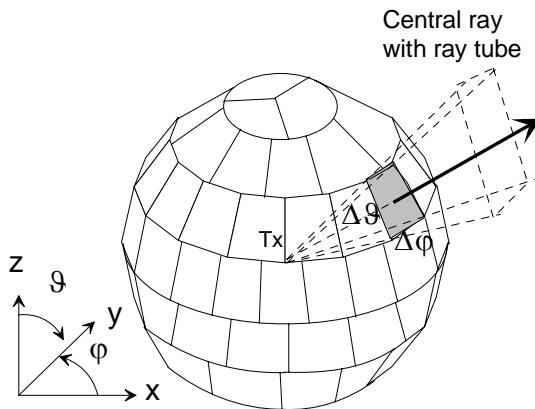


Fig. 3-29 Launched ray from a transmitter including their ray tube

In the ray launching method, a finite number of rays are launched from a transmitter position. At obstacles, the rays are divided in a reflected and possibly a penetrating part. Each ray is traced, until a given maximum of the pathloss is exceeded. Since in practice

only a finite number of rays can be launched, each ray represents a ray tube (Fig. 3-29). Rays launched in adjacent ray tubes diverge with increasing distance from the transmitter position. Thus, the spatial resolution is decreased. The effect can be minimized by ray splitting which keeps the resolution of the ray tubes approximately constant.

If a receiver is located within a ray tube, the complex E field components, excess delay, polarization and incident angles of that ray are stored at this receiver. The outcome of the propagation calculations is a detailed channel impulse response  $\underline{h}(x_i, y_i, z_i, \tau)$  for each Tx-Rx pair. The narrowband and broadband formulation of the radio channel can be derived from these impulse responses.

### 3.8 Plug-in Support

RPS provides an open interface for implementation of user-defined functionalities. This opportunity expands the usability of RPS dramatically, sometimes far beyond radio channel prediction only. The usage is very easy, i.e. custom functionalities can be integrated in a simple plug-in way.

The RPS plug-in interface enables the implementation of user-defined functions for:

- customized empirical and semi-empirical prediction models (see Section 5.5), and
- user-defined graphical surface plots for the presentation of data derived from prediction results, i.e. channel impulse response data including angular information (see Section 6.4)
- user-defined stand-alone application plug-ins (see Section 6.8) for further post-processing of result data at a position (point analysis), along a path (path analysis), or for a selected transmitter (Tx analysis)

by simply writing some lines of code. All available plug-ins for propagation models and surface plots are listed below the network configuration in the Configuration tab of the tree window. Fig. 3-30 shows an exemplary view with some plug-ins.

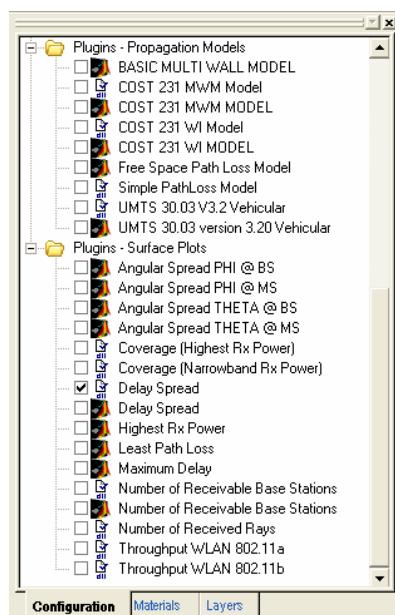


Fig. 3-30 Exemplary plug-ins in the Configuration tab of the tree window

User-defined plug-ins for propagation models and surface plots can be written in Visual C++ 6 as dynamic link libraries (DLLs), or they can be implemented in the MATLAB scripting language (m-files). Moreover, MATLAB stand-alone applications for point/path/Tx analysis are possible that have been compiled previously using the MATLAB compiler. In general, DLL based plug-ins work much faster in most cases than the processing of m-files by MATLAB, where MATLAB itself is working as an ActiveX automation server in the background. On the other hand, the MATLAB scripting language is one of the most powerful scripting languages. Furthermore, a MATLAB script can be written, modified, and debugged conveniently using the built-in script editor of the RPS framework.

MATLAB based plug-ins can be used if MATLAB is installed in addition to RPS. MATLAB can be purchased from The Mathworks Inc. ([www.mathworks.com](http://www.mathworks.com)) or through any local reseller.

RPS is able to call one of the following MATLAB versions as an engine: R11, R12, R12.1 or R13. Due to internal incompatibilities between the C/C++-libraries of the different MATLAB versions, a separate edition for each of them is provided. The MATLAB version to be used can be configured in the `rps.ini` file, see Section 1.8.

All plug-ins, located in the `%ApplicationPath%\RPS\plugins\` directory or any subdirectory, are automatically loaded when RPS is started.

### 3.8.1 Plug-in Directories

When RPS is executed, it searches for all valid dynamic link libraries (DLLs) and MATLAB scripts in the `%ApplicationPath%\RPS\plugins\` directory or any subdirectory. The plug-ins must be stored in the following subdirectories:

- `rps_pm_dll`: All propagation models, realized as DLLs
- `rps_pm_matlab`: All propagation models, realized as MATLAB scripts
- `rps_sp_dll`: All custom surface plot plug-ins, realized as DLLs
- `rps_sp_matlab`: All custom surface plot plug-ins, realized as MATLAB scripts
- `rps_pointanalysis`: All custom plug-ins for point analysis (MATLAB stand-alone applications)
- `rps_pathanalysis`: All custom plug-ins for path analysis (MATLAB stand-alone applications)
- `rps_txanalysis`: All custom plug-ins for Tx analysis (MATLAB stand-alone applications)

Also the shipped plug-ins are located in these directories. It is possible to create subdirectories below the given structure. Thus, the user can build his own structure of subdirectories. However, it must be assured that any user-defined plug-in will be placed in one of these directories or any subdirectory in order to be properly identified.

### 3.8.2 Attaching and Detaching the MATLAB Engine

Before MATLAB scripts can be executed, the MATLAB application (called engine) must be attached. RPS attaches MATLAB automatically before the first m-file is executed.

Alternatively, the engine can be attached and detached manually, using the corresponding items from the Tools menu. If no MATLAB engine is available, an according message appears in the message window on start of RPS, and the corresponding entries in the Tools menu are gray.

The communication between RPS and MATLAB is done using the Components Object Model (COM), introduced by Microsoft. Any COM application (MATLAB as well) can be executed on a different computer using the DCOM (now COM+) communication protocol. The configuration is done by the `dcomcnfg.exe` application, which is part of Windows NT or higher and is available for Windows 95/98/Me from Microsoft. Please refer to the documentation for details.

### 3.8.3 Working with Plug-ins

In the following, the general usage of user-defined plug-ins for propagation models and surface plots is described since these types of plug-ins are handled in a comparable way. The third type of plug-ins for specific result analysis is presented in Section 6.8.

#### 3.8.3.1 Plug-in Dialog

The user can modify all available plug-ins for propagation models and surface plots in the RPS Plug-in Dialog that can be opened by clicking the Tools -> Plugins... menu entry.

Alternatively, it can be opened by double clicking a certain plug-in item in the Configuration tab of the tree window or by choosing Settings... from its context menu. All attached plug-ins are presented in a tree view sorted by function and type, see Fig. 3-31. Plug-ins can be selected, custom parameters of DLL plug-ins can be adjusted, or the m-files can be modified directly within the RPS framework.

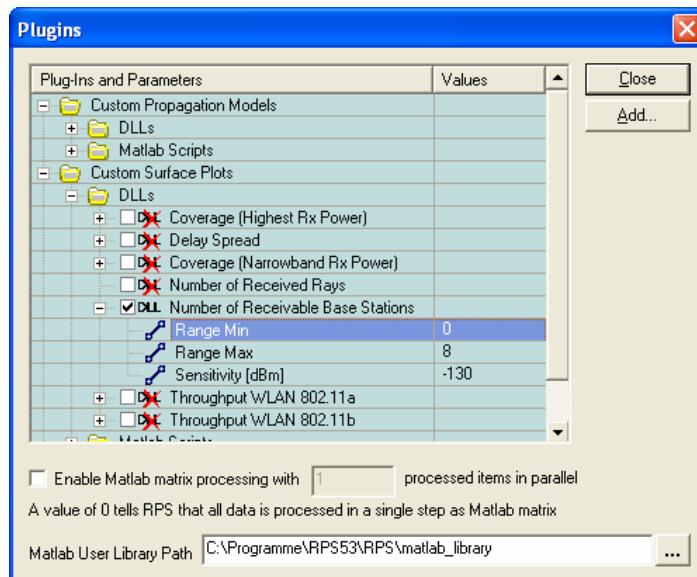


Fig. 3-31 RPS Plug-in Dialog

The root of the tree view contains the folders *Custom Propagation Models* and *Custom Surface Plots*. Both folders contain subfolders for *DLLs* and *Matlab Scripts*. A third main

folder collects all m-files, the type of which cannot be determined, e.g. due to an invalid syntax in the header information.

### 3.8.3.2 Selecting Plug-ins

Using the checkbox left of the DLL or MATLAB icon in either the plug-in dialog or directly in the Configuration tab of the tree window, one propagation model or surface plot plug-in can be set active at a time. These plug-ins are used for custom field strength predictions or for custom surface plots, respectively.

Remember: Only one plug-in can be set into active state at a time!

### 3.8.3.3 Editing Custom Parameters of DLL Plug-ins

All DLL based plug-ins contain a subtree, which allows the user to adjust custom parameters. Custom parameters are useful to modify variables such as e.g. pathloss exponents or sensitivity values in custom propagation models. A detailed description for the implementation of custom parameters into user-defined plug-ins is given in Section 5.5.2.1 for propagation model plug-ins and in Section 6.4.2.1 for surface plot plug-ins. Right of each custom parameter caption, the corresponding value is shown. The values can be modified in this dialog directly. A double click on the value entry activates the edit mode. The edit mode is left with a single mouse click outside the edit area.

Do not use the <Enter> key to switch into non-edit mode! This would cause a line break within the edit field.

### 3.8.3.4 Editing MATLAB Scripts

All MATLAB script based plug-ins contain a subitem with the caption “Double Click to Edit Script”. By double clicking this subitem, a powerful script editor appears as shown in Fig. 3-32. It allows the modification of m-files directly within the RPS framework. Alternatively, any other editor could be used as well.

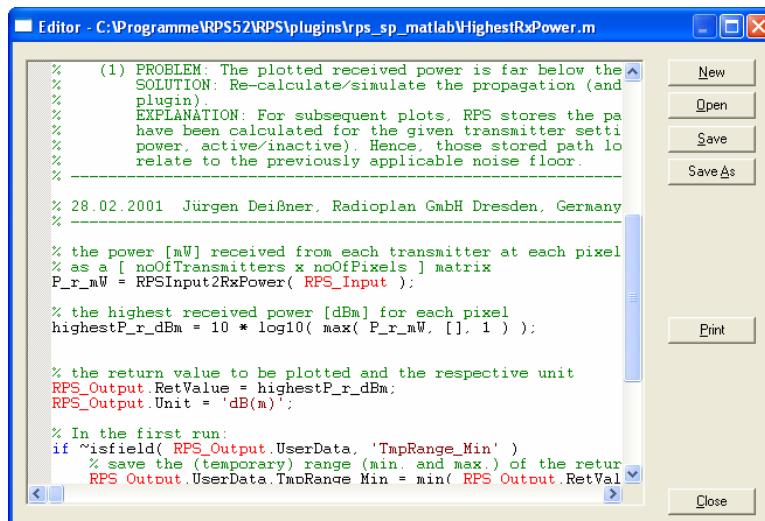


Fig. 3-32 Script Editor Window

The editor supports all state-of-the-art functions like printing support, unlimited undo/redo, syntax highlighting etc. The input and output parameters **RPS\_Input** and **RPS\_Output** are highlighted in red color. These MATLAB structure arrays are used for data exchange between the m-file and RPS. A detailed description of the MATLAB plug-in API is given in Section 5.5.2.2 for propagation model plug-ins and in Section 6.4.2.2 for surface plot plug-ins.

The size of this window can be changed appropriately. If a script is saved under a different filename, the new plug-in is not added automatically to the plug-in dialog. This must be done manually using the Add button.

### 3.8.3.5 Matrix Processing and MATLAB User Library Path

The execution time of surface plot plug-ins can be increased dramatically, if a block of matrix operations is done in one step. If a surface plot m-file supports matrix processing, the maximal number of items that are processed in parallel can be adjusted in the fields located under the tree view of the plug-in dialog. The optimal number with regard to processing speed depends on the available memory and the algorithms used in the corresponding m-file. If the number of simultaneously processed items is set to zero, RPS executes the script only one time with the whole data available.

Some plug-ins of the current release of RPS require general functions, that will pre-process the **RPS\_Input** structure array. These functions are stored in the %ApplicationPath%\RPS\matlab\_library\ directory. The path to this directory is set by default. It can be adjusted or extended using the Matlab User Library Path entry. This path is added to the path environment variable in MATLAB, when the MATLAB engine is attached, or the path is changed.

## 4 Working with RPS — Configuration

In this Chapter, the configuration of a complete RPS project is described. For each project, both the environment and the network structure must be configured.

For the environment, RPS provides an integrated Environment Editor whose usage is explained in Section 4.1. The network structure consists of transmitters and receivers along with their associated antennas. These elements can be configured in the 2D view which is described in Section 4.2.

### 4.1 Environment Configuration



RPS provides a comfortable editor for a graphical construction and modification of environment structures – the Environment Editor. It can be entered by clicking the corresponding icon in the standard toolbar. The Environment Editor contains many drawing and editing functions the usage of which is similar to AutoCAD.

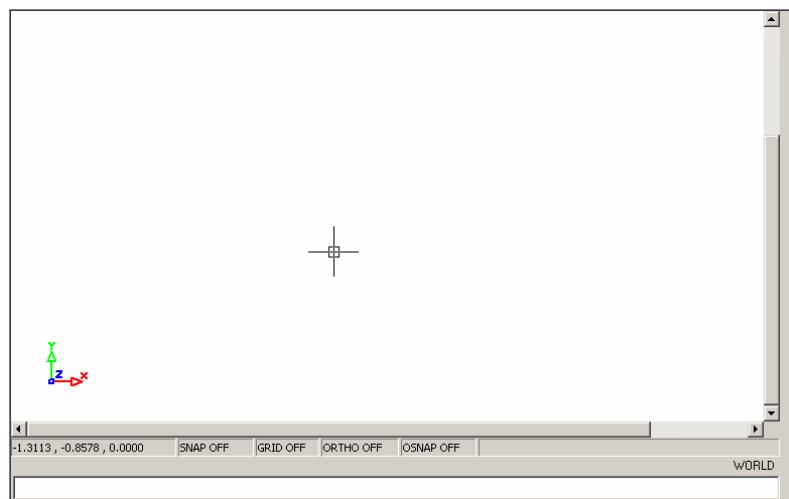


Fig. 4-1 Edit window of the Environment Editor in the viewing area

Generally, the modification of the environment has to be done in the integrated Environment Editor.

As apparent from Fig. 4-1, the Environment Editor provides a special edit window with a separate mouse cursor and a coordinate system indicator in the lower left corner. Below the editing area there is a status bar showing the coordinates of the mouse cursor and some additional status information of several flags. In the next line, the current coordinate system

of the Environment Editor is displayed at the right side (e.g. 'WORLD'). The text field at the bottom of the edit window is the command line that enables the input of commands and coordinates directly from the keyboard, refer to Appendix 10.2.1.

Above and below the edit window, two toolbars are shown in the Environment Editor mode. They provide all available functions and editing commands of the Environment Editor. See Appendix 10.2.2 for a detailed description of the toolbars.

#### 4.1.1 Functionality Overview

The integrated Environment Editor embraces a variety of functions and editing commands that make it a powerful tool for creating complex building structures including additional obstacles as persons, cars, trees, etc. In particular, the following functionalities are offered:

- commands for easily creating building components and additional obstacles
- usual functions for modifying an environment (scale, move, copy, remove, etc.)
- special functions for automated insertion of typical obstacles in clusters (e.g. persons, cars, trees, etc.)
- full support of the AutoCAD layer system including electrical properties (material layers)
- import and export of DXF/DWG files
- integrated stencil library for eased insertion of typical environment components and obstacles
- object snap for eased editing of environment
- different view perspectives and zoom functions
- supplementary functions (e.g. for adding flat roofs or detecting diffraction edges)

##### 4.1.1.1 General Command Usage

Generally, a command in the Environment Editor is performed in a sequence of several steps. Some steps can be omitted depending of the type of command. The current status of a command or the next action to be performed are indicated in the row above the command line. A pending command can be interrupted at any time by pressing the <ESC> key. The last command can be re-invoked by clicking the right mouse button. The general steps of a command are given in the following:

- Choice of a proper view / coordinate system by clicking one of the view icons from the upper toolbar. Usually the WORLD coordinate system should be chosen that uses absolute coordinate. All other views use their own local coordinate system in the perspective plane.
- Choice of a material layer to be used for new/copied objects (*optional*). This step is important for all commands where new objects are inserted into the environment since the selected material layer is assigned to them. The material layer can be chosen by simply clicking on the according layer item in the Materials tab of the tree window. The currently active layer is indicated by the colored icon (e.g. ).
- Selection of the command (by clicking an icon from the toolbars or typing the command name in the command line).

- Selection of objects that shall be modified by the command (*optional*). Normally, object can be selected by clicking on them with the left mouse button. The object snap can help locating objects. If no object is clicked, the editor automatically switches to the window select mode where a drag box can be drawn around objects to be selected. Undesired objects can be unselected by clicking them with the left mouse button while holding down the <Shift> key. The object selection is finished by either clicking the right mouse button or by pressing the <Enter> key.
- Input of coordinates and dimension arguments for new objects (*optional*). These coordinates can be entered either graphically by left mouse clicks or they can be typed in the command line. Positions are specified by three values (delimited by commas) for the x/y/z coordinates, whereas dimension arguments (e.g. width, height, etc.) are single values. Relative positions (useful for e.g. move operations) can be entered by prepending a position by a '@'.
- For some commands, a dialog appears where some additional parameters must be configured.

Refer to Appendix 10.2 for a tabular overview of all available Environment Editor commands.

#### 4.1.2 Coordinate Systems

The Environment Editor provides several coordinate systems depending on the current view. The WORLD coordinate system is the only one that uses absolute coordinates. It can be chosen by clicking the icon from the upper toolbar. It is always bound to the top view, i.e. the associated view shows the x-y plane from above. The WORLD coordinate symbol is given in Fig. 4-2 a).



Fig. 4-2 Lower left corner of edit window with coordinate system and coordinate display

All other coordinate systems represent local systems with two-dimensional coordinates in the currently presented plane. Especially in the perspective views, the displayed coordinates are related to the projection plane which is skew in a usual Cartesian system. An example of such a coordinate system in conjunction with a South-West view is given in Fig. 4-2 b).

All dimensions in all coordinate systems are measured in meters [m]. This is also the unit of the coordinate display in the lower left corner of the edit window. When entering coordinates as arguments of certain commands in the command line, then these numbers are given in [m], too.

#### 4.1.3 Views

For presentation and editing purposes, the Environment Editor provides several views and related display options. All of them are accessible from the upper toolbar.

#### 4.1.3.1 2D Views

All in all six 2D views are offered by the Environment Editor for top view, bottom view, front view, back view, left view, and right view. They can be activated by clicking the according icons in the upper toolbar.



#### 4.1.3.2 Perspective Views

Moreover, four perspective views are offered by the Environment Editor for South-West view, South-East view, North-West view, and North-East view. They can be activated by clicking the according icons in the upper toolbar.



#### 4.1.3.3 Shading

Three shading options are provided that control the display style and visibility of the objects. These modes can be toggled by clicking the appropriate icons in the upper toolbar. Their functionality is explained in Table 4-1.

Table 4-1 Shading options

| Icon | Description   | Tooltip                |
|------|---|------------------------|
|      | All objects are displayed as full shapes, i.e. they are shown as non-transparent surfaces. Hidden objects are invisible with respect to the current view.   | Shade                  |
|      | Only the borders of the objects are displayed. However, the objects are treated as full shapes where the fill color is always white. Concealed objects are invisible with respect to the current view | Hide Concealed Objects |
|      | Only the borders of all objects are displayed. The objects are transparent, i.e. all borders are visible.   | Wired View             |

#### 4.1.3.4 View Rotation

The complete environment can be rotated around its central point in the rotation mode. It can be entered by clicking the icon (tooltip Dynamic 3D Rotation) in the upper toolbar. The mouse cursor is changed to a special symbol . The environment can be rotated then by clicking and holding the left mouse button into the edit window and moving the mouse into the desired direction. When the mouse button is released, the current view is frozen. Note that also the coordinate system indicator is rotated accordingly.

The rotation mode can be left again by choosing another command or by clicking the right mouse button.

#### 4.1.3.5 Zoom

For navigation in the environment, several zoom functions as well as a pan mode are provided. They can be chosen by clicking the according icons in the upper toolbar.



The left icon (tooltip Zoom to Selection) is the normal zoom to a rectangular part of the current view. The zoom area can be defined by two mouse clicks in the opposite corners of a rectangle.

The next icon (tooltip Zoom to Previous) is for undoing previous zoom actions, whereas the next two icons are for zooming on the complete environment.

In addition to the zoom commands from the upper toolbar, the normal zoom shortcuts <F2> (Zoom In), <F3> (Zoom Out), and <F4> (Fit to Window) as well as the full screen mode <F5> are available. Alternatively, these functions can also be invoked from the according entries in the Zoom menu.

Finally, the pan mode can be used to shift the environment in the current view. It is entered by clicking the according icon (tooltip Pan) in the upper toolbar. The mouse cursor is changed to a special symbol . The environment can be shifted then by clicking and holding the left mouse button into the edit window and moving the mouse into the desired direction. When the mouse button is released, the current view is frozen. The pan mode can be left again by choosing another command or by clicking the right mouse button.

#### 4.1.4 Background Image

As a typical case of editing an environment, buildings shall be created according to a map. For that purpose a background image can be inserted. This function is initiated by clicking the icon (tooltip Insert Image) from the lower toolbar. A file open dialog appears where the image file can be selected. Then the origin (i.e. the insertion position for the lower left corner of the image) and the total dimension in x-direction must be specified in [m]. It is recommended to enter these arguments directly in the command line instead of defining them graphically with the mouse pointer.

For a better positioning of the background image it is possible to accompany the image file with a georeference file (extension \* .tfw). The georeference file contains information about the location and scaling of the image. Both files must have the same name, despite the extensions. RPS uses the georeference information to place the imported background image properly without asking for further arguments.

After all necessary arguments have been specified, the background image is inserted and added to the currently selected material layer. If required, the <F4> key can be pressed to zoom on the entire image. An example of an urban building structure with a background image is shown in Fig. 4-3.

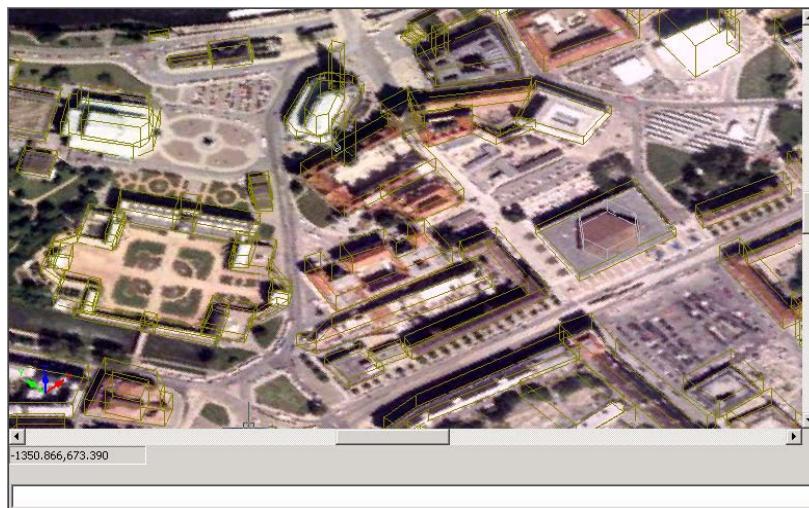


Fig. 4-3 Urban building structure (wired view) with background image

To remove a background image, the normal remove command using the icon from the lower toolbar can be used. This is explained in more detail in Section 4.1.10.4.

#### 4.1.5 Object Snap

The Environment Editor offers several snap functions that ease the editing of the environment. Especially, it could be very difficult to align objects with the mouse. Using the object snap, this task can be done very efficiently in a proper view.

The simplest form of the snap is the snapping grid. It is actually independent of any objects – so it is no real “object snap”. This function can be toggled by double-clicking the area of the status bar below the edit window where ‘SNAP ON/OFF’ is shown. If this snap is switched on, all mouse inputs will be applied to a certain raster. The grid width of that raster can be configured in the Environment Editor preferences. See Section 4.1.11.1 for more information on that.

Despite the snapping grid, all available object snap functions are summarized in the upper toolbar. They provide the possibility to snap on

- object border vertices,
- object border centers,
- object border intersections (with respect to current view),
- nearest object,
- perpendicular lines, and
- insertion point of object blocks

according to their self-explaining tooltips. All these snap options can also be combined, i.e. it is permissible to select several of them simultaneously. In order to switch off all snap options at once, the icon can be clicked.



If one or more snap options are selected, the vicinity of the mouse cursor in the edit window is steadily scanned for objects fitting on the current snap options. If a nearest object could be found, the appropriate element of it (e.g. a border or a vertex) is highlighted as exemplarily shown in Fig. 4-4.

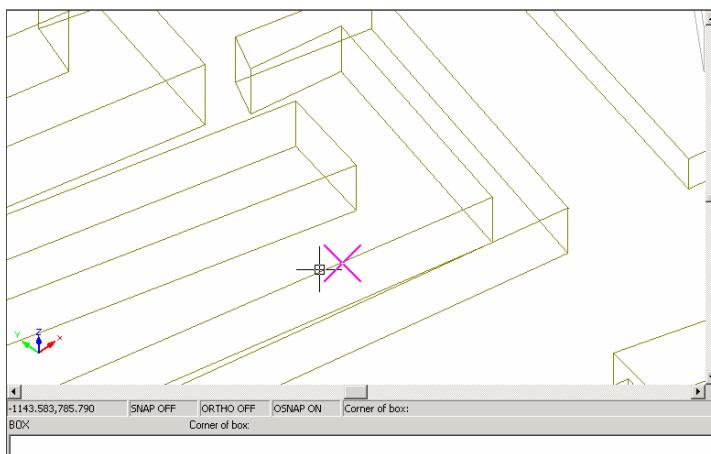


Fig. 4-4 Highlighting the nearest object

As a further support for drawing objects, a so-called “ortho” mode can be switched on. This function can be toggled by double-clicking the area of the status bar below the edit window where ‘ORTHO ON/OFF’ is shown. If this ortho mode is switched on, all mouse inputs can only be done in horizontal or vertical direction in the currently active coordinate system.

#### 4.1.6 Material Layer Definition

Each environment object is assigned a mandatory material layer that defines all relevant material properties of a class of objects. This basic principle is known from applications as AutoCAD. For many editing commands it is important to specify a material layer before the command is executed. New or copied objects would so be assigned the specified layer. This section describes the usage of material layers in RPS as well as the meaning of the properties contained in them.

##### 4.1.6.1 Choosing a Material Layer

All available material layers are listed in the Materials tab of the tree window. There is always a layer “0” which is mandatory. All other layers follow below it. A certain layer can be chosen for subsequent commands by simply clicking on its item in the tree. The currently active material layer is indicated by a colored icon (e.g. ).

It is strongly recommended to never use layer “0” for any objects. Instead, you should always create new material layers for creating or importing objects. These objects will then inherit the material properties of their respective layers.

Moreover, material layers can be selectively activated. Only environment objects of active layers are visible and considered in coverage simulations. The active state of a material layer can be toggled by clicking the checkbox left of the layer item in the tree.

Fig. 4-5 shows an example of the Materials tab of the tree window with several active layers and one chosen for editing commands. The latter one need not necessarily be active.

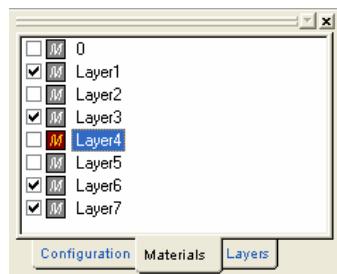


Fig. 4-5 Active and inactive material layers

#### 4.1.6.2 Creating a New Material Layer

A new material layer can be created by right-clicking an arbitrary layer item in the tree window and choosing Add New Layer from the appearing context menu. The new layer is directly added in the layer tree. It has a default name ('Layer*n*') and default settings. These settings can be modified in the settings dialog as described in the next Subsection.

Adding a new layer can be useful e.g. if a roof shall be added to a house skeleton, consisting of the walls only. Then usually the roof shall have different material properties than the walls. Thus a new layer must be created and its properties should be modified as needed. If the new roof layer is currently selected, it will be used for the newly created roof objects then.

#### 4.1.6.3 Inspecting Material Properties

The properties of a material layer can be inspected and modified in the corresponding settings dialog. It can be opened by choosing Settings... from a certain material layer's tree item or, alternatively, by double-clicking the layer item. A dialog appears that summarizes all material properties of the chosen layer, see Fig. 4-6. The distinct parameters are listed in Table 4-2.

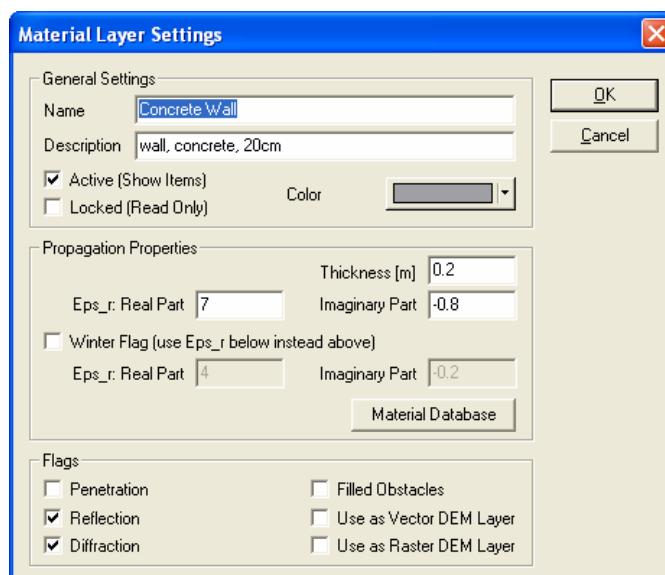


Fig. 4-6 Material layer settings dialog

Table 4-2 Material layer properties

| <i>Property</i>               | <i>Description</i>  |
|-------------------------------|---|
| <b>General Settings</b>       |   |
| Name                          | Name of the material layer. This can be an arbitrary string except "0" (reserved).<br><b>Can be imported/exported in DXF/DWG format.</b>  |
| Description                   | An arbitrary description for the material layer.  |
| Active Flag                   | If this flag is set, the layer is visible and is considered in coverage simulations.<br><b>Can be imported/exported in DXF/DWG format.</b>  |
| Locked Flag                   | If this flag is set, the objects assigned to this layer cannot be selected or modified any longer.<br><b>Can be imported/exported in DXF/DWG format.</b>  |
| Color                         | The presentation color of the material layer.<br><b>Can be imported/exported in DXF/DWG format.</b>   |
| <b>Propagation Properties</b> |   |
| Thickness                     | The thickness of objects of this layer that is considered for propagation effects (reflection, penetration). It is also used as the presentation thickness of vertical objects in the 2D view.  |
| Relative Permittivity         | The relative permittivity $\epsilon_r$ of objects of this layer that is used for propagation effects (reflection, penetration, diffraction).  |
| Winter Flag                   | With this flag an alternative setting for the relative permittivity can be activated.   |
| <u>Material Database</u>      | Button for access to the material database, refer to Section 4.1.6.5.   |
| <b>Flags</b>                  |   |
| Penetration                   | If this flag is set, the layer supports penetration.  |
| Reflection                    | If this flag is set, the layer supports reflection.   |
| Diffraction                   | If this flag is set, the layer supports diffraction.  |
| Filled Obstacles              | If this flag is set, objects of this layer are treated as if they belong to closed, completely filled obstacles (e.g. persons, cars, etc.).   |
| Use as Vector DEM Layer       | If this flag is set, objects of this layer are treated as DEM layer defining the ground level. As a consequence, network elements (transmitters, receivers) or other environment elements can be placed in a height relative to this DEM layer.   |
| Use as Raster DEM Layer       | If this flag is set, the properties of this material layer are used for a raster DEM matrix which can be loaded separately. It is not allowed to define more than one layer as raster DEM layer. Network elements or other environment elements can be placed in a height relative to this DEM layer. |

A special "winter flag" is provided that allows to switch between two different values of the relative permittivity for the layer. This feature enables a quick change between two different propagation scenarios in one and the same environment, e.g. representing two different climate conditions.

The flags in the lower part of the settings dialog can be used to selectively activate or deactivate certain propagation phenomena for that layer. If objects of a layer are combined

to form solid, filled obstacles (can be used e.g. for modeling persons or cars, etc.), the Filled Obstacles flag must be set.

Make sure that in an environment no objects of other layers intersect with filled obstacles. Moreover, it must be assured that the bodies of filled obstacles are fully enclosed by polygons of the according layer, and no polygons inside the obstacles. Only if these conditions are met, the simulation engine can handle filled obstacles properly.

The last two flags of the material layer dialog are for defining material properties of a DEM layer. Refer to Section 4.1.6.4 for details on how to work with DEM layers.

In order to get a better overview of the settings of all material layers together, it is possible to view an overview dialog by choosing Settings Overview... from an arbitrary material layer's context menu in the tree window, refer to Fig. 4-7. The settings overview dialog is resizable and maximizable such that the whole screen size can be used to view the layer settings.

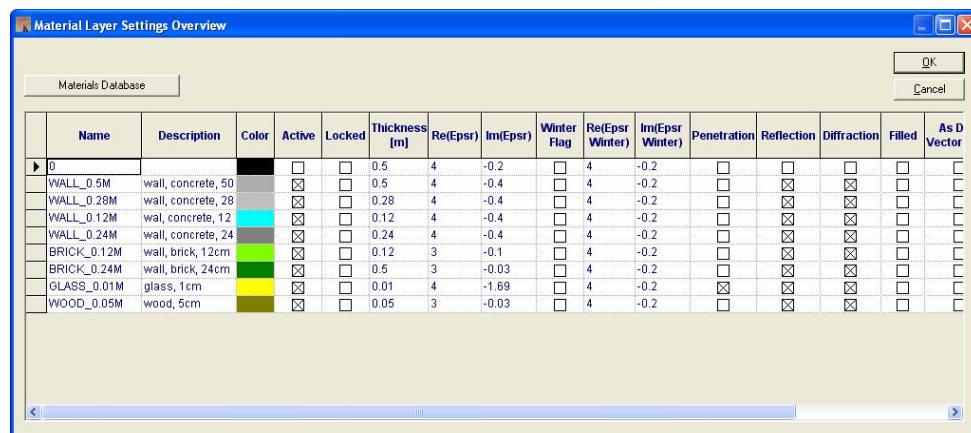


Fig. 4-7 Material layer settings overview

Besides the practical overview, this dialog offers the possibility to quickly change the material properties. In particular, the following actions can be performed in the table:

- Individual parameter values can be changed.
- Filters can be set individually for each column with the help of functions from the context menu. This enables a selective modification of a certain group of material layers in conjunction with the copy/fill function (see next bullet).
- Parameter values can easily be copied in a column (copy/fill function).
- The material database can directly be accessed from the upper left Material Database button. The selected material properties would be assigned to the selected row in the table.

See Section 4.2.2.3 for a short example on how to use the editing functions of the table.

#### 4.1.6.4 Defining a DEM Layer

A Digital Elevation Model (DEM) defines the terrain structure of an area. There are two possibilities to model such a DEM layer in RPS:

- **Vector DEM:** The DEM layer can be assembled of normal polygons. These polygons can be edited as any other environment objects in the Environment Editor, see Sections 4.1.7 to 4.1.10.
- **Raster DEM:** In many cases, DEM data are available as a raster matrix. Such a matrix can be directly imported and used as DEM layer in RPS.

The way to distinguish between these two options is by means of the DEM flags in the material layer settings dialogs.

Generally, any normal material layer containing some objects can be defined as a (vector) DEM layer by setting its vector DEM flag. The objects of this layer will be treated in the same way as before; in particular, they will have their specified material properties. In addition, some network elements (e.g. transmitters and receivers) and other environment objects can be assigned a height relative to the DEM layer. It is even possible to define more than one DEM layer – in this case, always the highest object is chosen for height reference if there would be overlapping DEM objects.

The advantage of a vector DEM layer is the high modeling accuracy which can be comparable as for the other 3D environment objects. In particular, all propagation phenomena can be fully considered during the ray tracing. A drawback could be the potentially large amount of data for larger areas.

Alternatively, a raster matrix can be loaded that defines the height level of the ground. This is accomplished by using the File -> Import -> Import Binary DEM Matrix... function from the menu. Once a raster DEM matrix has been imported, it inherits the material properties from the currently active layer where the raster DEM flag is set automatically. Only a single raster DEM layer is allowed in a project in order to avoid confusion about the assignment of the material properties to the raster DEM matrix. However, this assignment can be changed at any time by simply setting/unsetting the respective flag in the layers. A raster DEM matrix can be removed again by using the menu function File -> Import -> Remove Binary DEM Matrix from Project....

It is generally recommended to switch off penetration for DEM layers. According to their function, they mark the ground level which is usually not penetrable.

Diffraction can only be considered for vector DEM layers. However, in large areas it is not recommended to allow diffraction at the DEM layer due to the high computational complexity.

Sometimes it might be required to gain from the more precise modeling of a vector DEM matrix in an area where only raster DEM data are available. To this end, a function is available that allows to convert a raster DEM matrix into a vector DEM layer automatically. By using File -> Import -> Convert Binary DEM Matrix to Polygons..., an already existing raster DEM matrix is converted into a vector layer. When invoking this function, a dialog appears where the resolution of the vector layer (i.e. the size of the polygons) can be specified as shown in Fig. 4-8. Usually, the raster DEM matrix can be removed by setting the according flag in the dialog, if it is no longer needed.

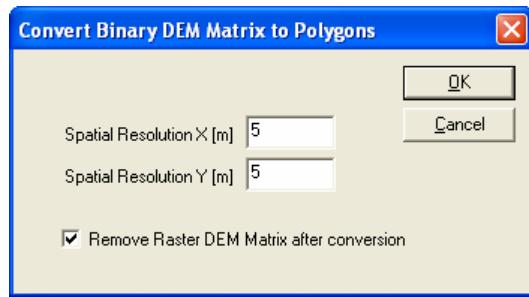


Fig. 4-8 Converting a raster DEM matrix into a vector layer

The newly created polygons are added to the currently active layer, irrespective whether this is a vector DEM layer or just a normal material layer. Make sure to set the DEM flag properly afterwards.

#### 4.1.6.5 Material Database

For a convenient material configuration, a material database with typical values of the relative permittivity is provided in RPS. When configuring a material layer in the appropriate settings dialog, the material database can be opened by hitting the Material Database button, see Fig. 4-9.

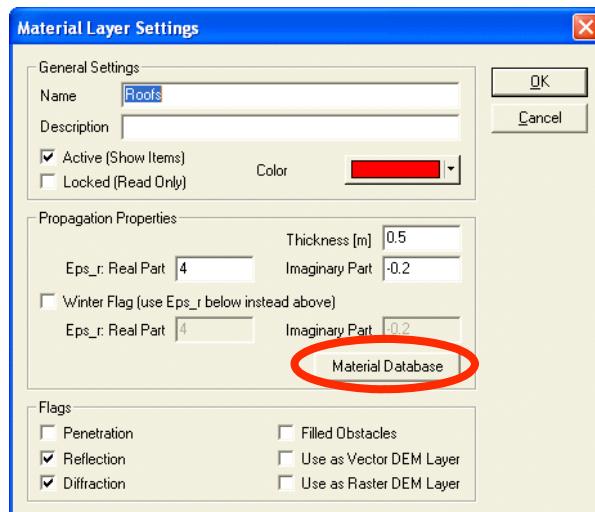


Fig. 4-9 Access to the material database

The dialog window of the material database is opened that is shown in Fig. 4-10. In the list field the desired material can be selected. Then the according value of the relative permittivity is assigned to the material layer by pressing the Insert button. The material database entries rely on the `MaterialList.ini` file in the RPS installation directory.

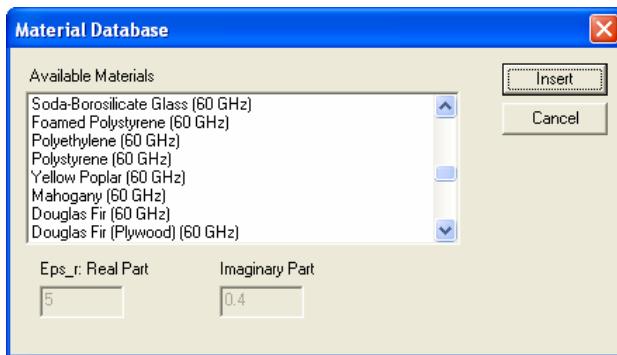


Fig. 4-10 Material database window

#### 4.1.7 Constructing Environment Objects

This Section is dedicated to the description of how to construct typical elements of an environment structure as for instance houses. Only very few steps are necessary to efficiently edit a basic propagation environment.

##### 4.1.7.1 Material Layer Selection

As the first step, a material layer must be chosen for a subsequently created environment object. A layer is selected by clicking on its item in the Materials tab of the tree view. If necessary, a new layer can be created. Its properties can be edited as explained in Section 4.1.6.3.

##### 4.1.7.2 Constructing Typical Building Components

Here, different ways of constructing a building shall be described. First, a more conventional version with starting wall-by-wall and then adding the roof shall be introduced. Later, an advanced way of constructing buildings in a map is shown.

For building a house wall-by-wall, please execute the following steps:

- Make sure that the WORLD coordinate system is activated or, alternatively, the top view is currently selected. Only in this global coordinate system an accurate positioning of objects is possible since all other coordinate systems are defined locally, independent of the WORLD coordinate system.
- Create a new layer and call it "Wall".
- Construct the first wall of the house by clicking the icon (tooltip Draw Surface (3DFace)). Specify the coordinates of the four vertices of the wall by typing the comma-delimited numbers in the command line of the Environment Editor. Generally, it is recommended to start defining an environment component from the keyboard. Later, the already existing objects can be used for an alignment of the object snap. The sample coordinates of the wall shall be:  $0,0,0 / 10,0,0 / 10,0,6 / 0,0,6$ .
- Construct a second wall in the same way as the first one. Its coordinates shall be  $0,0,0 / 0,6,0 / 0,6,6 / 0,0,6$ .

Now the view can be changed to a perspective view, e.g. the SW view. The current situation should be similar to the one depicted in Fig. 4-11.

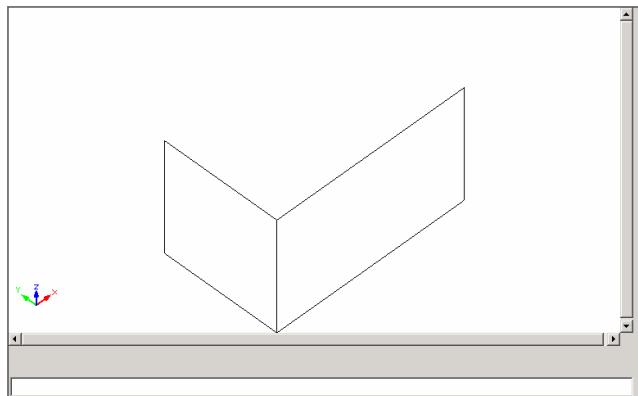


Fig. 4-11 Two walls of a building

The remaining two walls will be created by copying the existing ones. This will be accomplished by the following steps:

- Activate the object snap on end vertices (tooltip Object Snap End).
- Copy the first wall by clicking the icon (tooltip Duplicate Objects) in the lower toolbar. Select the first, longer wall by first focusing the mouse cursor on one of its vertices until it is highlighted by the object snap and then clicking on it. Since this is the only object to be copied for now, click the right mouse button to stop the selection of objects.
- Next, the moving vector must be specified. For the first position, click on the lower right corner of the second wall after it was highlighted by the object snap. Likewise, click on the lower left corner of the second wall to define the end point of the moving action. Then the new wall is inserted on the back side of the building.
- Copy the second wall in a similar way as just described to close the build from all four sides. The current situation should look as depicted in Fig. 4-12.

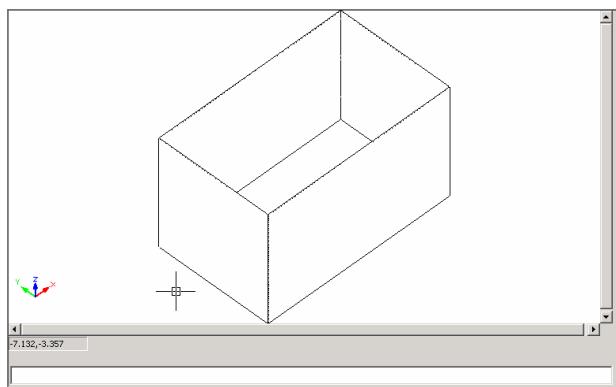


Fig. 4-12 Four walls of a building (hiding concealed objects)

For adding a flat roof on the building, simply a fifth rectangle is drawn. To do so, the following steps should be executed:

- Create a new layer and call it “Roof”. It is recommended to do this because roofs have usually different electrical properties than the walls.

- Construct the roof plate of the house by clicking the icon (tooltip Draw Surface (3DFace)). Specify the coordinates of the four vertices of the roof by clicking with the mouse cursor on the corresponding corners of the house using the object snap.
- Optionally, the colors of the wall and roof layers, respectively, can be modified. Then the house should look as shown in Fig. 4-13.

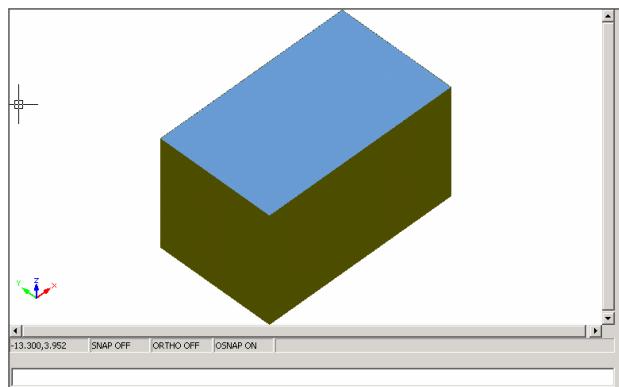


Fig. 4-13 A complete building

As an alternative, a house could be drawn on a map. To try out this version, one has to pursue through the following steps:

- Activate a snapping grid with a grid size of e.g. 2m. If needed, switch on the ortho mode, too.
- Make sure that the WORLD coordinate system is activated or, alternatively, the top view is currently selected.
- Insert a background image of the map to be used. Read in Section 4.1.4 on how to realize this.
- If necessary, create a new layer for the house and give it an appropriate name. Select that layer.
- Now the house ground can be drawn as a polyline. This is done with the command (tooltip Draw Polyline) from the lower toolbar. The distinct vertices of the polyline are set with the left mouse button. Take care that the last vertex is exactly equal to the first vertex. In order to safely accomplish this, the Object Snap End mode should be switched on. Then a situation as shown in Fig. 4-14 should be reached.

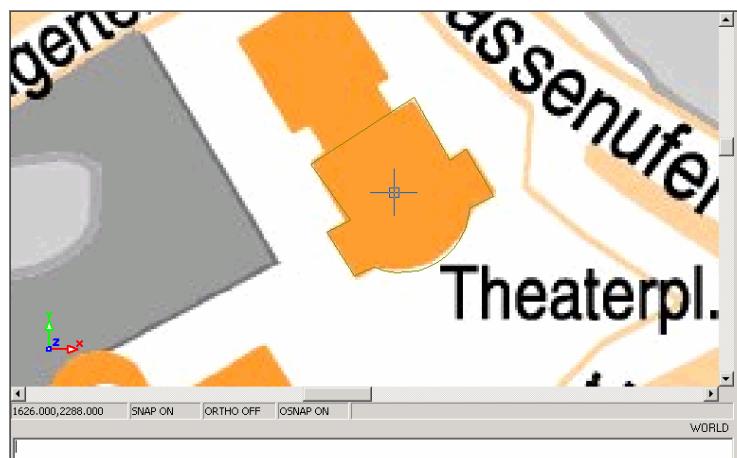


Fig. 4-14 Ground of a building

The next step would be to extrude the ground polygon to a solid body with a specified height. For this the Convert Polyline to 3D Object command (icon in the lower toolbar) is used that first asks the user to select an object. Then the height of the building must be given (called extrusion thickness here) which should be entered from the keyboard in the command line. Also a negative number is possible which would extrude the building downward. The following question should usually be answered with “n”. Then the building is finished with the specified height and closed by polygons on bottom and top. However, using the extrusion function it is not possible to assign different layers to roof and bottom. If this is desired, it must be done manually. The complete building on the map is depicted in Fig. 4-15.

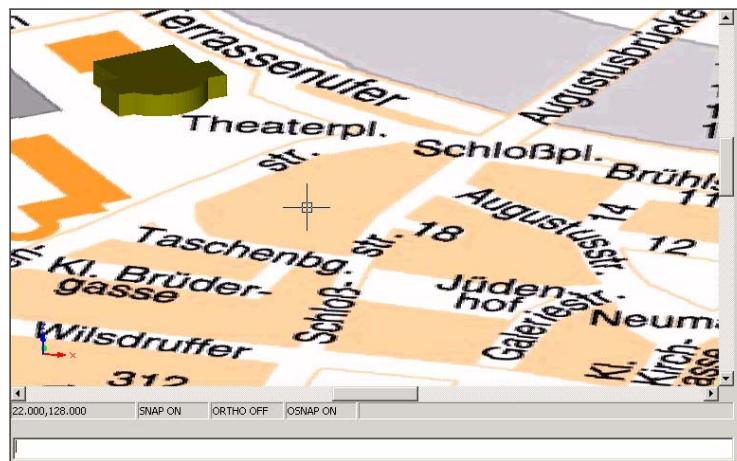


Fig. 4-15 Complete building on the map

Note that for the building just created, diffraction edges for a propagation simulation would also be detected at the ground line of the building as apparent from Fig. 4-16. If this is not desired, the block must be first exploded (i.e. decomposed into its individual polygons), and then the polygons of the bottom must be removed.

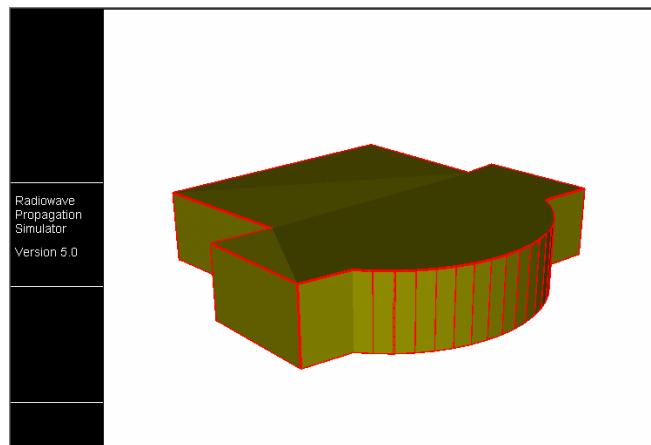


Fig. 4-16 Diffraction edges (red lines) of extruded building block

A building only consisting of walls can also be constructed in a similar way as just demonstrated. This is accomplished by using the icon (tooltip [Draw Wall along Polyline](#)) from the lower toolbar. In this case, the roof of the building could be added later using a different material layer. The usage of this function is analogous to the Draw Polyline method as just introduced. At the beginning, a height must be specified that is assigned all walls of the building to be constructed.

#### 4.1.7.3 Placing Vertices with Relative Positions

As apparent from the previous subsection, for an exact placement of environment objects it is recommended to type the coordinates in the command line instead of placing them out with the mouse pointer. In many cases it is then easier to specify a relative vertex position in relation to the position of the previous vertex. Such a relative position is given by prefixing the coordinates with a '@', e.g.

```
@0,0,10
```

which would denote a new vertex being placed exactly 10m above the previous one. This method can e.g. be used to define all vertices of a polygon after the first vertex or especially for moving objects.

#### 4.1.7.4 Constructing the Ground

If not an extremely simple environment shall be investigated, usually the definition of a ground level (i.e. a DEM layer) is indispensable. It is good practice to create a separate material layer for the ground for which the vector DEM flag is set. Furthermore, only reflection and possibly diffraction should be activated for this layer, whereas it is not recommended to allow penetration for the DEM layer.

The simplest way to construct the ground level is to define a (in most cases rectangular) ground plane. This can be accomplished by using the command [Draw Surface \(3DFace\)](#) the usage of which has already been introduced.

It is also possible to use a raster matrix as DEM layer. However, such a matrix is not visible in the Environment Editor because the Environment Editor can only handle vector objects. The import of a raster DEM matrix is explained in Section 4.1.8. If it is desired to have a vector DEM layer instead, it is even possible to convert an existing raster DEM matrix accordingly by using the menu entry [File -> Import -> Convert Binary DEM Matrix to Polygons....](#)

#### 4.1.7.5 Other Environment Objects

The lower toolbar of the Environment Editor still provides other, more sophisticated objects that can be used to construct a building environment. In particular, it is possible to insert meshes, boxes, cones, and spheres. The usage of these commands is self-explanatory – the user just needs to follow the instructions in the status bar of the edit window.



#### 4.1.8 Environment Import and Export

RPS supports the import and export of complex three-dimensional environments in the DXF- or DWG file formats. These functions are available from the entries Import -> Import DXF or DWG Environment Data... or Export -> Export Environment Data as DXF or DWG, respectively, in the File menu.

Alternatively, environment structures can also be imported and exported in RPS' own environment data format called OBL (object list). These functions are available from the entries Import -> Import Object List... or Export -> Export Environment Data as Object List, respectively, in the File menu. The OBL format is documented in Appendix 10.5.

Separate raster terrain data can be imported in a matrix form from a raster file by using the File -> Import -> Import Binary DEM Matrix... menu entry. Make sure that a special raster DEM material layer is created and selected before this function is invoked. Note that a raster DEM layer is not visible in the Environment Editor.

When saving an entire RPS project, the environment data are automatically stored in a separate DWG file. Thus environment data and network/result data are stored in different files which also enables a usage of the environment structure elsewhere.

#### 4.1.9 Working with Blocks

As soon as somewhat more complex environment components are created, it seems to be useful to group certain objects in a block. The handling of blocks in the Environment Editor is comparable to that of single objects. Blocks can be imported and exported. Moreover, an integrated stencil library provides a collection of often-used blocks like persons, cars, trees, etc. All block commands are available from the according icons in the lower toolbar.



Each block has a so-called insertion point. This is the position that all manipulating actions (like moving, copying, inserting) of the block are related to. The insertion point need not immediately be inside the block; however, it is good practice to select a prominent point in the block as the insertion point (e.g. the lower left corner or the balance point). When a block is created, the insertion point must be specified.

A block is always associated as a whole with a certain material layer. This is the currently selected layer at the time when the block is created or inserted. This layer is independent of the layers of the individual objects inside the block. The material properties of the objects are determined by the individual object layers, not by the block layer. This feature can be exploited for efficiently switching off or on complete blocks in the environment. If, for instance, all person blocks are assigned to a separate layer, the environment setup can easily be investigated with and without persons by simply toggling the active flag of that separate block layer.

Blocks as a whole are assigned to a certain material layer which can be chosen independently. The material properties of the individual objects inside a block are, however, determined by their respective layers.

#### 4.1.9.1 *Creating a Block*

An arbitrary collection of objects can be combined in a block. These objects can belong to different material layers – they will retain their properties. Before the block is created, a proper material layer must be selected that the new block shall be assigned to. For grouping objects the Make Block command ( icon) is used. First, a name of the block must be chosen. This can be an arbitrary string. Then the insertion point is to be specified. In order to select one of the vertices of the objects to be grouped, the object snap functions can be used. Last, all objects of the block must be selected. The command is finished by pressing the right mouse button.

#### 4.1.9.2 *Exploding a Block*

The opposite action, i.e. decomposing a block into its individual objects, is done with the help of the Explode Block command ( icon in the lower toolbar). After choosing this command, the block to be decomposed is to be selected which will explode it. All objects contained in the block will retain their original material layers.

Some more complex shapes like cone, sphere, or extruded polylines, are internally handled as polyfaces, i.e. they represent a single object with many vertices and interconnecting lines. Such complex objects can also be decomposed into primitive polygons (mostly triangles) by applying the Explode Block command. Hence it is possible to erase the bottom plane of an extruded polyline as it was described in the second example of Section 4.1.7.2.

#### 4.1.9.3 *Stencil Library*

RPS has an integrated stencil library that provides a collection of predefined blocks (stencils) for often-used obstacles like persons, cars, etc. Make sure that before a block from the stencil library is added, a proper material layer must be selected that the inserted block shall be assigned to. The stencil library can be accessed by pressing the  icon (tooltip Block Library...) in the lower toolbar. The window of the stencil library is opened where all components of the library are listed, grouped into different categories. The stencil library window is shown in Fig. 4-17. The controls of this window are described in Table 4-3. The blocks in the stencil library are documented in Appendix 10.3.

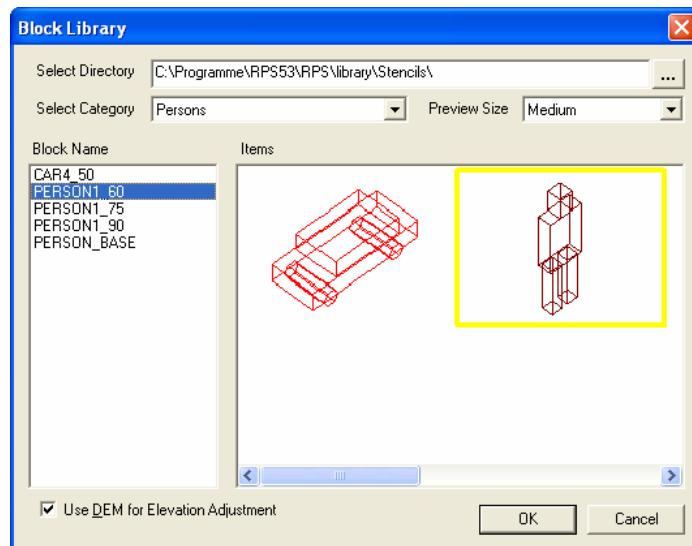


Fig. 4-17 Stencil library window

Table 4-3 Stencil library handling

| Element                          | Description   |
|----------------------------------|---|
| Select Directory                 | The stencil library path. The user could choose an alternative stencil library by using the <input type="button" value="..."/> button.  |
| Select Category                  | A list field with all available categories of the stencil library.  |
| Preview Size                     | A list field where the preview size (small, medium, or large) for the blocks in the Items window can be chosen.   |
| Block Name                       | A list box that shows all available blocks in the current category. The selected block is also highlighted in the Items window to the right.  |
| Items                            | In this window, small previews of all blocks in the current stencil category are shown. The selected stencil has a yellow border. Stencils can also be directly selected in the Items window by a left mouse click. |
| Use DEM for Elevation Adjustment | If this flag is set, the insertion point of the block will be aligned relative to the height of the DEM layer.  |

After the desired block has been selected, it can be inserted into the environment by clicking the OK button. The location of the insertion point and a horizontal rotation angle need to be specified yet. If the inserted block contained some new material layers that did not exist yet in the environment then these layers are created automatically.

The specific electrical properties of newly created material layers on block insertion have to be edited manually since they cannot be saved in a block.

Internally, the stencil library is a hierarchical directory structure where each block is stored in a separate DWG file. The main directory is located under %ApplicationPath%\library\Stencils\. All available categories are subdirectories in this path. This opens the opportunity to individually extend the stencil library by user-defined blocks and even to create new stencil categories by simply adding new subdirectories in the library.

#### 4.1.9.4 Block Import/Export

A block can be exported into a DWG file by using the icon (tooltip Write Block as new DWG file) in the lower toolbar. A dialog appears, see Fig. 4-18, where either an existing block can be chosen or a new block can be defined. Then the file name for the DWG file can be specified in the file save dialog. If the option Select Entities... was chosen before, the objects of the new block must be selected and its insertion point must be defined.

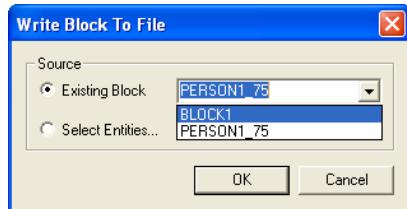


Fig. 4-18 Block selection dialog for block export

Blocks can also be inserted at an arbitrary position. Before the insertion is performed, a proper material layer for the new block should be selected. There are two alternatives: Either a saved block is imported or an existing block in the current environment is copied. The corresponding command is accessible from the icon (tooltip Insert Block) in the lower toolbar. The block insertion dialog (Fig. 4-19) is shown where the different options can be chosen, see Table 4-4.

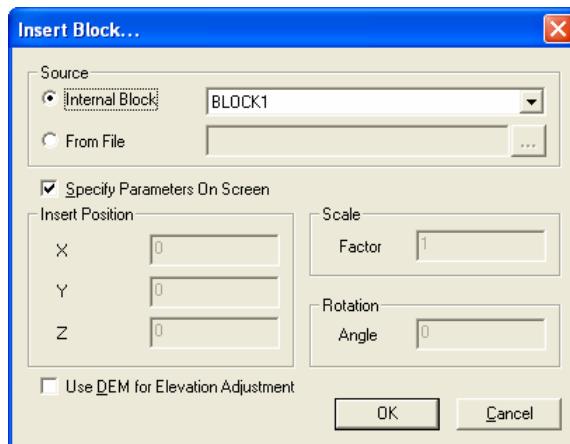


Fig. 4-19 Block insertion dialog

Table 4-4 Block insertion options

| Option         | Description  | Unit |
|----------------|--|------|
| Internal Block | If this radio button is checked, a block from the according list field can be chosen. All existing blocks in the environment are listed there.   | —    |
| From File      | If this radio button is checked, a block from a file can be loaded. The file name can be specified in the according text field. Alternatively, a file open dialog can be used which is accessible from the  icon to the right. | —    |

| Option                           | Description   | Unit |
|----------------------------------|---|------|
| Specify Parameters on Screen     | If this flag is set, the positioning of the block can be done by mouse commands or by specifying the parameters in the command line from the keyboard. Otherwise, the following parameters determine the positioning of the inserted block. | —    |
| Insert Position                  | The position of the insertion point of the block in Cartesian coordinates.  | m    |
| Scale Factor                     | The scale for the inserted block. A factor of 1 inserts the block in its original size.   | —    |
| Rotation Angle                   | The horizontal rotation angle of the block relative to its original orientation. This value must be in the range [0, 360).  | °    |
| Use DEM for Elevation Adjustment | This flag determines whether the height of the block is positioned relative to the DEM layer.   | —    |

If a block from a file is inserted, all objects contained in the selected file are combined to a new block irrespective of their possible internal block grouping. The new block is assigned to the currently selected material layer; however, the material properties of the distinct objects inside the block are determined by their original layers. If the file contains other layers than those already existing in the environment, then these other layers are newly created.

**Important Notice for Block Import:** The specific electrical properties of newly created material layers on block insertion have to be edited manually since they cannot be saved in a DXF/DWG file.

#### 4.1.10 Modifying Environment Objects

After a coarse building structure of an outdoor environment has been imported (Section 4.1.8) or created (Section 4.1.7), it can be refined by adding some more obstacles as, e.g., persons or cars to model the propagation environment as realistically as possible. Also, in an indoor environment, the accuracy of the results can be improved by adding smaller obstacles.

The additional obstacles are provided by the stencil library; this Section explains the commands to quickly and effectively insert these obstacles into the environment. Most of them can be accessed from the according icons in the lower toolbar of the Environment Editor. They provide functions as to move, copy, scale, wobble, rotate, and delete objects, as well as to modify single object vertices.



##### 4.1.10.1 Inserting Additional Obstacles

In this Subsection, altogether three examples are presented that train the user step by step how to efficiently exploit the Environment Editor for the insertion of additional obstacles:

- creating a regular Manhattan grid of block buildings with different heights
- irregularly located persons in a street of the Manhattan grid

- covering an area with forest

The key to all of these examples is a complex command in the Environment Editor: [Create Array of Objects](#) ( icon from the lower toolbar). When it is invoked, a dialog is opened that allows to multiply blocks (not objects!) and arrange them in different ways, see Fig. 4-20. If an object shall be multiplied, it must first be converted to a block. The parameterization of this complex command is described in Table 4-5.

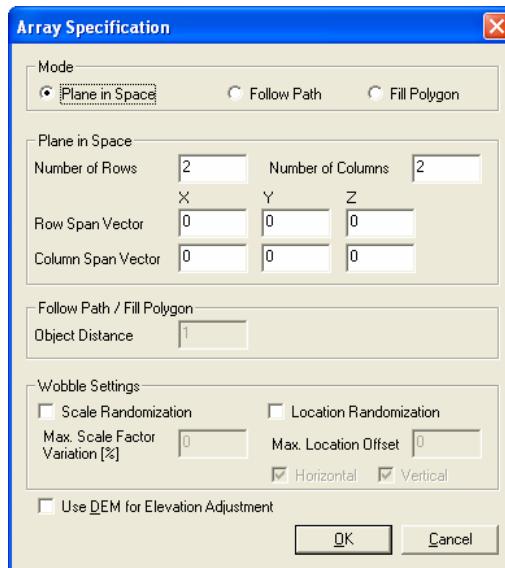


Fig. 4-20 Object array specification dialog

Table 4-5 Object array parameters

| Parameter             | Description  | Unit |
|-----------------------|--|------|
| <b>Mode</b>           |  |      |
| Plane in Space        | If this radio button is selected, the command will arrange the selected objects in a grid with a given number of rows and columns. Furthermore, span vectors for the rows' and columns' direction. Thus it is possible to create skew matrices, too. The span vectors are started at the selected objects' position. | —    |
| Follow Path           | If this radio button is selected, the command will arrange the selected objects along a polyline. The polyline is started at the selected objects' position.   | —    |
| Fill Polygon          | If this radio button is selected, the command will arrange the selected objects inside a polygon. The polygon can be arbitrarily defined. In this case, the originally selected objects are removed, and only the polygon is filled with them according to the specified spacing.                                    | —    |
| <b>Plane in Space</b> |  |      |
| Number of Rows        | The number of rows for the object grid.<br><b>Used for option Plane in Space only.</b>   | —    |
| Number of Columns     | The number of columns for the object grid.<br><b>Used for option Plane in Space only.</b>  | —    |

| Parameter                         | Description  | Unit |
|-----------------------------------|--|------|
| Row Span Vector                   | The span vector for the rows of the object grid. If a regular matrix shall be created, it is recommended to define this vector in x-direction.<br><b>Used for option Plane in Space only.</b>              | m    |
| Column Span Vector                | The span vector for the columns of the object grid. If a regular matrix shall be created, it is recommended to define this vector in y-direction.<br><b>Used for option Plane in Space only.</b>           | m    |
| <b>Follow Path / Fill Polygon</b> |  |      |
| Object Distance                   | The mean distance between objects on a polyline or inside a polygon.<br><b>Used for options Follow Path or Fill Polygon only.</b>  | m    |
| <b>Wobble Settings</b>            |  |      |
| Scale Randomization               | If this option is checked, the dimensions of the selected objects will be scaled with a scale factor uniformly distributed. The maximal deviation of the scale factor is given by the following parameter. | —    |
| Max. Scale Factor Variation       | The maximal variation of the scale factor.   | %    |
| Location Randomization            | If this option is checked, the locations of the selected objects will randomized on placing. Thus it is possible to realize a slightly irregular arrangement for, e.g. persons or trees.                   | —    |
| Max. Location Offset              | The maximal location offset of the selected objects from their regular position.   | m    |
| Horizontal                        | This flag specifies that the displacement is only applied to the horizontal plane, i.e. in x- and y-direction.   | —    |
| Vertical                          | This flag specifies that the displacement is only applied to the vertical axis, i.e. in z-direction.   | —    |
| Use DEM for Elevation Adjustment  | This flag determines whether the height of the objects is positioned relative to the DEM layer.  | —    |

This complex command allows to place objects in different ways: in a regular matrix with possibly skew span vectors, along a polyline with a given spacing, and inside a polygon with a given spacing. Since in a real environment, obstacles usually are not completely regularly placed, a certain “wobbling” can be specified. This wobbling can affect the size of the objects (option Scale Randomization) and the location of the objects (option Location Randomization).

In the following, the examples shall be presented in order to provide a better understanding how to effectively use this complex command. In the first example, a regular Manhattan grid with houses of different height shall be created. The following steps are necessary:

- Create a new material layer for the ground plane and call it “Ground”. Edit its properties accordingly (e.g. the DEM flag should be activated) and select that layer in the Materials tab of the tree window.
- Create a ground plane by using the Draw Surface (3DFace) command. The vertices of the plate should be (0, 0), (0, 3000), (3000, 0), and (3000, 0).

Now a block building shall be constructed as a box that functions as a prototype for all houses to be inserted later on. For the first house, the following steps have to be executed:

- Create a new material layer for the buildings and call it “Building”. Select that layer.
- Draw a box at (0, 0) with a length and width of 50m and a height of 300m. The rotation angle can be set to 0°.

In order to generate houses of different heights, it is planned to partially hide the houses in the ground without changing the actual height of the box just drawn. For that purpose, the first house needs to be moved to a certain start position for the Manhattan grid and dump it half into the ground. Furthermore, the original box (which is a normal object) still needs to be converted into a block because the array command can only work on blocks.

- The original block building should be moved to its start position for the Manhattan grid by using the Move Objects command ( icon in the lower toolbar). If you, as an alternative, wish to retain the original box, you can also apply the Duplicate Objects command ( icon in the lower toolbar). The parameters of both commands are the same.
- Move the building to the position (1000, 1000, -140).
- Create a new material layer for the block of the buildings. Call it “Block buildings” and select it.
- Convert this box into a block by the Make Block command. Call the block “building”, for example. The insertion point should be (1025, 1025).

The block of the first building will now be used as a template to create the regular Manhattan grid. This multiplication of the block is accomplished with the help of the Create Array of Objects command. In the appearing specification dialog the following adjustments should be made:

- Select the option Plane in Space.
- Both the number of rows and columns shall be 12.
- The span vectors for rows and columns shall be (80, 0, 0) and (0, 80, 0). Since the houses are 50m wide, the resulting street width will be 30m.
- Select the option Location Randomization. Adjust the max. location offset to 140m and activate vertical randomization only (not horizontal). The location offset should be chosen with care such that it is assured that the buildings cannot “float” above ground.
- Select the DEM flag if appropriate.

After all settings have been confirmed by pressing the OK key, the user is queried to select the entities to be arranged. Only the single block building needs to be selected, then the right mouse button can be pressed. Immediately, the buildings are multiplied and arranged in the predicted Manhattan grid. The result of this action can be seen in Fig. 4-21.

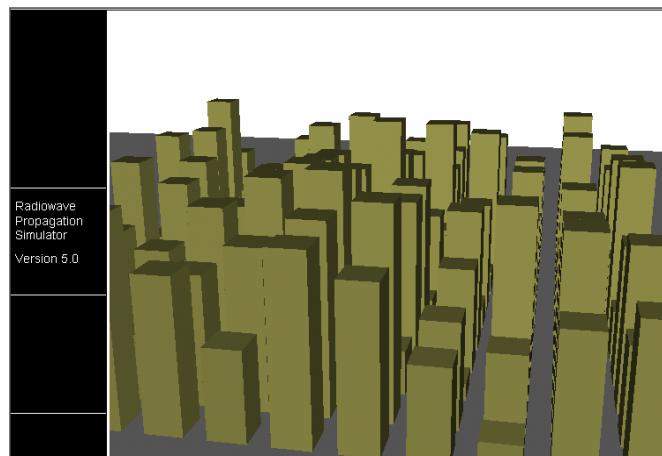


Fig. 4-21 Regular Manhattan grid with buildings of different heights

If you inspect this environment carefully in the 3D view, you will see that the buildings reach partially below the ground level. The reason for this is that all boxes have the same total height; only the heights above ground are different.

In the next example, people shall be placed along an East-West street in the Manhattan grid just created. The start configuration of the environment is shown in Fig. 4-22.

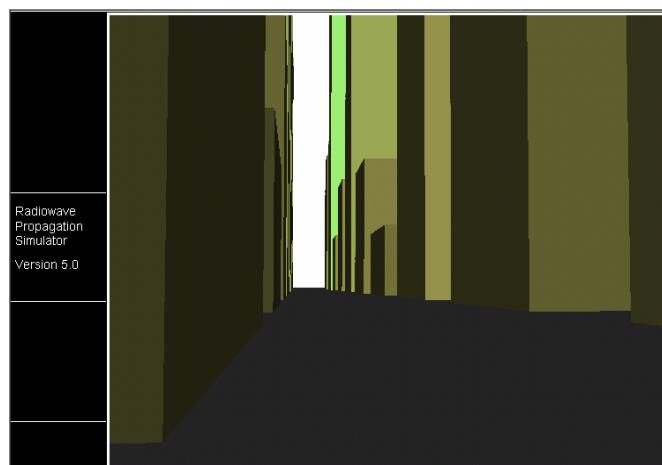


Fig. 4-22 Street in a regular Manhattan grid

To add the people, the following steps are performed:

- Create a new material layer for the block of the persons and call it “People”. Select that layer.
- Insert a person from the stencil library (e.g. “PERSON1\_75”) and place it at the Western end of the street. Optionally, the person can be rotated by 90° when placing it. Read more on using the stencil library in Section 4.1.9.3.

Since the person block in the stencil library is oriented for North-South streets, it still needs to be rotated properly if this was not yet done during the import action. A separate rotation can be executed by means of the Rotate Objects command ( icon in the lower toolbar). First a rotation axis must be specified. In our case, the person shall be rotated around the z-axis, i.e. in the horizontal plane. Then the person must be selected and rotated by 90°.

Note that the person imported from the stencil library has an own material layer "PERSON" which was automatically created on the import action. However, the electrical properties of that layer need still to be edited since they could not be stored in the library. In particular, the Filled Obstacles flag should be set.

The first person that has been imported from the stencil library will now be used as a template to create many people along a straight line. This multiplication of the person block is also accomplished with the help of the Create Array of Objects command. In the appearing specification dialog the following adjustments should be made:

- Select the option Follow Path. Adjust the distance between the people to 2m.
- Select the option Scale Randomization and adjust the max. scale factor variation to 10%.
- Select the option Location Randomization. Adjust the max. location offset to 0.5m and activate horizontal randomization only (not vertical). The location offset should be chosen with care such that it is assured that the blocks to be placed cannot intersect with each other.
- Select the DEM flag if appropriate.

After all settings have been confirmed by pressing the OK key, the user is queried to select the entities to be arranged. Only the single imported person needs to be selected, then the right mouse button can be pressed. Next, the path along which the persons shall be arranged is to be drawn. Note that the starting position of that polyline is the insertion point of the person. It is recommended to specify the second position in the command line by typing e.g.

@1000 , 0

which will set the next point of the path 1000m away from the first person along the x-axis, i.e. along the East-West street. The z-coordinate can be omitted which will then be defaulted to 0. Since this shall be the last position (only a straight line as path is used), now the right mouse button can be clicked in the edit window to conclude the command. Immediately, all the people are inserted with the specified wobbling along the street. The result of this action could look as in Fig. 4-23.

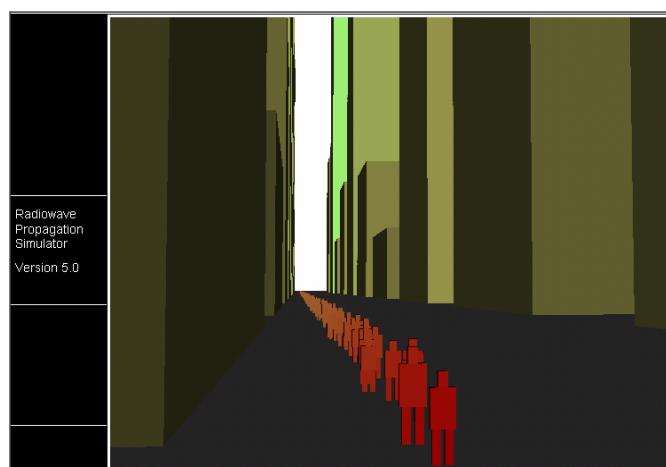


Fig. 4-23 People walking along a street

In order to prepare the third example, some of the blocks should be removed by means of the Erase Objects command to give space for a park. After this is done, the starting situation for this last example could look as in Fig. 4-24.

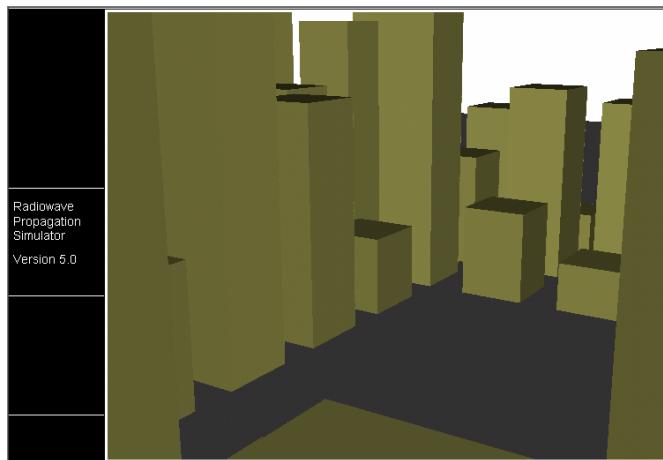


Fig. 4-24 Space for a park

Next, a tree should be imported from the stencil library. Before this is actually done, another material layer “Trees” should be created for the tree blocks. Then, choose “TREE4\_50” from the stencil library and place it somewhere outside the building area. Make sure that for the following actions the previously created “Trees” layer is selected.

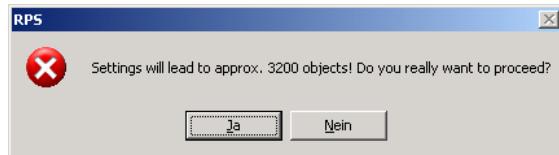
Since we would like to use a bigger tree, the imported one should still be scaled before the park trees are created from it. Scaling is done by means of the Scale Objects command ( icon in the lower toolbar). After this command was invoked, perform the following steps:

- Select the tree. Then click the right mouse button to stop the object selection.
- Specify the scale base point. This should be the same position where you just placed the tree because then the scaling will be related to the tree’s insertion point.
- Enter ‘2’ as scale factor. Alternatively the scaling could be specified with the mouse.

Immediately, the tree size is doubled in all dimensions. Now this tree can be used as a template to create a small forest in the park area. This multiplication of the tree block is also accomplished with the help of the Create Array of Objects command. In the appearing specification dialog the following adjustments should be made:

- Select the option Fill Polygon. Adjust the distance between the trees to 5m.
- Select the option Scale Randomization and adjust the max. scale factor variation to 30%.
- Select the option Location Randomization. Adjust the max. location offset to 1.5m and activate horizontal randomization only (not vertical). The location offset should be chosen with care such that it is assured that the blocks to be placed cannot intersect with each other.
- Select the DEM flag if appropriate.

After all settings have been confirmed by pressing the OK key, the user is queried to select the entities to be arranged. Only the single imported tree needs to be selected, then the right mouse button can be pressed. Next, a polyline that surrounds the forest area in the park is to be drawn. Due to the wobbling it is accurate enough to specify the vertices of the polyline with the mouse. The area is automatically closed when the right mouse button is pressed. A warning message like the one shown below could appear if a very large number of objects will be inserted.



Note that the original tree has disappeared after the insertion action. This is only the case for the Fill Polygon option of the object array command. Also in this example, the electrical properties (in particular the Filled Obstacles flag) should be manually edited for the tree layer. The result of the last example is shown in Fig. 4-25.

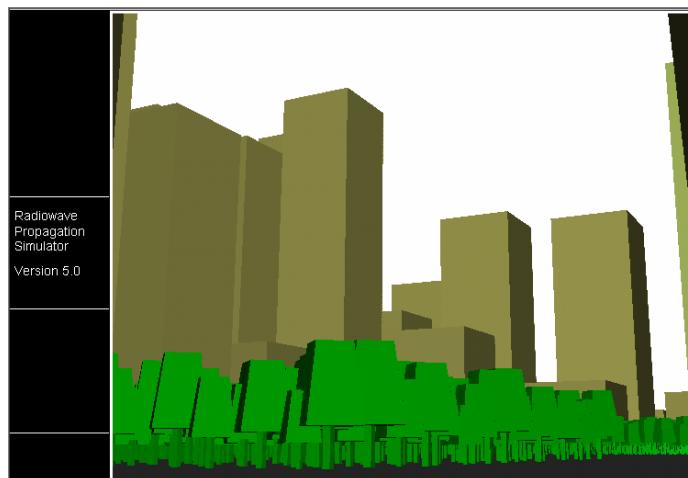


Fig. 4-25 “Central Park”

The procedure presented in these examples is just a demonstration of the vast range of possibilities that this mighty command offers. Likewise, it could be used to quickly and efficiently create other scenarios. Fig. 4-26 presents another example where cars were placed along a road that is flanked by avenue trees.

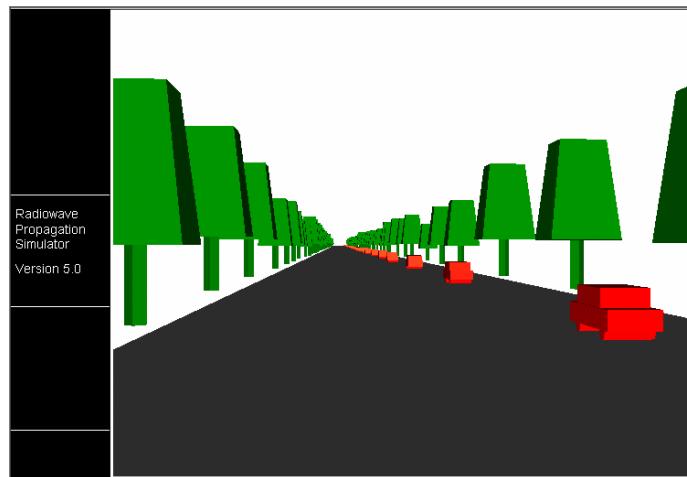


Fig. 4-26 Cars in the road

#### 4.1.10.2 Moving Object Vertices

Distinct vertices of objects can be modified separately. This special mode is entered by selecting an object to be changed. The vertices of that object are highlighted then by small blue squares.

A vertex of the selected object can be moved by clicking on that vertex with the left mouse button. The vertex can then be dragged to a new position. Alternatively, the new position can be entered from the keyboard in the command line. For a relative movement from the current vertex position, it is especially useful to prepend the shifting vector by a “@”.

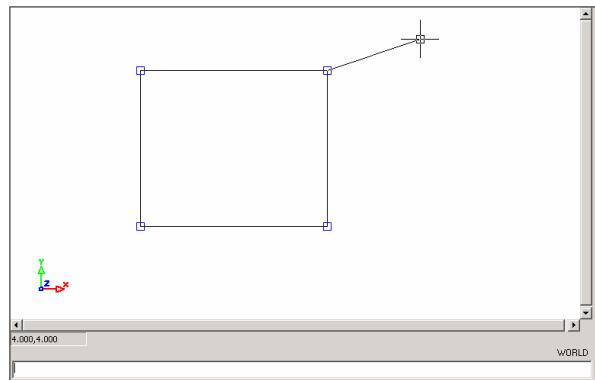


Fig. 4-27 Moving the upper right corner of a rectangle

The object selection mode can be left again by pressing the <ESC> key or by starting another command.

#### 4.1.10.3 Changing Object Properties

The assigned material layer of an object can be modified in the object properties dialog. It is accessible from the icon (tooltip Modify Properties) in the lower toolbar. The layer can be changed in the lower list field that contains all available layers, see Fig. 4-28. Moreover, an optional label can be specified for the object. This label would be shown as a tooltip as soon as the mouse cursor is located above the object in the edit window of the Environment Editor. When a block is inserted, its name is automatically assigned to it as label.

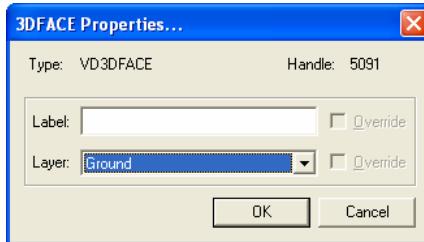


Fig. 4-28 Object properties dialog

#### 4.1.10.4 Removing Objects

Objects can be removed by clicking the icon (tooltip Erase Objects) from the lower toolbar. After this command is invoked, the objects to be deleted must be selected. Then a click with the right mouse button finishes this action and removes the objects.

Sometimes it is difficult to exactly locate and select those objects that shall be removed. In such cases the Locked flag of the material layers could help, especially if the objects to be erased all belong to the same layer. If the Locked flag of all the other layers (those that shall *not* be removed) is set, the window select option can be used to select the objects while drawing the dragbox around all objects. Then only the objects of the layer(s) that shall be removed can be selected – all others are locked.

#### 4.1.10.5 Cleaning up the Environment Database

During the creation and modification of a complex environment it can easily happen that intermediate blocks or layers have been created that are not pertained in the final environment. In order to clean up the database and thus to remove any unused layers or blocks, the Purge Unreferenced command ( icon in the lower toolbar) must be invoked.

### 4.1.11 Supplementary Functions

The introduction to the Environment Editor is concluded by describing some supplementary functions that enhance its usefulness considerably. Despite some general preferences and a distance measurement function, especially the possibility to automatically add flat roofs and to detect diffraction edges in complex environments are very convenient functions.

#### 4.1.11.1 Environment Editor Preferences

The general preferences of the Environment Editor are accessed by clicking the icon (tooltip Environment Editor Preferences) in the upper toolbar. A dialog appears that is shown in Fig. 4-29. The distinct parameters are explained in Table 4-6.

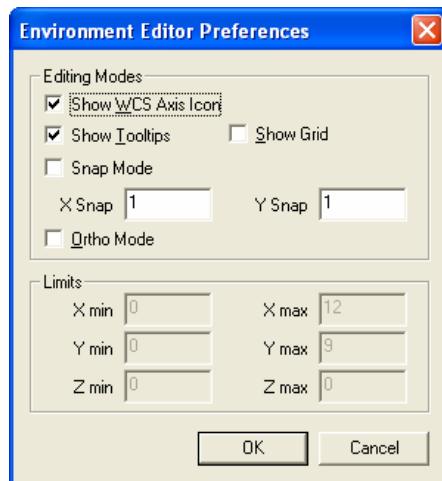


Fig. 4-29 Environment Editor preferences dialog

Table 4-6 Environment Editor preferences

| Parameter          | Description   | Unit |
|--------------------|---|------|
| Show WCS Axis Icon | If this flag is set, the coordinate system axis icon is shown in the lower left corner of the edit window.  | —    |
| Show Tooltips      | If this flag is set, the optional object labels are shown in tooltips.  | —    |
| Show Grid          | If this flag is set, a snapping grid is shown with the specified resolution.  | —    |
| Snap Mode          | This flag activates the snap mode. It can also be toggled by double-clicking the area in the status bar below the edit window where 'SNAP ON/OFF' is shown.   | —    |
| X/Y Snap           | The grid size of the snapping grid in the x- and y-directions.  | m    |
| Ortho Mode         | This flag activates the ortho mode. It can also be toggled by double-clicking the area in the status bar below the edit window where 'ORTHO ON/OFF' is shown. | —    |
| Limits             | The limits of the environment. These coordinates are just shown for information; they cannot be modified. The <u>Zoom All</u> command uses these values.      | m    |

#### 4.1.11.2 Distance Measurement

The distance between two positions can be measured with the Measure Distance command ( icon in the lower toolbar). After clicking the according icon, the two positions must be selected. It is very helpful to use the object snap to safely identify a desired position in 3D space. Both positions and their distance are reported in the message window as in the following example:

```
Point 1: x: -5.000, y: -2.000, z: 0.000
Point 2: x: 1.000, y: 3.000, z: 0.000
Distance: 7.810, dx: 6.000, dy: 5.000, dz: 0.000
```

#### 4.1.11.3 Refreshing the View

After complicated actions in the Environment Editor it can happen that the view is slightly distorted. Then it could help to try either of the commands Redraw ( icon) or Regenerate ( icon) from the upper toolbar. Especially in conjunction with shaded or hidden views, it can be additionally beneficial for a clear view to deactivate and then reactive layers.

#### 4.1.11.4 Adding Flat Roofs

In Section 4.1.7.2, two possibilities to construct houses with roofs were discussed. As another option, RPS provides the function to globally add flat roofs to *all* closed buildings in an environment. This function can be invoked by choosing the menu entry Preferences -> Detect Missing Roofs.

This function should generally be used to convert outdoor environments from older RPS projects prior to version 5. RPS does not support roofless houses any more with respect to the over-rooftop diffraction since no diffraction edges at the roofs could be detected.

The implemented algorithm to detect flat roofs on closed house skeletons (consisting of walls only) has the following limitations:

- It is not detected whether a certain house already has a roof. Thus, this function should only be applied *one time* for an environment.
- The roof that is added on the entire house that is surrounded by its walls. More advanced constructions like court yards without a roof are not supported.

#### 4.1.11.5 Detecting Diffraction Edges

RPS allows the automatic detection of diffraction edges using the menu command Preferences -> Detect Diffraction Edges. The process could be interrupted by pressing the <ESC> key. At the beginning of the identification process all existing diffraction edges are removed.

All diffraction edges can be manually removed at once using the menu command Preferences -> Remove Diffraction Edges.

Note that a manual detection of diffraction edges is not necessary. At each simulation start, RPS automatically detects the diffraction edges in an environment if the Preferences -> Auto Detect Diffraction Edges before each Simulation Run flag is checked.

### 4.1.12 Printing from the Environment Editor

It is possible to print the current view of the Environment Editor. The print dialog can be accessed by choosing File -> Print... when the Environment Editor is currently active. This dialog is shown in Fig. 4-30.

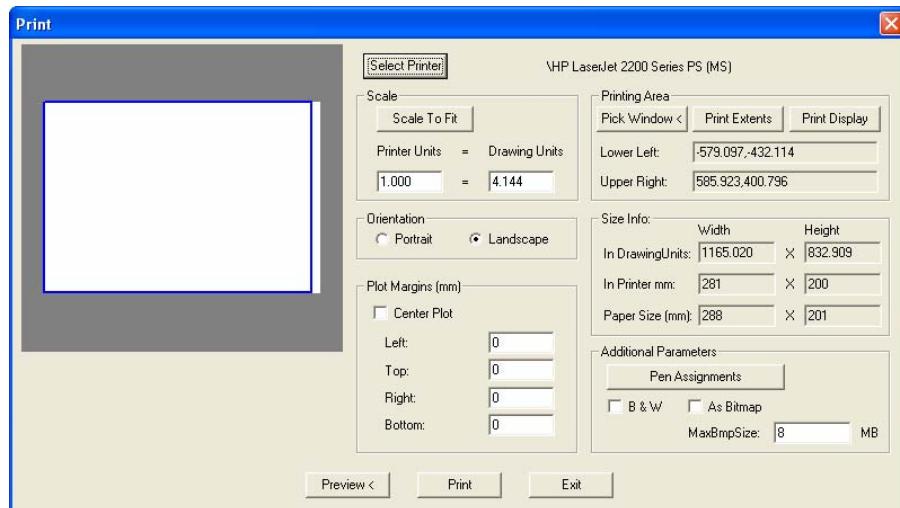


Fig. 4-30 Print dialog of the Environment Editor

In order to get good printing results, it is recommended to press the Scale to Fit button to place the entire environment in the printable area. The Preview button can be used to show the print preview window. This window can also be directly accessed by clicking the icon (tooltip Print Preview) from the standard toolbar or, alternatively, by using the menu entry File -> Print Preview....

The actual printing process is initiated by clicking the Print button in the print dialog or, alternatively, by directly clicking the icon (tooltip Print) from the standard toolbar.

## 4.2 Network Configuration

A network configuration in RPS is made up of the arrangement of transmitters and receivers together with their associated antennas. In the following subsections these entities and their setup are introduced in more detail.

### 4.2.1 Antenna



An antenna comprises both a vertical and a horizontal diagram that are used to interpolate a three-dimensional antenna pattern. Each distinct antenna represents a template that can be referenced arbitrary times from the transmitters and receivers.

The antenna diagram is specified by a horizontal ( $\varphi$ -plane), and a vertical ( $\vartheta$ -plane) antenna diagram. Additionally, a gain value and a polarization can be specified for the antenna. The horizontal pattern must be normalized such that the attenuation has its minimum at  $\varphi = 0^\circ$  with  $C_\varphi(\varphi = 0^\circ) = 1$ . For the vertical pattern it is only demanded that  $\min_{\vartheta} C_\vartheta(\vartheta) = 1$ ;  $-90^\circ < \vartheta \leq 90^\circ$ . Thus it is possible to use antenna patterns with an incorporated downtilt, i.e. the minimal attenuation is at  $\vartheta \neq 0^\circ$ .

The three-dimensional antenna pattern is interpolated from the given horizontal and vertical diagrams. An example pattern comprising a horizontal and a vertical diagram is shown in Fig. 4-31, its three-dimensional presentation from two different perspectives is given in Fig. 4-32.

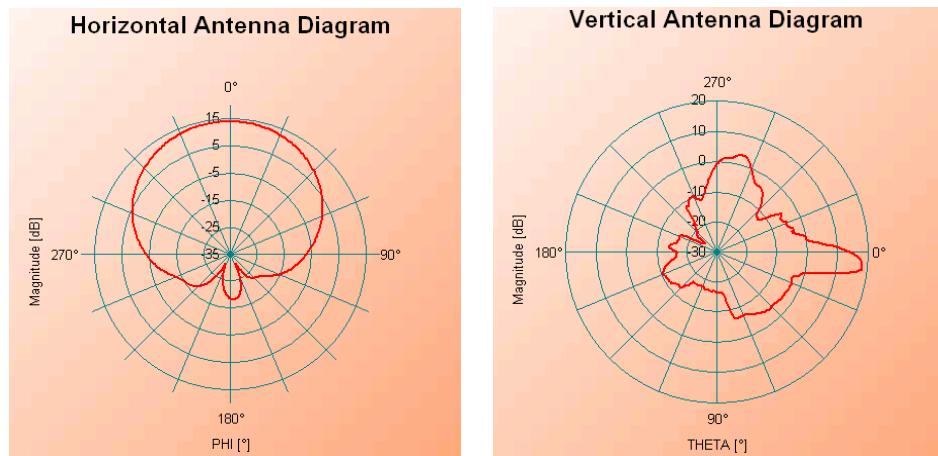


Fig. 4-31 Example antenna pattern described by a horizontal and a vertical cut

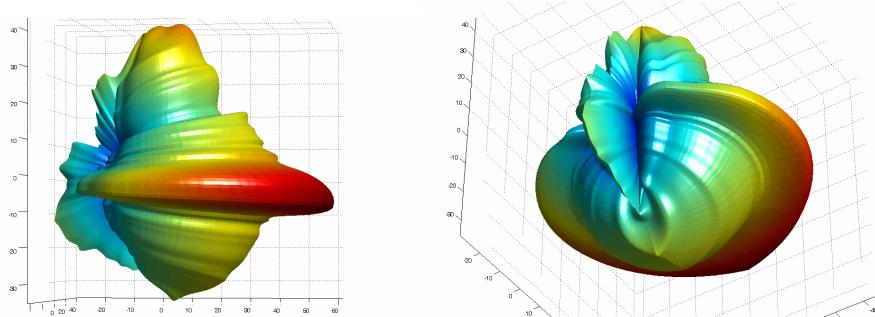


Fig. 4-32 Three-dimensional presentation of the antenna pattern from Fig. 4-31

A description of the antenna pattern interpolation rule as well as the antenna file format can be found in Appendix 10.6.

Due to a change in the antenna data format and the new incorporated antenna library, the antennas of older RPS versions (v4.1 and older) cannot be imported any more. Thus if old RPS projects are used, the antennas have to be assigned manually to the transmitters and receivers again.

#### 4.2.1.1 Importing Antennas

By default, only an isotropic antenna is available for the network configuration. In order to use different and more complicated antennas, it is possible to import antennas from the incorporated antenna library.

Antennas reside in a separate section of the Configuration tab of the tree window called **Antennas**. To import one or more antennas, select **Import Antennas...** from the context menu of the **Antennas** node. A file open dialog is shown where the desired antenna file(s) can be chosen. By default the available library items are presented. More than one antenna can be selected by using the **<Shift>** and **<Ctrl>** keys. Then the new antennas are imported and added to the tree together with possibly other antennas.

Note that the import of an antenna not immediately means that it is used in the network configuration. It must be referenced by the transmitters and receivers in order to be used.

#### 4.2.1.2 Assigning an Antenna to the Receivers

In RPS, all receivers use the same antenna with the same orientation. There are two possibilities to assign an antenna to the receivers.

First, an antenna can be assigned by simply selecting it in the Configuration tab of the tree window. There the currently used receiver antenna can easily be observed and changed, if necessary. An example of assigning a certain antenna to the receivers is shown in Fig. 4-33.

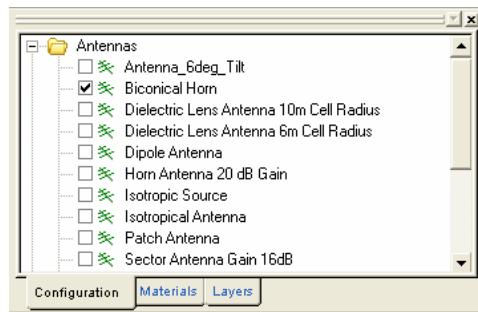


Fig. 4-33 Assigning a biconical horn antenna to the receivers

Second, when creating one or more new receivers, the antenna with its orientation can be specified. Also, the receiver antenna can be changed from a single receiver's context menu in the 2D view. These options are explained in Sections 4.2.3.2 and 4.2.3.4 in more detail.

Note that the antenna settings in the receiver settings dialog influence the antenna of *all* receivers. Also, the selected antenna in the Configuration tab of the tree window is updated accordingly.

#### 4.2.1.3 Inspecting the Antenna Configuration Data

The configuration data of an antenna can be inspected in the according configuration dialog. To access this dialog select Settings... from the respective antenna's context menu (by right-clicking the antenna item) or double-click the antenna item in the tree, alternatively. The configuration dialog of the antenna is shown in Fig. 4-34. A description of the configuration parameters is found in Table 4-7.

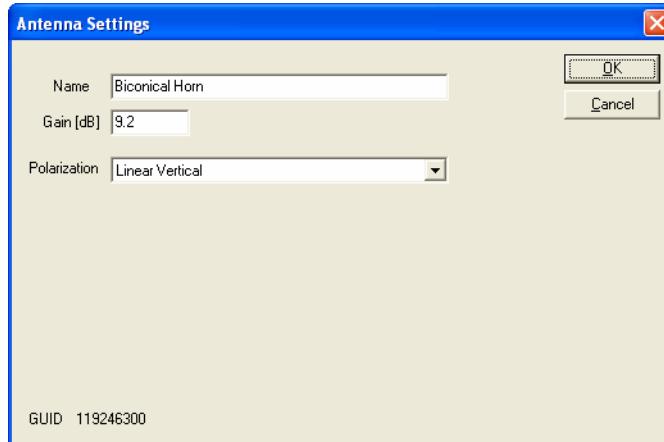


Fig. 4-34 Antenna configuration dialog

Table 4-7 Antenna parameters

| Parameter    | Description   | Unit |
|--------------|---|------|
| Name         | The identifier of the antenna. This is a string that is unique in the simulation configuration such that it can be distinguished from other antennas. | —    |
| Gain         | The antenna gain.   | dBi  |
| Polarization | The polarization for the receivers. Possible values are 'Linear Vertical', 'Linear Horizontal', 'Left Hand Circular', and 'Right Hand Circular'.      | —    |

#### 4.2.1.4 Viewing the Antenna Diagram

The horizontal and the vertical diagrams of an antenna can be shown in a common polar diagrams. This view can be accessed by the entry Show Antenna Diagrams from the corresponding antenna's context menu (by right-clicking the antenna item). A new window is opened that presents the horizontal and vertical antenna diagrams, see sample in Fig. 4-35.

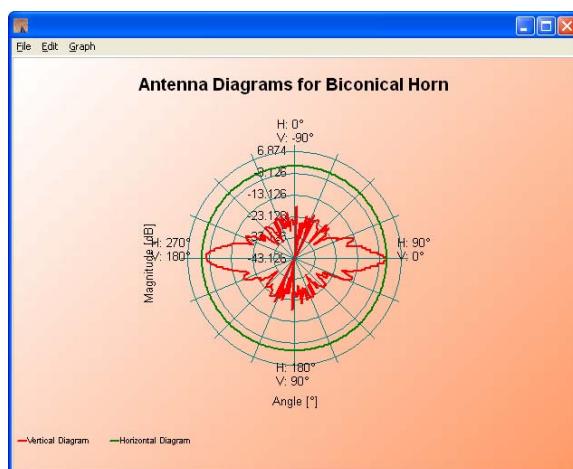


Fig. 4-35 Sample view of an antenna diagram

#### 4.2.1.5 Deleting an Antenna

An antenna can be deleted from the network configuration by selecting Delete... from that particular antenna's context menu. In order to avoid undesired erasure, an alert dialog appears that asks the user to confirm the deletion of the antenna.

Note that all references in those transmitters that used the deleted antenna are removed, too. Without assigning another antenna to those transmitters, the simulation will not run. If the receiver antenna was erased, a new one must be selected by checking its checkbox, see Section 4.2.1.2.

### 4.2.2 Transmitter



A transmitter represents a base station location in a network. It is characterized by its position, its height, has an antenna with certain orientation and polarization, and a transmit power and carrier frequency. During a coverage simulation, rays are emitted from the transmitter's antenna that propagate through the environment.

#### 4.2.2.1 Creating a Transmitter

Prior to placing a transmitter in the environment, it is useful to define a proper grid width. The new transmitter will then be set in this snapping grid, if the position has not been changed manually afterwards. To adjust the grid width, click the icon (tooltip Grid Size) from the paint toolbar. A dialog appears where the grid width can be modified, see Fig. 4-36.

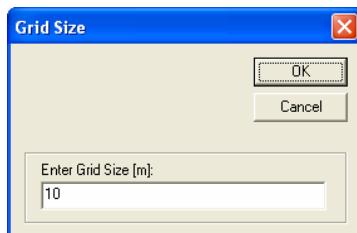


Fig. 4-36 The grid size dialog

Then a new transmitter can be placed at a given position in the 2D view. To use this function, the Transmitter Editing Mode must be entered by clicking the icon (tooltip Edit Transmitters) in the paint toolbar. A snapping grid is shown with the raster size as just determined. Then to set the transmitter, click the right mouse button at a position in a surface plot where a transmitter shall be added. A context menu will pop up. Choose Add... from this context menu to insert the new transmitter at the closest grid vertex. A configuration dialog for the transmitter appears that is shown in Fig. 4-37.

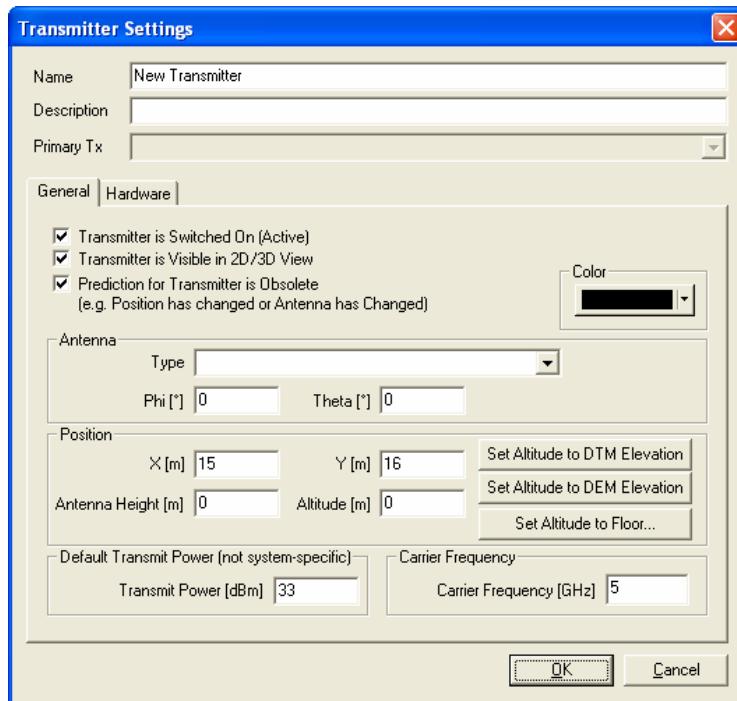


Fig. 4-37 Configuration dialog of a transmitter

The main parameters of the new transmitter can be adjusted in this dialog as, e.g., the name (ID), an optional description, the antenna with its orientation, the position, the transmit power, and the carrier frequency. Make sure that a valid antenna is selected since initially no antenna is chosen when the dialog is opened for the first time. The initial carrier frequency is taken from the current setting of the frequency in the preferences dialog (see Section 5.2.3), while the transmit power and the other hardware settings are based on the default settings in the `RPSDefaultParameters.ini` file. Other parameters can be adjusted as described later in Section 4.2.2.3.

The complete process of adding a new transmitter is shown in Fig. 4-38.

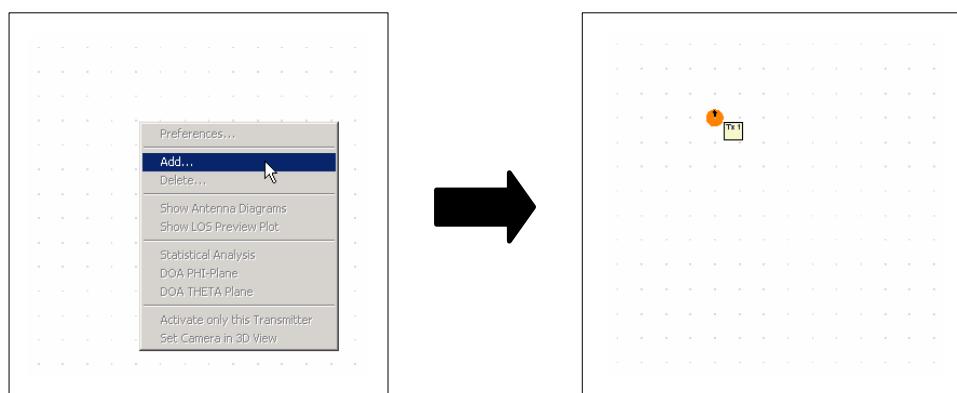


Fig. 4-38 A new transmitter is added in the Transmitter Editing Mode

#### 4.2.2.2 Moving a Transmitter

In the Transmitter Editing Mode ( icon in the paint toolbar), the position of a transmitter can easily be changed with the mouse in 2D view. To use this function, just click with the

left mouse button on the transmitter to be moved and hold the button pressed. Then it can be dragged with the mouse to another position with respect to the current snapping grid. When the mouse button is released again, the transmitter is placed at its new position.

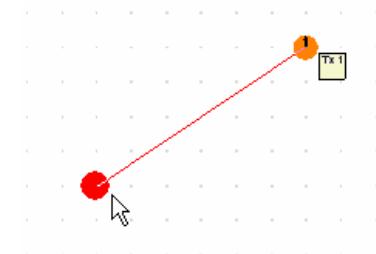


Fig. 4-39 Moving a transmitter

Alternatively, the position of a transmitter could be changed manually in its settings dialog. See Section 4.2.2.3 below on how to accomplish that.

Due to the position change, a recalculation for this transmitter would be necessary. That is why the Obsolete Flag of the transmitter will be automatically set after the movement. This state can be recognized on the orange color of the transmitter symbol.

#### 4.2.2.3 Inspecting Transmitter Data

The configuration data of a transmitter can be inspected in the according configuration dialog. To access this dialog select Settings... from the respective transmitter's context menu (by right-clicking the transmitter item in the tree window) or double-click the transmitter item in the tree, alternatively. Also the Transmitter Editing Mode ( icon in the paint toolbar) offers a possibility to display the transmitter configuration dialog by choosing Settings... from a transmitter's context menu in the 2D view. The configuration dialog has two tabs. The General tab is shown in Fig. 4-40, the corresponding parameters are explained in Table 4-8.

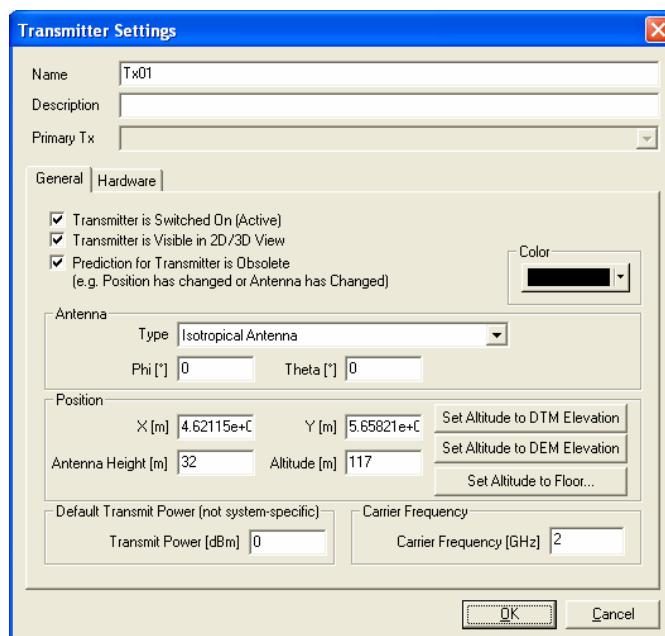


Fig. 4-40 General tab of the transmitter configuration dialog

Table 4-8 General transmitter parameters

| Parameter                              | Description   | Unit |
|--|---|------|
| Name                                   | The identifier of the transmitter. This is a string that is unique in the simulation configuration such that it can be distinguished from other transmitters.   | —    |
| Description                            | An arbitrary description for the transmitter.   | —    |
| Primary Tx                             | The superordinate primary Tx for this transmitter.  | —    |
| Transmitter is Switched On             | Normally, this field can be left unspecified for pure radio propagation predictions and coverage analyses; thus this parameter has no relevance. In a network, a secondary Tx would just function as a repeater for its primary Tx.   | —    |
| Transmitter is Visible in 2D/3D View   | If this flag is set, the transmitter is active, i.e. it will be considered in coverage simulations and its results are shown. In contrast to that, inactive transmitters are not considered in simulations and their results are disabled.  | —    |
| Prediction for Transmitter is Obsolete | This flag indicates if the prediction results for this transmitter shall be recalculated in a new simulation or not. After essential changes in the transmitter configuration, this flag is automatically set.  | —    |
| Color                                  | The color of the best cell area covered by this transmitter. It is used in the Best Server Plot.  | —    |
| <b>Antenna</b>                         |   |      |
| Type                                   | The antenna that is assigned to this transmitter. In the list field, one of the currently imported antenna templates can be selected.   | —    |
| Phi                                    | The horizontal orientation angle of the antenna (azimuth). This angle is oriented as defined by the polar coordinate system described in Section 3.2.2  | °    |
| Theta                                  | The vertical orientation angle of the antenna (tilt). This angle is oriented as defined by the polar coordinate system described in Section 3.2.2   | °    |
| <b>Position</b>                        |   |      |
| X                                      | The x-coordinate of the transmitter position (in Cartesian coordinates). The projection coordinate system is used.  | m    |
| Y                                      | The y-coordinate of the transmitter position (in Cartesian coordinates). The projection coordinate system is used.  | m    |
| Antenna Height                         | The height of the transmitter with respect to the altitude (part of z-coordinate in Cartesian coordinates). The projection coordinate system is used.   | m    |
| Altitude                               | The altitude (corresponding to ground level) for the transmitter position (part of z-coordinate in Cartesian coordinates). The projection coordinate system is used.  | m    |
| <u>Set Altitude to DTM Elevation</u>   | When pressing this button, the altitude parameter is automatically filled with the highest intersection point with a 3D object at the transmitter position. This could be, e.g., the roof of a building or, if no other objects are present at the transmitter position, the height of the DEM layer. | —    |
| <u>Set Altitude to DEM Elevation</u>   | When pressing this button, the altitude parameter is automatically filled with the DEM elevation at the transmitter position, if a valid DEM layer is defined there.  | —    |

| Parameter                       | Description   | Unit |
|---------------------------------|---|------|
| <u>Set Altitude to Floor...</u> | When pressing this button, the user can choose one of the available floor heights at the x/y position of the transmitter (e.g. within a building) and set the altitude parameter to that height. See more details below this table. | —    |
| Default Transmit Power          | The transmit power. It is initially filled with the default value from the <code>RPSDefaultParameters.ini</code> file.  | dBm  |
| Carrier Frequency               | The carrier frequency assigned to the transmitter. It is initially filled with the currently set frequency in the preferences dialog, see Section 5.2.3.  | GHz  |

**Note:** When performing propagation predictions, it is not necessary to distinguish between primary and secondary transmitters. This distinction was introduced to correctly handle repeaters in a network environment, which has no relevance for propagation predictions.

There are several ways to determine the altitude of a transmitter. Within a building, there is a very convenient function to place transmitters on a special story. This function can be invoked by pressing the Set Altitude to Floor... button in the transmitter configuration dialog. Then a dialog is shown that lists all available non-vertical polygons below/above or close to the transmitter with the respective height and material layer name. If, for example, in a building there are several floor planes, then they would be listed in this dialog as shown in Fig. 4-41.

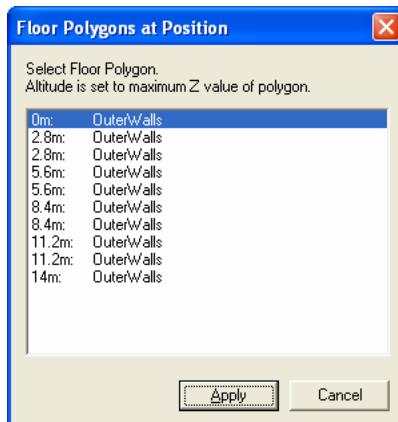


Fig. 4-41 Positioning a transmitter relative to a certain story in a building

After choosing one of the floor polygons and pressing the Apply button, the according height is assigned to the altitude parameter of the transmitter. This way transmitters can easily be placed on different floors with the same x/y coordinates.

Parameters regarding the hardware properties of the transmitter are summarized in the Hardware tab of the transmitter configuration dialog. This tab is shown in Fig. 4-42, the hardware parameters are described in Table 4-9.

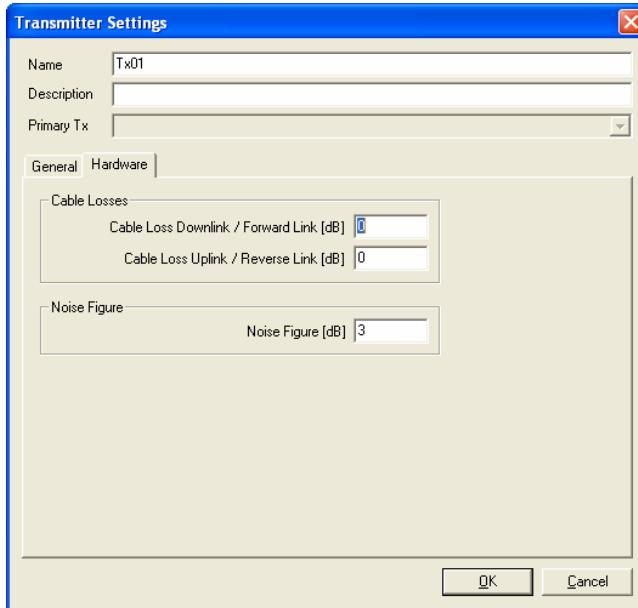


Fig. 4-42 Hardware tab of the transmitter configuration dialog

Table 4-9 Hardware properties of a transmitter

| Parameter           | Description  | Unit |
|---------------------|--|------|
| <b>Cable Losses</b> |  |      |
| Cable Loss Downlink | An additional cable loss in the downlink of the transmitter.                 | dB   |
| Cable Loss Uplink   | An additional cable loss in the uplink of the transmitter.                   | dB   |
| <b>Noise Figure</b> |  |      |
| Noise Figure        | The noise figure of this base stations receiver (for network related plots). | dB   |

In order to get a better overview of the settings of all transmitters together, it is possible to view an overview dialog by choosing Settings Overview... from an arbitrary transmitter's context menu in the tree window, refer to Fig. 4-43. The settings overview dialog is resizable and maximizable such that the whole screen size can be used to view the transmitter settings. Note that for secondary transmitters, the respective parameter values of the associated primary transmitter are shown in this overview table.



Fig. 4-43 Transmitter settings overview

Besides the practical overview, this dialog offers the possibility to quickly change the transmitter configurations. In particular, the following actions can be performed in the table:

- Individual parameter values can be changed.
- The currently selected line in the table can be duplicated by using the Duplicate Tx button. The additional line for the new transmitter is appended at the bottom of the table. This function can be used to quickly add new transmitters. Afterwards, new coordinates could be copied into the settings overview via the clipboard.
- The currently selected line in the table can be deleted by pressing <Del> which erases the according transmitter.
- Filters can be set individually for each column with the help of functions from the context menu. This enables a selective modification of a certain group of transmitters in conjunction with the copy/fill function (see next bullet).
- Parameter values can easily be copied in a column (copy/fill function).

#### How to selectively modify a network configuration?

This example shall explain how to exploit the filter functions and the copy/fill functions in table grids to change a configuration parameter in a well determined subset of transmitters using the settings overview dialog. The same basic mechanisms can also be applied in the other settings overview dialogs, of course.

It shall be demonstrated how to change the antenna orientation in all those transmitters that have a height different from 7m. First, display the transmitter settings overview dialog as described above. Scroll down until you find a transmitter with a z-coordinate (height) equal to 7m. Click into this field with the right mouse button to show the table grid context menu and choose the entry Set Filter. Then all those transmitters are excluded from the settings overview that have a height different from 7m.

| Location           | Name | Color | Active                              | Obsolete                            | Visible                             | X [m] | Y [m] | Altitude [m] | Height [m] | Carrier Frequency [GHz] | Transmit Power [dBm] | Antenna       |
|--------------------|------|-------|-------------------------------------|-------------------------------------|-------------------------------------|-------|-------|--------------|------------|-------------------------|----------------------|---------------|
| Stralburger Platz  |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 1395  | -695  | 40           | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Blockhaus          |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 260   | 625   | 15           | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Hilton             |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 276   | 40    | 7            | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Albertplatz        |      |       | <input checked="" type="checkbox"/> | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | 695   | 1245  | 7            | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Friedrichstädter K |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | -1415 | 680   | 30           | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Ibisotel Bastei    |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | -185  | -970  | 40           | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Albertstraße       |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 510   | 940   | 25           | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Haus der Presse    |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | -585  | 570   | 50           | 0          | Isotropic Ant           | 0                    | Isotropic Ant |
| Italienisches Dörf |      |       | <input type="checkbox"/>            | <input checked="" type="checkbox"/> | <input checked="" type="checkbox"/> | 11    | 260   | 7            | 0          | Isotropic Ant           | 0                    | Isotropic Ant |

To get a presentation of the transmitters that have a height different from 7m, the table grid filter must be inverted by selecting Invert Filter from the context menu. Then all the desired transmitters are shown in the settings overview dialog. Now let us set the antenna azimuth angle to 90°. Select the topmost field in the Reference Angle Phi [°] column and type '90'. Then select the complete column by clicking on the corresponding headline field and press <Ctrl-D> to copy the first value in the selection to the other selected fields of the column. Alternatively, you could choose Fill Down from the table grid context menu after the column was selected. Then the new setting '90' will be copied to all other transmitters that match the above stated condition.

If the table grid filter is removed then by selecting [Remove Filter](#) from the context menu, all transmitters are shown in the settings overview dialog again. You will notice that the antenna azimuth angle of the transmitters with 7m height has not been changed as was demanded.

#### **4.2.2.4 Viewing the Antenna Diagram of a Transmitter**

A separate window with the polar graphs of the horizontal and vertical antenna diagrams that are applied at the transmitter can be shown by choosing Show Antenna Diagrams from the context menu of that transmitter in the 2D view in the Transmitter Editing Mode. See also Section 4.2.1.4.

#### 4.2.2.5 Viewing an LOS Preview Plot

In order to get a quick and rough overview of the coverage area of a certain cell, all the receivers that have an LOS path to the respective transmitter can be colored in red. In addition to the LOS condition, also the maximally allowed pathloss in relation to the configured noise floor (see Section 5.2.3) is considered. This function can be reached in the Transmitter Editing Mode by choosing Show LOS Preview Plot from a certain transmitter's context menu. During the calculation a progress bar is shown. Fig. 4-44 shows an example of the LOS coverage area from a certain transmitter.

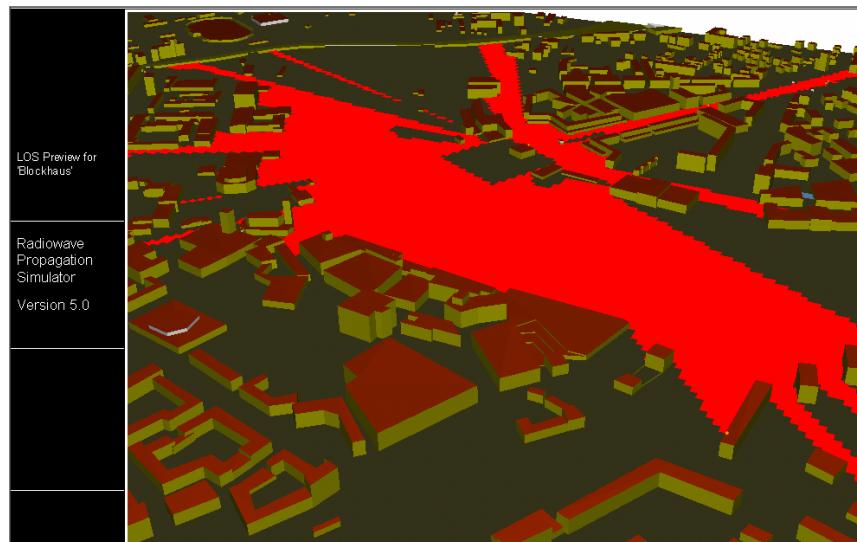


Fig. 4-44 Sample LOS preview plot

#### 4.2.2.6 Changing Transmitter States

For the incremental simulation mode, it is possible to make a base station obsolete, so that only for these obsolete base stations the radio channel characteristics are recalculated. When the position of a transmitter is changed, RPS automatically declares it as obsolete and removes all pixel values and rays related to this transmitter. The Obsolete Flag can be toggled in the transmitter settings dialog.

A transmitter can be set to active or inactive state. In inactive state, the transmit power parameter is set to 0 Watt (transmitter is effectively switched off). This property can be useful, when a simulation is performed with numerous transmitters and the coverage, SIR etc. should be determined for different transmitter configurations without any recalculation of the field strength. The Active Flag can be toggled in the transmitter settings dialog.

Alternatively, the checkbox of a transmitter item in the Configuration tab of the tree window can be selected to set the active state of a transmitter.

If only the coverage of a single base station shall be calculated, this transmitter can be solely activated by choosing the entry Activate only this Transmitter from the context menu of that transmitter in the 2D view in the Transmitter Editing Mode. As a consequence, all other transmitters in the network will be switched off, and a simulation for the single selected transmitter can be done.

The states of a transmitter are presented by different colors. Fig. 4-45 gives detailed explanations on the meaning of the distinct combination of state flags.



Fig. 4-45 Representation of different transmitter states

#### 4.2.2.7 Deleting a Transmitter

A transmitter can be deleted from the network configuration by choosing the menu entry Delete Transmitter from a transmitter item's context menu in the Configuration tab of the tree window. Alternatively, a transmitter could directly be deleted from the 2D view by selecting Delete... from that particular transmitter's context menu if the Transmitter Editing Mode ( icon in the paint toolbar) is active. In order to avoid undesired erasure, an alert dialog appears that asks the user to confirm the deletion of the transmitter.

When using the transmitter settings overview dialog, a transmitter can also be deleted by deleting its line from the table using <Del>.

### 4.2.3 Receiver



A receiver represents a location where the channel impulse response is observed. By arranging receivers along a line or in a matrix grid, the spatial distribution of the field strength can be efficiently determined.

A receiver is characterized by its position and its height. All receivers in a network configuration have an associated common antenna with certain orientation and polarization. After a simulation, the antenna pattern or its orientation/polarization could be changed quickly. This enables a very efficient way of analyzing the influence of different antennas at the receivers without any recalculation of the radio channel.

#### 4.2.3.1 Receiver Groups

For efficient maintenance and analysis possibilities, receivers are always assigned to a certain Rx Group. All created Rx Groups are listed in the Configuration tab of the tree window and can be separately switched on/off there. Only active Rx Groups are considered in a coverage simulation which opens very flexible configuration opportunities of the network. Fig. 4-46 shows an example with several Rx Groups.

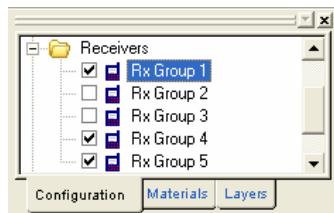


Fig. 4-46 Example of Rx Groups in a network configuration

The name of an Rx Group can be changed in its settings dialog. This is accessible from the entry Settings... in the context menu of the corresponding Rx Group item in the Configuration tab of the tree window. Alternatively, the dialog can be opened by double-clicking the Rx Group item in the tree. Fig. 4-47 shows the Rx Group settings dialog.

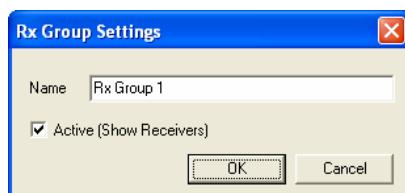


Fig. 4-47 Rx Group settings dialog

#### 4.2.3.2 Placing Receivers

Prior to placing receivers in the environment, it is useful to define a proper grid width. The new receivers will then be set in this snapping grid, if the positioning has not been changed manually afterwards. To adjust the grid width, click the icon (tooltip Grid Size) from the paint toolbar. A dialog appears where the grid width can be modified, see Fig. 4-48.

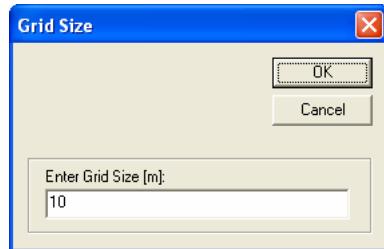
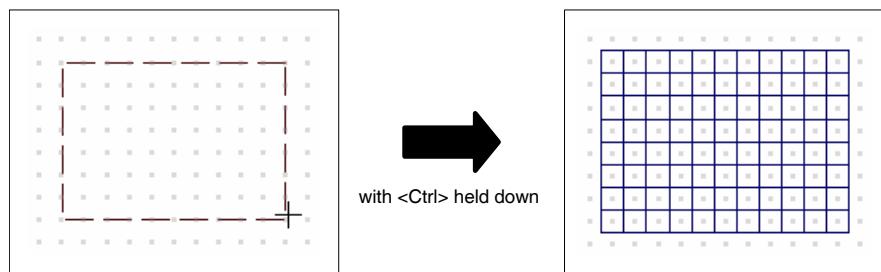


Fig. 4-48 The grid size dialog

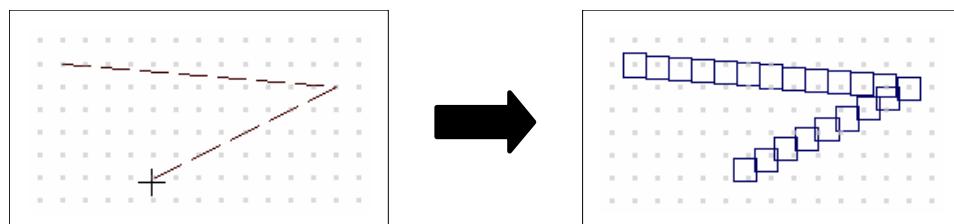
Then new receivers can be placed in the 2D view by using different Receiver Editing Modes. These functions can be activated by clicking the icons (tooltip Edit Receivers), (tooltip Line of Receivers), and (tooltip Field of Receivers) in the paint toolbar. A snapping grid is shown with the raster size as just determined. The three Receiver Editing Modes are explained shortly in the following.



- **Single receiver / rectangular array:** A single receiver can be placed at a point of the snapping grid by simply clicking with the left mouse button on the desired position. When holding down the <Ctrl> key in this Receiver Editing Mode during dragging the mouse pointer, a rectangular area can be defined where a grid of receiver would be placed.

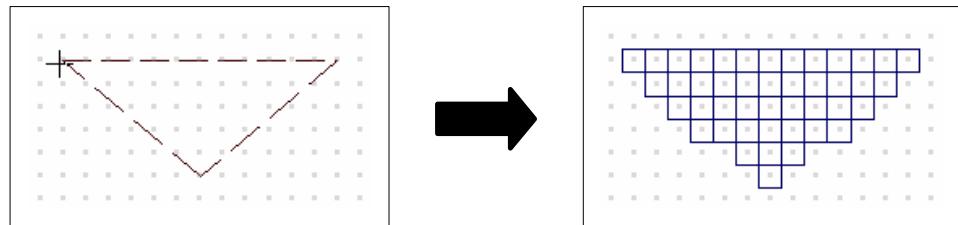


- **Receiver line:** Receivers can be placed along a polyline by setting each vertex of the line with the left mouse button. As soon as the right mouse button is pressed, the receivers will be placed. The distance between the receivers is determined by the predefined grid width.



- **Receiver grid:** Receivers can be placed in an arbitrary polygon by setting each vertex of the polygon with the left mouse button. As soon as the right mouse button is clicked,

the receivers will be placed. The distance between the receivers is determined by the predefined grid width.



After the placing operation has been finished, a configuration dialog for the receivers appears that is shown in Fig. 4-49. Depending on the Receiver Editing Mode, some of the parameters in the dialog might be disabled (grey). The configuration parameters of the receivers are summarized in Table 4-10.

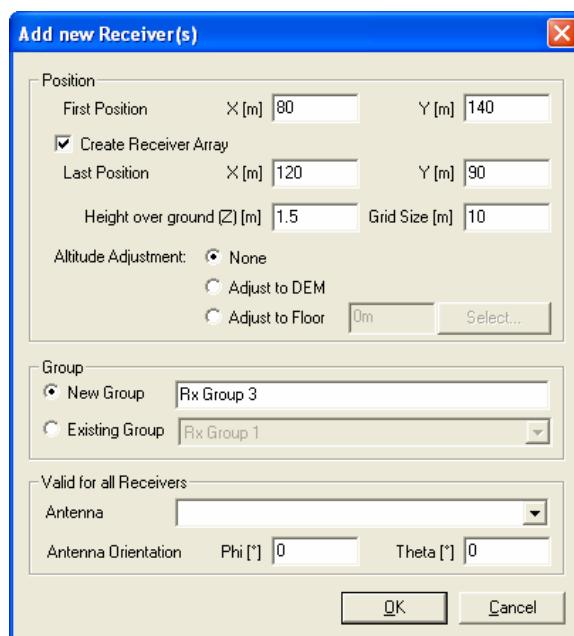


Fig. 4-49 Configuration dialog for new receivers

Table 4-10 Receiver configuration parameters

| Parameter             | Description  | Unit |
|-----------------------|--|------|
| First Position        | The first position of newly added receivers (single receiver / start point of receiver line / corner of receiver grid) in the horizontal plane (X/Y-coordinates).  | m    |
| Create Receiver Array | If this flag is set, the receivers are created in a grid (array) defined by the First and Last Positions. If the polygon drawing mode was used, this flag is automatically checked.                                      | —    |
| Last Position         | The last position of newly added receivers (end point of receiver line / opposite corner of receiver grid) in the horizontal plane (X/Y-coordinates). This parameter is not used for the polyline/polygon drawing modes. | m    |

| Parameter                      | Description   | Unit |
|--------------------------------|---|------|
| Grid Size                      | The currently used grid size. Initially it is equal to the predefined grid size, but it can be modified here. It determines both the distance between newly added receivers as well as their dimensions in the horizontal plane.  | m    |
| Height over ground (Z)         | The height of the newly created receivers.  | m    |
| Altitude Adjustment            | When defining the height (Z coordinate) of new receivers, there are different ways to determine this:<br><b>None:</b> An altitude of 0m is assumed. The height setting in the "Height over ground (Z)" field is used directly for the receivers.<br><b>Adjust to DEM:</b> The altitude for the receivers is determined by the DEM layer, i.e. the receivers are placed with an offset of "Height over ground (Z)" above the DEM.<br><b>Adjust to Floor:</b> The user can select a certain floor in a multi-story building for the receiver placement. When pressing the Select... button, a dialog is opened that lists all available floor polygons, see Fig. 4-41. This way of positioning is similar to the transmitter placing as described in Section 4.2.2.3. | —    |
| <b>Group</b>                   |   |      |
| New Group                      | If a new Rx Group shall be created for the new receivers, this radio button must be selected. The name of the new Rx Group can be edited in the associated text field. This new Rx Group is then assigned to the newly created receivers.   | —    |
| Existing Group                 | A radio button to choose an existing Rx Group for the new receivers. In the associated list field, one of the existing Rx Groups can be selected.   | —    |
| <b>Valid for all Receivers</b> |   |      |
| Antenna                        | The antenna that is assigned to all receivers. In the list field, one of the currently imported antenna templates can be selected.<br>This setting overrides any previous antenna assignment for <i>all</i> receivers.  | —    |
| Antenna Orientation            | The orientation of the main lobe of the receiver antenna. The angles $\varphi$ and $\theta$ are oriented in the polar coordinate system as defined in Section 3.2.2.<br>These settings override any previous antenna orientation for <i>all</i> receivers.  | °    |

Make sure that a valid antenna is selected since initially no antenna is chosen when the dialog is opened for the first time. It is possible to specify receiver positions (in terms of start/end vertex and grid width) that do not fit with the predefined grid size.

Receivers can be ordered and selectively considered for simulations by grouping them in Rx Groups. New receivers can be added to an existing Rx Group by choosing Existing Group along with a certain Rx Group selected in the according list field. All receivers belonging to one Rx Group are handled together, e.g. they can be collectively switched on or off for coverage simulations. Alternatively, a new Rx Group can be created for the new receivers by choosing New Group and specifying a new group name. Then these receivers can be treated separately further on.

If in an outdoor environment, receivers shall only be placed in the actual outdoor area and not inside the buildings, then it is recommended to first place receivers in the entire environment and later remove the receivers inside the buildings by means of the special

function Preferences -> Remove Receivers Below Roofs. Its usage is explained in detail in Section 4.2.3.6.

#### 4.2.3.3 Duplicating a Receiver Group

An existing Rx Group can be duplicated by choosing the entry Duplicate from an existing Rx Group's context menu in the Configuration tab of the tree window. Then a new Rx Group is created with a generically derived name, containing receivers at the same positions as the original Rx Group. This means that all receivers of the original Rx Group are copied for the new one.

This function can be used to easily compare simulation results under different simulation conditions (e.g. different propagation models or different propagation phenomena). By first selecting one Rx Group in a simulation run and later selecting the other Rx Group in a second run, propagation results can be gained at exactly the same positions under different conditions. These different results can be compared by simply switching between the Rx Groups.

#### 4.2.3.4 Inspecting Receiver Data

The configuration data of a single receiver can be inspected in the according settings dialog. To access this dialog, the Receiver Editing Mode ( icon in the paint toolbar) must be active. The one can select Settings... from the respective receiver's context menu (by right-clicking the receiver item in the 2D view). The configuration dialog is shown in Fig. 4-50, the parameters are explained in Table 4-11.

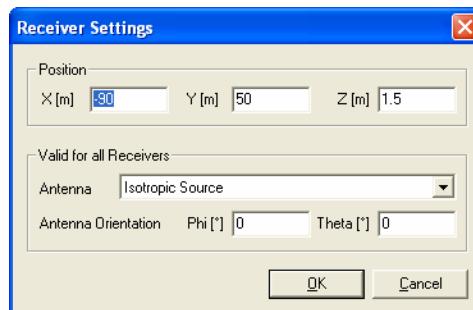


Fig. 4-50 Settings dialog of a single receiver

Table 4-11 Receiver parameters

| Parameter                      | Description  | Unit |
|--------------------------------|--|------|
| Position                       | The position of the receiver given in Cartesian coordinates.   | m    |
| <b>Valid for all Receivers</b> |  |      |
| Antenna                        | The antenna that is assigned to all receivers. In the list field, one of the currently imported antenna templates can be selected.<br>This setting overrides any previous antenna assignment for <i>all</i> receivers. | —    |

| Parameter           | Description  | Unit |
|---------------------|--|------|
| Antenna Orientation | The orientation of the main lobe of the receiver antenna. The angles $\varphi$ and $\theta$ are oriented in the polar coordinate system as defined in Section 3.2.2.<br>These settings override any previous antenna orientation for <i>all</i> receivers. | °    |

#### 4.2.3.5 Viewing the Antenna Diagram of the Receivers

A separate window with the polar graphs of the horizontal and vertical antenna diagrams that are applied at the receivers can be shown by choosing Show Antenna Diagrams from the context menu of an arbitrary receiver in the 2D view in the Receiver Editing Mode. See also Section 4.2.1.4.

#### 4.2.3.6 Deleting Receivers

Receivers can be deleted in the Receiver Deleting Mode that can be entered by choosing the icon from the paint toolbar (tooltip Delete Receivers) in the 2D view. Then a drag box can be drawn around all receivers to be deleted. Before erasing the receivers, a confirmation dialog is shown to avoid undesired deleting. Note that receivers could overlap if they were created at the same x-y position. If such a position was enclosed in the dragbox for deleting then *all* receivers at that position (irrespective of their height) are deleted.

Alternatively, a complete Rx Group can be deleted together with all receivers belonging to it. This can be accomplished by choosing Delete... from the respective Rx Group's context menu in the Configuration tab of the tree window.

If all receivers have been deleted from a network configuration, the 2D view is switched automatically back to the Receiver Editing mode ( icon in the paint toolbar) such that new receivers can be added.

A special function to selectively delete receivers inside buildings e.g. in an outdoor region is provided by the menu entry Preferences -> Remove Receivers Below Roofs. When using this function, all receivers that are placed below any non-vertical object (which are typically roofs in an outdoor environment) are removed. This can be useful to eliminate problems arising from the statistical analysis over both indoor and outdoor pixels. The principle of removing receivers below roofs in an outdoor environment is demonstrated in the two 2D view shown in Fig. 4-51.

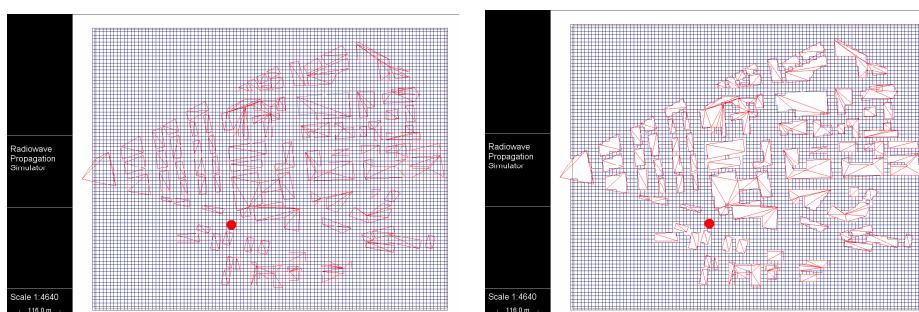


Fig. 4-51 Before (left) and after (right) removing receivers below roofs





## 5 Working with RPS — Simulation

This Chapter inducts the user into the details of performing radio channel prediction simulations. In the first Section, the different simulation algorithms are introduced. Moreover, recommendations are given on the deployment of the distinct methods in different scenarios. In the next Sections the simulation setup is described for both local and client-server simulations. The Chapter is concluded by an explanation on the implementation of user-defined propagation algorithms as well as the usage of dynamic simulations.

### 5.1 Simulation Algorithms

For the calculation of the wave propagation two general ray tracing algorithms are available. Both models allow the three-dimensional calculation of potential propagation paths. They are based on the ray launching technique, see Section 3.7.5. More detailed information on this topic can be found in [Cic94a], for example.

Furthermore, RPS provides semi-empirical COST prediction algorithms for both indoor and outdoor environments. Details are discussed in Section 5.1.2.

All RPS radio channel prediction algorithms are streamlined for multi-threaded architectures. This means that both on single-processor and multi-processor workstations, RPS will use the available resources optimally. Furthermore, the algorithms can be efficiently parallelized, thus enabling distributed computing in a TCP/IP LAN. This latter option is only available in RPS Enterprise.

#### 5.1.1 Ray Tracing Algorithms

##### 5.1.1.1 3D Ray Tracing Algorithms

Using the 3D ray launching approach, rays are launched from each transmitter into all directions in the three-dimensional space and traced until a given noise floor is reached. For a simulation, the distinct propagation phenomena (reflection, penetration, and diffraction) can be separately switched on/off. Usually, the number of those phenomena is unlimited. However, it is possible to restrict the number separately for each phenomenon. Additionally, a maximal excess delay of the rays can be specified. The 3D ray tracing algorithm can be selected by choosing the 3D Ray Tracing entry from the respective list field in the standard toolbar or, alternatively, by checking the Simulation -> 3D Ray Tracing menu entry.

In the three-dimensional case by using an angular resolution of 1°, approx. 42,000 rays have to be launched from each transmitter. Each ray has to be traced separately.

In order to increase the accuracy especially in large areas, ray splitting can be used. Then the width of the ray tubes will effectively be kept constant on average over the entire propagation path of a ray.

Besides local calculations on one computer, the 3D ray tracing algorithm also exists in a highly parallelized version for a virtually unlimited number of servers (RPS Enterprise only). This ray tracing algorithm allows for coverage simulations with unprecedented performance while keeping its very high level of accuracy. It can be selected by choosing the [3D Client-Server](#) entry from the respective list field in the standard toolbar or, alternatively, by checking the [Simulation -> 3D Client-Server](#) menu entry.

#### 5.1.1.2 Satellite Mode

The satellite mode is a special case of the 3D ray tracing algorithm. The satellite mode is automatically enabled as soon as a transmitter has a height of more than 1000km away from the environment. In this case, rays are no longer launched from the transmitter position into all directions. Instead, parallel rays are created which arrive at the environment (obstacles and receivers) from the direction of the transmitter. For the satellite mode it is recommended to set a much finer angular resolution of the rays because of the large distance of the transmitter in order to achieve a proper accuracy.

#### 5.1.1.3 2.5D Ray Tracing Algorithms

Common 3D procedures require a high amount of compute time due to the huge number of rays to be traced throughout the environment. This computational effort can be reduced drastically if only rays are launched from a transmitter that can contribute to the receiving field strength at a given position with a high probability. For estimation of such propagation paths a special ray pre-selection algorithm is used. This so-called 2.5D ray tracing algorithm can be selected by choosing the [2.5D Ray Tracing](#) entry from the respective list field in the standard toolbar or, alternatively, by checking the [Simulation -> 2.5D Ray Tracing](#) menu entry.

The plain 2.5D algorithm can only find rays reflected/transmitted at vertical obstacles and diffracted at (almost) vertical diffraction edges. Especially in indoor environments this restriction is completely sufficient. Additionally, there are two options. As one of them, the user can choose to consider single reflections at non-vertical obstacles, too. The other option is to trace over-rooftop diffracted rays which is only useful in outdoor environments.

**Advantages:** By using the 2.5D algorithm, the required computational effort can be reduced by up to a factor of 50 compared to the 3D algorithm in practice! The wave propagation calculation is fully polarimetric and also three-dimensional (see limitations below). This simulation method can be used in urban, suburban or indoor scenarios efficiently if the height of the transmitter(s) is lower than or in the order of the surrounding building structure.

**Limitations:** The diffracted rays can be reflected, transmitted, or diffracted in a potentially unlimited order. The consideration of reflections at non-vertical objects is restricted to single reflections which is still sufficient in most practical cases.

In analogy to the 3D algorithm, also the 2.5D procedure is optimized for multi-threaded architectures which promotes an unlimited parallelization of the algorithm as well. The parallelized simulation mode is only available in RPS Enterprise. It can be selected by choosing the [2.5D Client-Server](#) entry from the respective list field in the standard toolbar or, alternatively, by checking the [Simulation -> 2.5D Client-Server](#) menu entry.

#### 5.1.1.4 Best Fitting Algorithm for Ray Tracing Simulations

Both ray tracing algorithms have advantages and limitations. In general, the fully three-dimensional (3D) algorithm can be used in any kind of environment, independent of the positions of transmitters or receivers.

The 2.5D algorithm has a great advantage in urban, suburban and indoor environments if the surrounding buildings or walls are higher than or in order of the transmitter(s)' height. Under these conditions the algorithm works much faster than 3D ray tracing and gives accurate results.

If the transmitters are placed above the building height level, the 3D algorithm is more applicable because the ray pre-selection algorithm of the 2.5D algorithms will not work that efficiently in such environments. To reduce the computational effort when using the 3D algorithms in such configurations, the  $\vartheta$  range of launched rays can be limited appropriately. If receiver positions are far away from the transmitter, the  $\vartheta$  step size should be reduced to values lower than  $1^\circ$ . Alternatively, ray splitting could be used.

Moreover, the 3D algorithm should be used if a high accuracy of the DoA information is required at the receiver positions. Since the 2.5D algorithm performs a pre-selection of rays, it is possible that the impulse responses do not contain all relevant paths for a DoA analysis.

#### 5.1.2 Empirical and Semi-Empirical Prediction Algorithms

Empirical field strength predictions are much faster than ray tracing algorithms, but considerably less accurate in most cases. These algorithms consider only a direct path from a transmitter to a receiver position. The attenuation depends on the distance, the frequency, and the building structure between transmitter and receiver.

Moreover, empirical algorithms cannot determine the multi-path radio channel characteristics. E.g. delay spread or angular distributions cannot be calculated from simulation results obtained by empirical models.

In general, RPS stores each path from a transmitter position to a receiver position as a ray. In case of an outdoor model, the ray propagates over the roofs of the buildings. On the other hand, an indoor model ray goes straight through the walls towards the receiver.

Besides some built-in empirical and semi-empirical prediction models, there is the possibility to write your own prediction models using the RPS Plug-in Interface. A detailed description of how to modify plug-in parameters and how to write user-defined prediction models using plug-ins is discussed in Section 5.4.2.

##### 5.1.2.1 Empirical Model for Outdoor Environments

In outdoor environments, it is assumed that the dominant path from a transmitter to a receiver position is a function of distance, frequency, mean building height, number of buildings, and some additional parameters about the building structure around the receiver as, e.g., incident angles or street width. For this purpose, the COST 231 Walfish-Ikegami (WI) model was implemented in RPS. It can be selected by checking the [Simulation -> COST 231 WI \(Outdoor\)](#) menu entry. The typical ray path over the roofs of the buildings calculated in this model is apparent from Fig. 5-1.

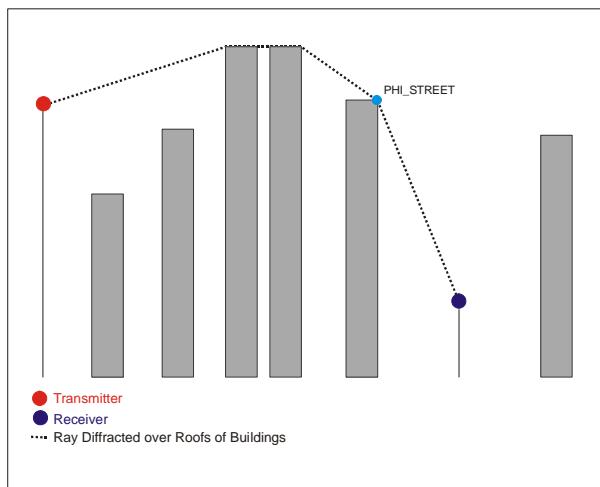


Fig. 5-1 Realization of the COST 231 WI outdoor propagation model in RPS

### 5.1.2.2 Empirical Model for Indoor Environments

In indoor environments, the pathloss is determined using, besides frequency and distance, information about the number of intersecting walls, material type, or their thickness (or attenuation). For this purpose, the COST 231 Multi-Wall Model (MWM) was implemented in RPS. It can be selected by checking the **Simulation -> COST 231 MWM (Indoor)** menu entry. In this model, the ray path goes straight through the walls as indicated in Fig. 5-2.

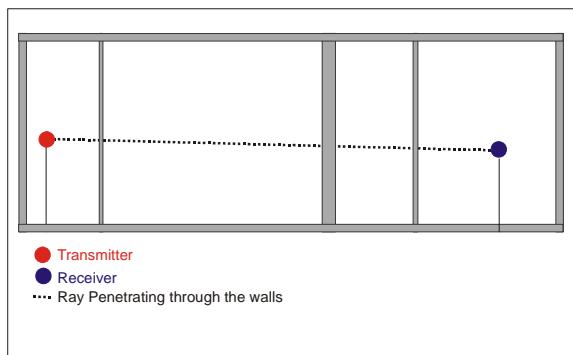


Fig. 5-2 Realization of the COST 231 MWM indoor propagation model in RPS

## 5.2 Simulation Setup

### 5.2.1 General Simulation Workflow

For a coverage simulation it is assumed that the environment and the network setup (transmitters and receivers) have been configured properly. Then the following steps have to be executed:

- Make sure that all relevant material layers are active (have their checkbox flagged in the Materials tab of the tree window).
- Make sure that all transmitters are active (have their checkbox flagged in the Configuration tab of the tree window) for whom the radio channel shall be predicted. Moreover, all desired Rx Groups must be active in order to be considered in the simulation.

- If diffraction shall be considered, it is recommended to have the Preferences -> Auto Detect Diffraction Edges before each Simulation Run entry checked. This will ensure that all relevant diffraction edges will be automatically determined when the simulation is started.
- Make all necessary settings in the Simulation tab of the preferences dialog. It can be accessed by clicking the icon from the standard toolbar or by choosing Preferences -> General Preferences from the menu bar. Specifically, all the propagation effects can be selectively enabled/disabled and optionally restricted there.
- Select one of the available channel prediction algorithms from the standard toolbar or the Simulation menu, respectively.
- Start the simulation by clicking the icon from the standard toolbar or by choosing Simulation -> Run from the menu bar. The simulation progress is shown in the message window.

In the following Subsections, first some useful hints about how to perform RPS channel predictions most efficiently are given. Later the setup of the simulation in the preferences dialog is explained in detail.

## 5.2.2 Increasing Prediction Efficiency

There are several mechanisms in RPS that enable an efficient exploitation of the available algorithms and help the user to avoid unnecessary recalculations. Thus RPS can be used in the most time-effective manner for radio network planning. The underlying principles are introduced in this Subsection.

### 5.2.2.1 When is a Channel Recalculation Necessary?

In a coverage simulation, the field strength is calculated regardless any radiation pattern or transmit power. The only criterion to stop the tracing of a ray is the maximal pathloss that is derived from the Tx power, the Tx antenna gain, and the global noise floor. The resulting power budget is shown in Fig. 5-3. In both cases, only the relative pathloss is essential for radio channel prediction.

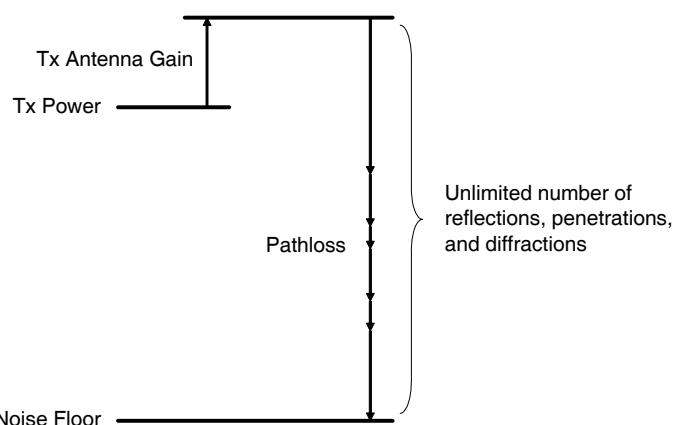


Fig. 5-3 Power budget

These principles allow for a reduction of the necessary simulation runs in the radio network planning process to a minimum. In particular, no recalculation of the radio channel is required if the environment, the receiver and transmitter positions, and the polarization state of all transmitters will remain constant. Fig. 5-4 gives an overview of the situations where

new planning results can be gained by just modifying parameters (left hand side) or where the radio channel must be recalculated (right hand side).

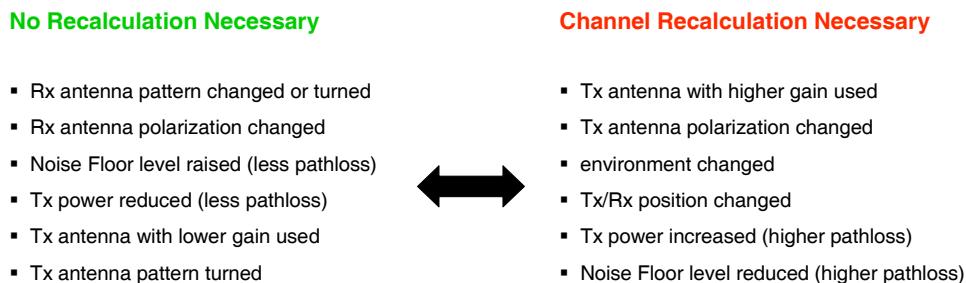


Fig. 5-4 Conditions for the necessity of channel recalculations

Any modification of the Rx antenna and any effective reduction of the maximal pathloss (due to Tx antenna, Tx power, or noise floor) is possible without any re-simulation. The resulting channel characteristics of such changes are presented in the graphical user interface of RPS immediately.

If you are not so familiar with RPS, a simulation using isotropic sources as antenna patterns should be the most suitable way at the beginning of your planning activities. The resulting antenna diagrams can be applied later as a next step after the simulation process. The Getting Started Chapter 2 gives a brief demonstration of how to efficiently work with RPS by exchanging antennas without recalculation.

#### 5.2.2.2 Incremental Simulations

The simulation algorithms allow for incremental calculations where only the propagation channel of those transmitters is recalculated that have been changed in essential parameters (as, e.g., the position) compared to prior simulations.

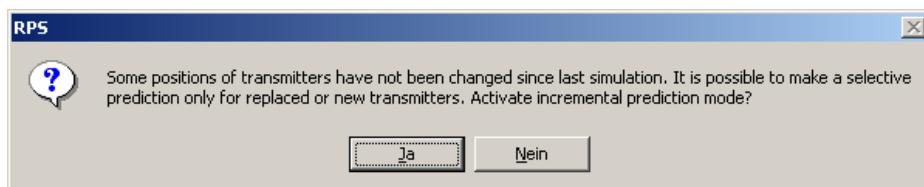


Fig. 5-5 Selection dialog for activation of incremental simulations

The incremental prediction mode can be switched on by checking the [Simulation -> Suggest Incremental Predictions](#) menu entry. It exploits the obsolete status flag of the transmitters that is automatically set if the according transmitter was changed essentially. If the environment consists of more than one base station and prior simulations were performed, RPS gives you the choice to recalculate the propagation paths only of those transmitters the position of which has been changed in the meantime, see dialog in Fig. 5-5. The propagation results for all other transmitters (those who have not the obsolete flag set) are retained. Incremental simulations are possible for all simulation algorithms, also using parallelized computations (RPS Enterprise only).

If the environment structure was changed, you should press No in the dialog from Fig. 5-5 in order to recalculate the coverage for the entire network. Alternatively you can manually set transmitters obsolete (see Section 4.2.2). In incremental prediction mode, the channel is only recalculated for the obsolete base stations.

It is generally recommended to deactivate the incremental prediction mode if a new calculation with a different prediction algorithm shall be performed. In this case, the old results need to be overridden safely.

### 5.2.2.3 Hybrid Simulation Mode

The built-in ray launching algorithms are highly efficient and accurate propagation models. Due to their deterministic nature, it is possible that in a large receiver array of an outdoor scenario, some receivers might not be hit by rays. In order to obtain entirely covered pathloss matrices e.g. for a radio network planning tool, it can be desirable to fill these uncovered areas.

RPS offers the hybrid mode for such purposes. It can be enabled for the next following simulation by selecting the option Simulation -> Hybrid Mode for next Simulation from the menu. Then rays are detected at only those receivers that have not been previously hit. If, for example, an empirical model is used for the hybrid mode, the possible holes in a prediction map can be easily closed. The hybrid mode flag in the Simulation menu is automatically disabled after one simulation. Nevertheless, the hybrid mode could be applied iteratively with arbitrary channel prediction algorithms. Fig. 5-6 illustrates the principle of the hybrid simulation mode.

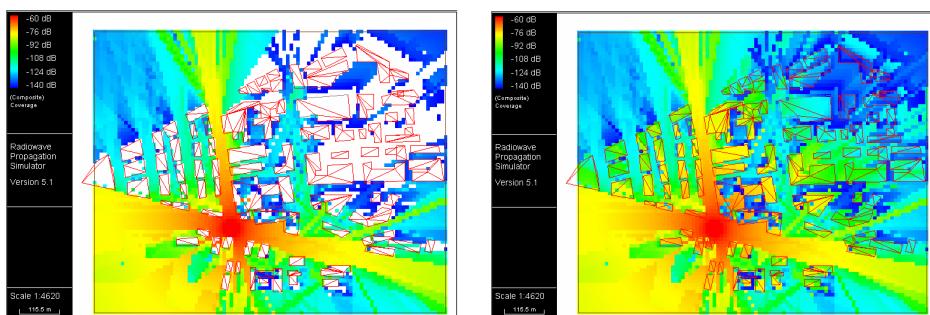


Fig. 5-6 Result of ray launching (left) and COST231 model (right) in hybrid mode

### 5.2.2.4 Active and Inactive Transmitters

An inactive transmitter describes a base station that is switched off. Therefore, the field strength is not predicted for inactive transmitters. A deactivation of a transmitter position after the simulation gives the user the chance to observe coverage plots for different transmitter configurations (only a subset of base stations is in active state) simultaneously without any recalculation.

## 5.2.3 Configuring a Simulation

The main simulation settings are configured in the Simulation tab of the preferences dialog. It can be accessed by clicking the icon from the standard toolbar or by choosing Preferences -> General Preferences from the menu. This configuration dialog embraces several types of settings that shall be introduced shortly:

- **Angular Settings:** RPS allows to make global angular settings, relevant for all transmitters. These settings determine the angular resolution of the launched ray tubes and the angular range at each transmitter. It is possible to automatically detect the angular settings with respect to the minimal object size and the environment dimensions.
- **Channel Parameters:** The center frequency and the assumed signal bandwidth can be specified. Note, however, that only the center frequency influences the radio propagation properties during a simulation. The bandwidth is only relevant for the result analysis. Also, a system template can be selected to automatically determine the proper carrier frequency and bandwidth.
- **Propagation Phenomena:** In all propagation models, the distinct propagation phenomena such as reflection, penetration, and diffraction can be selected separately. In general, the number of consecutive phenomena is not limited. However, it is possible to induce restrictions on the number of certain propagation phenomena and on the maximum excess delay of the rays.
- **Propagation Options:** For 2.5D/3D ray tracing, additional options for reflections at non-vertical obstacles (2.5D only) and over-rooftop diffraction (in outdoor environments) can be selected. Furthermore, ray splitting can be chosen to compensate for the negative effect of reducing ray tube resolution in case of far propagating rays.
- **Miscellaneous:** Some more simulation options can be determined like determining whether the ray paths (incl. deviating points) shall be stored in addition to the pure impulse responses at the receivers. Also the number of simulation threads per microprocessor can be specified in order to efficiently exploit the available hardware.

The Simulation tab of the preferences dialog is shown in Fig. 5-7 whereas the parameters are described in brief in Table 5-1. Some further explanations are given below the table.

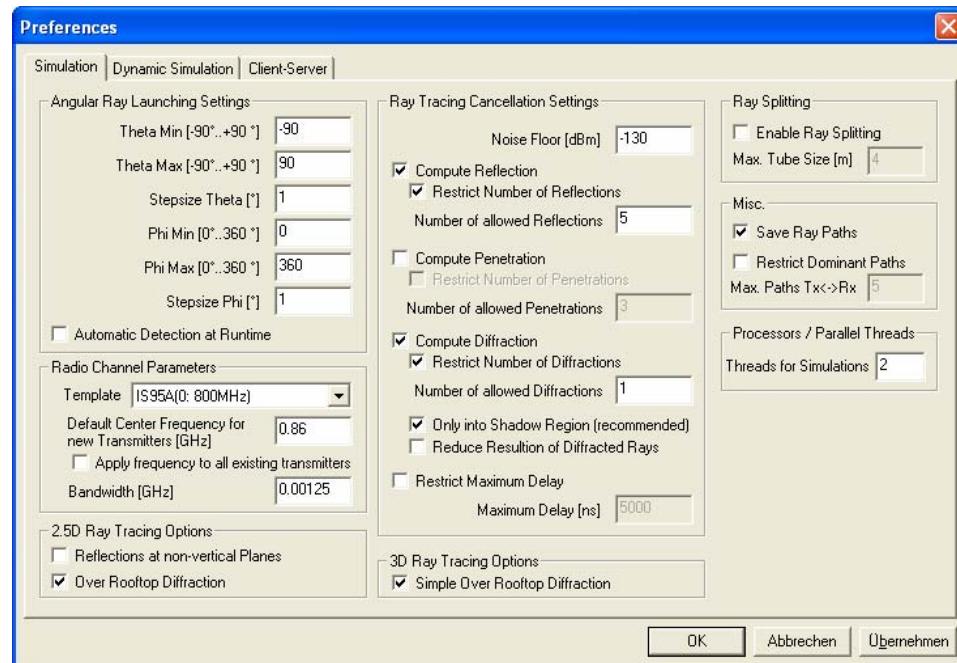


Fig. 5-7 Simulation tab of the preferences dialog

Table 5-1 Ray tracing simulation parameters

| Parameter                                     | Description  | Unit     |
|---|--|----------|
| <b>Angular Ray Launching Settings</b>         |  |          |
| Theta Min                                     | The starting elevation angle of the ray launching at the transmitters. This angle must be in the top-to-bottom range of $(-90^\circ, 90^\circ)$ where $0^\circ$ corresponds to a horizontally launched ray.  | $^\circ$ |
| Theta Max                                     | The ending elevation angle of the ray launching at the transmitters. This angle must be in the top-to-bottom range of $(-90^\circ, 90^\circ)$ where $0^\circ$ corresponds to a horizontally launched ray. Additionally, it must be greater than or equal to Theta Min. | $^\circ$ |
| Stepsize Theta                                | The angular resolution of the ray tubes with regard to the elevation angle.  | $^\circ$ |
| Phi Min                                       | The starting azimuth angle of the ray launching at the transmitters. This angle must be in the range $(0^\circ, 360^\circ)$ where $0^\circ$ corresponds to North direction.  | $^\circ$ |
| Phi Max                                       | The ending azimuth angle of the ray launching at the transmitters. This angle must be in the range $(0^\circ, 360^\circ)$ where $0^\circ$ corresponds to North direction. Additionally, it must be greater than or equal to Phi Min.                                   | $^\circ$ |
| Stepsize Phi                                  | The angular resolution of the ray tubes with regard to the azimuth angle.  | $^\circ$ |
| Automatic Detection at Runtime                | If this flag is set, the angular settings are detected automatically at runtime. After simulation start they can be inspected in the preferences dialog.   | —        |
| <b>Radio Channel Parameters</b>               |  |          |
| Template                                      | The system template for which the carrier frequency and the bandwidth are to be adopted automatically. Both frequency and bandwidth can be freely modified afterwards.   | —        |
| Default Center Frequency for new Transmitters | The default carrier frequency that is assigned newly created transmitters. Inside each transmitter, the frequency can be freely modified afterwards  | GHz      |
| Apply frequency to all existing transmitters  | This temporary flag can be set in order to assign the configured center frequency to all transmitters in the network once when clicking the <u>OK</u> button.  | —        |
| Bandwidth                                     | The assumed signal bandwidth for the result analysis. This value has no impact on the simulation.  | GHz      |
| <b>Ray Tracing Cancellation Settings</b>      |  |          |
| Noise Floor                                   | The global noise floor level for the criterion to stop ray tracing.  | dB/dBm   |
| Compute Reflection                            | This option enables the tracing of reflected propagation paths for 3D and 2.5D ray tracing. The number of possible multiple reflections is not limited in general.   | —        |
| Restrict Number of Reflections                | If this flag is set, the number of reflections is restricted by the number given in the following parameter.   | —        |
| Number of allowed Reflections                 | The maximal number of reflections in one ray if the corresponding restriction flag is set.   | #        |

| Parameter                                | Description  | Unit |
|--|--|------|
| Compute Penetration                      | This option enables the consideration of propagation paths penetrating through obstacles for 3D and 2.5D ray tracing. The number of possible multiple penetrations is not limited in general.                    | —    |
| Restrict Number of Penetrations          | If this flag is set, the number of penetrations is restricted by the number given in the following parameter.  | —    |
| Number of allowed Penetrations           | The maximal number of penetrations in one ray if the corresponding restriction flag is set.  | #    |
| Compute Diffraction                      | This option enables the tracing of diffracted propagation paths for 3D and 2.5D ray tracing. The number of possible multiple diffractions is not limited in general.   | —    |
| Restrict Number of Diffractions          | If this flag is set, the number of diffractions is restricted by the number given in the following parameter.  | —    |
| Number of allowed Diffractions           | The maximal number of diffractions in one ray if the corresponding restriction flag is set.  | #    |
| Only into Shadow Region<br>(recommended) | With this option, rays are only diffracted into the shadow region of a diffraction edge. See Section 3.7.3 for a definition of the shadow region.  | —    |
| Reduce Resolution of Diffracted Rays     | With this option, the resolution of the ray tubes of diffracted rays is automatically reduced. This can also reduce the computational effort considerably.   | —    |
| Restrict Maximum Delay                   | This option restricts the maximum excess delay of the propagation paths for 3D and 2.5D ray tracing. Longer rays (with higher delay) are not considered.   | —    |
| Maximum Delay                            | The maximal delay until which the rays are traced if the corresponding restriction flag is set.  | ns   |
| <b>2.5D Ray Tracing Options</b>          |  |      |
| Reflections at non-vertical Planes       | Additional option for 2.5D ray tracing that considers also rays that are single reflected at non-vertical obstacles.   | —    |
| Over Rooftop Diffraction                 | Additional option for 2.5D ray tracing that considers also rays that are diffracted over the roofs of buildings (only useful in outdoor scenarios).  | —    |
| <b>3D Ray Tracing Options</b>            |  |      |
| Simple Over Rooftop Diffraction          | Additional option for 3D ray tracing that separately calculates over-rooftop diffracted rays to accelerate computation speed (only useful in outdoor scenarios).   | —    |
| <b>Ray Splitting</b>                     |  |      |
| Enable Ray Splitting                     | This option enables ray splitting and, thus, ensures that the ray tube width is kept approx. constant over the entire ray distance.  | —    |
| Max. Tube Size                           | The maximally allowed width (diameter) of the ray tube after which the ray would be split.   | m    |
| <b>Miscellaneous</b>                     |  |      |
| Save Ray Paths                           | If this flag is set, the paths of all received rays including all intermediate deviating points are stored in the according RPS project. Otherwise, only the pure impulse responses at the receivers are stored. | —    |

| Parameter                            | Description   | Unit |
|--------------------------------------|---|------|
| Restrict Dominant Paths              | If this flag is set, the number of detected rays at each receiver will be limited to a maximum of $n$ dominant paths where $n$ is given by the following parameter.   | —    |
| Max. Paths Tx<->Rx                   | The maximal number of detected dominant rays for each Tx/Rx pair that is detected if the respective option is selected.   | #    |
| <b>Processors / Parallel Threads</b> |   |      |
| Threads for Simulations              | Using modern microprocessor architectures, the ray tracing algorithms can be distributed among several threads. This value specifies the number of threads used. Generally, a value of 4 or more is recommended which will exploit the available hardware most efficiently. | —    |

#### 5.2.3.1 Angular Settings

The angular settings are valid for all transmitters. Within a range of  $\vartheta$  and  $\varphi$  rays are launched into all directions of the three dimensional space. By limiting the region, the simulator can be much faster. For instance, if the transmitter positions are higher than the highest obstacle,  $\vartheta_{\min}$  can be set to  $0^\circ$  such that only rays are generated that have initial angles of  $\vartheta \geq 0^\circ$ .

The step size parameters of  $\vartheta$  and  $\varphi$  determine the angular resolution of the launched ray tubes (see section 3.7.5). In the literature, angular resolution of ray tubes in the range of  $0.5^\circ$  to  $5^\circ$  are used. Angles of one degree are a good compromise between spatial resolution and required computational effort. In general, smaller step size angles require higher computation times. For larger step sizes, the accuracy of the ray tracing algorithms can be increased by using ray splitting, see Section 5.2.3.3.

Generally, a setting of  $1^\circ$  for the angular resolution of the ray tubes is a good choice. Ray splitting can be used to keep a high accuracy even with less angular resolution. It is also possible to automatically detect proper angular settings at runtime.

#### 5.2.3.2 Channel Parameters

The center frequency defines the default value for the (central) frequency of the radio waves that are radiated from each transmitter. However, despite the default setting, each transmitter can have its individual carrier frequency, refer to Section 4.2.2.3. Free space loss, attenuation due to penetrated obstacles as well as reflection and diffraction coefficients depend on this frequency.

The signal bandwidth has no influence on the simulation results. It is only relevant for charts and matrix export. The time resolution  $\Delta\tau$  of the channel impulse response is equal to the inverse of the bandwidth of the radio channel. For instance, a bandwidth of 1 MHz results in a resolution in the time domain of 1  $\mu$ s.

The available system templates that can be selected to set the default frequency and bandwidth are defined in the `RPSNetworkSystems.ini` file.

### 5.2.3.3 Propagation Phenomena and Options

A sensitive parameter is the noise floor value. It defines the criterion when the tracing of a ray can be finished. The max. pathloss is the difference of transmit power (plus radiation pattern gain) and the noise floor, as defined in the power budget given in Fig. 5-3. An increase of 10 dB can lead to a much higher required computation time.

It is a basic principle of the ray launching approach that there is no limitation in the order and number of the different propagation phenomena. For a flexible simulation configuration and for a proper adaptation of the propagation algorithms to the requirements of the environment at hand, reflection, penetration, and diffraction can be separately disabled or restricted in the number of occurrences. A proper selection of these options can reduce the computation time dramatically.

Especially, restricting the extent of diffraction can lead to a significant speedup of the simulation. Further opportunities to accomplish this are to only create diffracted rays into the shadow region behind a diffraction edge or to generally reduce the ray tube resolution after a diffraction. Of course, this is always a trade-off between simulation speed and accuracy which is highly dependent on the particular environment.

In spite of these global settings, the distinct propagation phenomena can also be selectively switched on or off for each material layer separately. This concept offers the greatest possible flexibility of configuration and enables another possibility for another reduction of the computational effort. For example, it is not useful to consider diffraction at a DEM layer, even if the automatic detection of diffraction edges might determine a potentially high number of diffraction edges. Independent of that, diffraction can be considered at buildings if the global flag and the according flag of the building layer are selected.

### 5.2.3.4 2.5D and 3D Ray Tracing Options

The pure 2.5D ray tracing algorithms can only consider propagation effects at vertical obstacles and diffraction edges. Investigations have shown that pure 2.5D ray tracing usually provides results that only differ less than 2% from those gained by 3D ray tracing. However, there are two options that improve the results especially in terms of angular spread.

The first of those options allows to consider single reflected rays at non-vertical obstacles. Especially in indoor scenarios, this provides a much better impression of the angular spread. However, the contribution of such reflected rays at floor or ceiling is usually very small.

The other option is the over-rooftop diffraction that should only be used in outdoor scenarios. It is also available for 3D ray tracing. If this option is activated, dedicated ray paths over the roofs are searched that are diffracted at (almost) horizontal diffraction edges. Usually such diffraction edges are the edges of roofs. The algorithm is similar to the implementation of the COST 231 WI model, see Section 5.1.2.1. Its principle is drawn in Fig. 5-8. In 3D mode, the over-rooftop diffracted rays are calculated separately. In this case, the normal diffraction algorithm only considers vertical diffraction edges which considerably increases the computation speed.

Note that the diffraction processes during over-rooftop propagation do not contribute to the simulation statistics that are collected and presented in the statistics dialog (see Section 6.1). The reason for this is that there is neither a restriction on the number of admissible diffractions for the over-rooftop propagation.

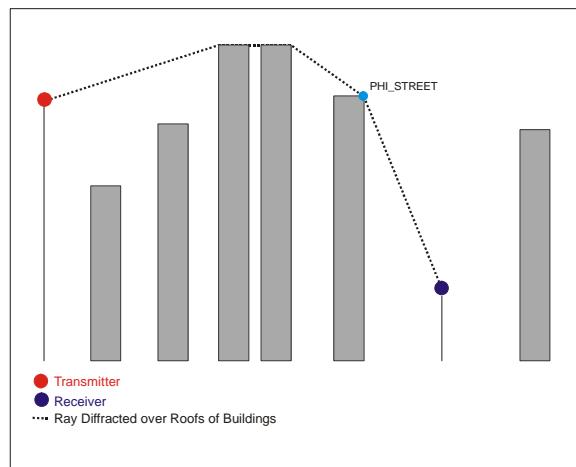


Fig. 5-8 Principle of over-rooftop diffraction

#### 5.2.3.5 Ray Splitting

A disadvantage of the ray launching approach is that adjacently launched rays diverge with increasing distance. The result is a dilution of the spatial resolution. This effect can be limited by using ray splitting as demonstrated in Fig. 5-9. After having reached a configurable ray tube width (max. width), the ray tubes are divided into somewhat tighter ray tubes. Thus, the mean spatial resolution remains constant. The maximal ray tube size should be chosen in relation to the minimal object size that exists in the environment to be investigated. Note that with activated ray splitting, the required computation time may increase.

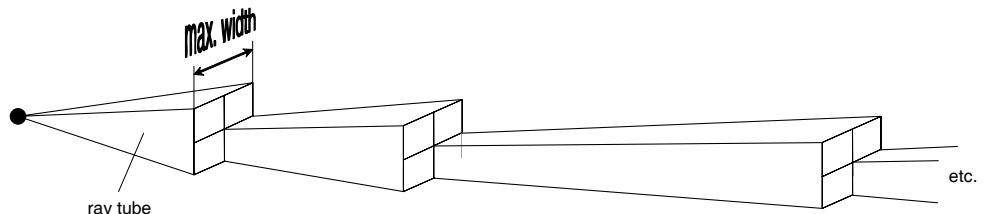


Fig. 5-9 Principle of ray splitting

#### 5.2.3.6 Miscellaneous

For a graphical presentation of the rays found, it is necessary to store the detected ray paths. If, however, a very large environment is investigated, the available RAM could not be enough to store all ray positions. Then it is appropriate to switch off this option.

Irrespective if the ray paths are stored or not, all information about the impulse responses at each receiver (spatial, temporal, and amplitude) is generally saved in the results.

A way to further save RAM consumption, is to restrict the number of the detected ray paths for each Tx/Rx pair to a certain upper limit. A recommended setting for this limit might be 3 to 5 rays.

### 5.3 Simulator Invocation

After all simulation settings have been properly defined, the simulation can be started. The basic sequence of actions is as follows:

- Choose the desired propagation algorithm.
- Start the simulation.
- Retrieve the results.

#### 5.3.1 Choosing a Propagation Algorithm

At first, one of the propagation algorithms must be chosen. It can be selected in the standard toolbar as shown in Fig. 5-10 or, alternatively, in the Simulation menu. Table 5-2 presents all available algorithms with a short explanation.

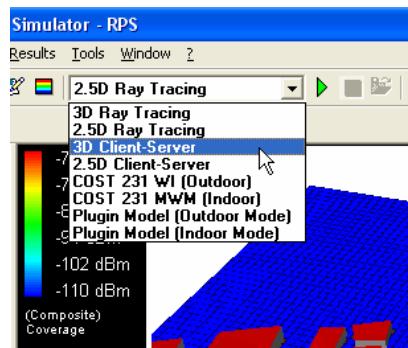


Fig. 5-10 Selecting a propagation algorithm from the standard toolbar

Table 5-2 Propagation algorithms

| Algorithm                               | Description   |
|---|---|
| <u>3D Ray Tracing</u>                   | A fully three-dimensional ray tracing algorithm for the local computer. Arbitrarily arranged polygons and diffraction edges can be considered. An unlimited number of reflections, penetrations, and diffractions can be detected in a ray until the given noise floor is reached. The algorithm is prepared for multi-threaded operation.  |
| <u>2.5D Ray Tracing</u>                 | A simplified three-dimensional ray tracing algorithm for the local computer. Only vertical obstacles and diffraction edges are considered; optionally, single reflections at non-vertical obstacles and diffracted rays over the roofs can be included. An unlimited number of reflections, penetrations, and diffractions can be detected in a ray until the given noise floor is reached. The algorithm is prepared for multi-threaded operation. |
| <u>3D Client-Server</u>                 | A fully three-dimensional ray tracing algorithm for a virtually unlimited number of server workstations in a TCP/IP LAN. Arbitrarily arranged polygons and diffraction edges can be considered. An unlimited number of reflections, penetrations, and diffractions can be detected in a ray until the given noise floor is reached. The algorithm is prepared for multi-threaded operation.   |
| <b>Only available in RPS Enterprise</b> |   |

| <i>Algorithm</i>                   | <i>Description</i>  |
|------------------------------------|---|
| <u>2.5D Client-Server</u>          | A simplified three-dimensional ray tracing algorithm for a virtually unlimited number of server workstations in a TCP/IP LAN. Only vertical obstacles and diffraction edges are considered; optionally, single reflections at non-vertical obstacles and diffracted rays over the roofs can be included. An unlimited number of reflections, penetrations, and diffractions can be detected in a ray until the given noise floor is reached. The algorithm is prepared for multi-threaded operation.<br><b>Only available in RPS Enterprise</b> |
| <u>COST 231 WI (Outdoor)</u>       | A semi-empiric propagation model according to the COST 231 Walfish-Ikegami outdoor model. Rays are propagated over the rooftops from each transmitter to each receiver.   |
| <u>COST 231 MWM (Indoor)</u>       | A semi-empiric propagation model according to the COST 231 Multi-Wall Model for indoor scenarios. Rays are propagated straight through the walls from each transmitter to each receiver.  |
| <u>Plugin Model (Outdoor Mode)</u> | A user-defined semi-empiric or empiric propagation model for outdoor scenarios. Rays are propagated over the rooftops from each transmitter to each receiver. It is only available if an appropriate plug-in has been selected.   |
| <u>Plugin Model (Indoor Mode)</u>  | A user-defined semi-empiric or empiric propagation model for indoor scenarios. Rays are propagated straight through the walls from each transmitter to each receiver. It is only available if an appropriate plug-in has been selected.   |

The selected propagation algorithm is persistently stored in the project. This means that when the project is re-opened at a later time, the propagation algorithm last used in that project is preselected in the standard toolbar.

### 5.3.2 Start of Simulation

It is generally recommended to have the Simulation -> Suggest Incremental Predictions flag checked. Then the user will be queried on every simulation start if he wants to exploit the incremental prediction mode where only obsolete transmitters are (re-)calculated. If this flag is not set, the complete radio channel will be predicted for all receivers irrespective of their obsolete status. Read more on the incremental prediction mode in Section 5.2.2.2.

The simulation is started by simply clicking the  icon in the standard toolbar or, alternatively, by choosing the Simulation -> Run menu entry. Immediately, the simulation will be started.

During the simulation, according statements in the message window inform about the progress of the calculations. In addition, some statistical information about the current status of the ongoing simulation can be obtained from the statistics window. It can be opened by choosing the Results -> Statistics menu entry. See Fig. 5-11 for an example.

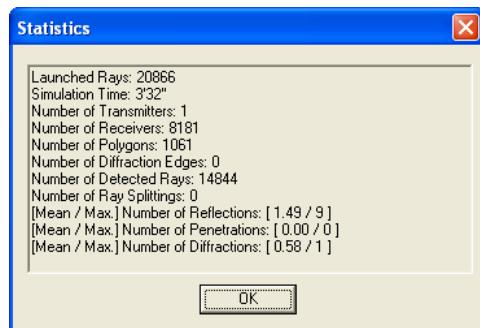


Fig. 5-11 Exemplary statistics window

### 5.3.3 Retrieving Results

It is possible to retrieve intermediate results during an ongoing coverage simulation. This can be done by clicking the icon in the standard toolbar or by choosing the Simulation -> Retrieve Intermediate Results menu entry. These results can then be analyzed as described in Chapter 6. At the end of a simulation, all the results are automatically loaded into the GUI in order to be analyzed.

### 5.3.4 Stopping a Simulation

A running simulation can be stopped at any time by clicking the icon or by choosing the Simulation -> Stop menu entry. All results that have been calculated so far are loaded into the GUI immediately. Transmitters where that ray launching was not yet finished are automatically marked as obsolete. In a new simulation they would be completely recalculated.

## 5.4 Parallelized Ray Tracing

RPS Enterprise provides parallelized ray tracing algorithms that extend RPS to a distributed radio channel prediction system using a TCP/IP network. In practice, this can be a LAN or an internet connection. The RPS application itself works as a client then, and RPS Servers running on several workstations under Windows or Linux (kernel 2.x or higher) perform the calculations.

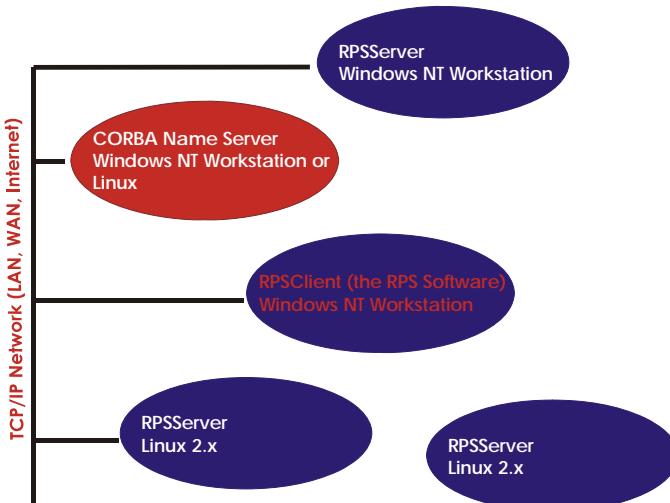


Fig. 5-12 Communication between the RPS client and several RPS servers

RPS Client and RPS Servers communicate via the CORBA protocol. RPS uses the OmniORB package (available at <http://omniorb.sourceforge.net/>). It is an Object Request Broker (ORB) that implements the 2.1 specification of the Common Object Request Broker Architecture (CORBA), and – it is provided under the GNU GPL. The complete omniORB packages are available on the RPS CD-ROM including the full sources and documentation. This implementation can handle multiple threads and is faster than other packages available today.

### 5.4.1 Simulation Setup

The configuration of the environment, the network, and the simulation parameters is done in the same way as described before. For a client-server simulation then the following things must be done:

- The name server must be started. This is important for the communication between the client and the servers.
- The RPS Servers must be started on the corresponding workstations in the TCP/IP network.
- Servers must be selected for a simulation. It is possible to run several simulations from different clients simultaneously that use disjoint servers out of a server pool.

Then a simulation can be started in the usual way. Make sure that one the client-server ray tracing algorithms was selected in the Simulation menu; otherwise a local simulation would be started. Also the retrieving of intermediate results and a preterm stop of a simulation can be used exactly in the same way as for a local simulation. Status messages of the communication between the client and the servers are shown in the message window in addition to the normal progress messages.

The necessary steps for performing a highly parallelized simulation in the client-server mode is described in detail in the following Subsections.

#### 5.4.1.1 Name Server Setup

At first, the name server `omnинames` must be started on the designated host workstation (usually the workstation where the RPS Client is running). This host must be the same workstation that was configured in the server batch file as described in Section 1.8.5. The `omnинames` application can be found in the  
`%ApplicationPath%\CORBA\OmniNamesWIN32` directory. For the convenience of the user, two batch files for Windows systems are provided in this directory, too.

In general, it is a difference whether the name server is started for the first time or not. With a first time start, the name server creates some log files in a temporary directory that contain information about e.g. the available servers and the port number used. If the name server would be restarted at a later time, it could use the information from those log files to re-establish the connections. Only if the port number or the host were changed, the log files have to be created anew, i.e. the name server has to be started as if it would be for the first time.

The provided batch files can be used to start the name server with the default port 2809. Of course, they could also be modified for another port as well. The usage of the batch files is as follows:

- If the name server shall be started for the first time, the batch file `_firststart.bat` may be used. It contains two commands as in the following example. First it deletes the log files in the temporary directory, and then the name server is started.

```
del c:\temp\omnianames-%COMPUTERNAME%.*  
omnianames -start
```

- If you would like to use another temporary directory or another port number than the default port 2809, you can modify this batch file accordingly. As you can see, the host name is always included in the temporary directory name. Thus several log file configurations can be maintained for different hosts.
- If the name server shall be started at a later time with the same configuration as before, the batch file `_start.bat` may be used. It just contains a single line that starts the name server. The configuration data is retrieved from the log files in the temporary directory.

```
omnianames.exe
```

You can also start the name server manually. There exist versions for many operating systems. `omnianames` is a command line tool. Under Windows NT systems, `omnianames` is started for the first time by executing the following command:

```
omnianames -start <Port>
```

`Port` is a free port of the workstation, where `omnianames` will listen to requests. `omnianames` then creates a log file in the `temp` directory. If `omnianames` is to be started again and the log file was not removed, the command `omnianames` is sufficient. In this case, the name server reads its configuration from the last log file.

More detailed setup instructions for Windows NT and other operation systems can be found in the documentation of OmniORB (`omniORB.pdf` or HTML-documentation).

#### 5.4.1.2 Starting RPS Servers

On each designated server workstation an RPS Server should be started now. It is not recommended to start more than one RPS Server on a workstation. On the other side, it is even possible to start RPS Servers on normal terminal workstations because the servers run with lower priority. Thus they can run in the background and do not interfere with the user actions on that workstation.

The RPS Servers are independent of a running name service. However, if an RPS Server is started when no name service is available yet, it simply waits until the name server is up. Since the name server and the RPS Servers do not depend on each other and, thus, can be started independently, this starting procedure could also be automated in a startup script. Even if the workstations do not always come up in the same order, this procedure will work safely.

For the convenience of the user, also for the server start an according batch file is provided in `%ApplicationPath%\WIN32\startrpsserver.bat`. For Linux systems the comparable shell script can be found in `%ApplicationPath%\LINUX2\startrpsserver.sh`. These files must be edited before concerning the host name and the port number used as described in Section 1.8.5. Then the batch file needs just to be started on all the server workstations.

RPS Servers could also be started manually by the following command:

```
rpsserver <NAME> -ORBInitialHost <IP-Address> -ORBInitialPort <Port>
```

The RPS server contacts the name server on the machine with the given IP-Address on the given Port and it gets connected under the Name NAME. OmniORB is able to resolve host names to addresses, so that also the host name of the machine where the name server is running can be used. The NAME is a freely configurable string (for instance fox or RPSServer1...). However, this name must be a unique string. The client may crash if different servers use identical names!

As soon as the RPS Server has been started on a workstation, a console window is opened. After a couple of seconds, the server should respond with the notification that it is initialized and ready. Later, when a simulation is running, some status messages about the progress of the simulation are shown in this window. To stop a server, simply type any key in the console window.

#### 5.4.1.3 Selecting RPS Servers

Next, the CORBA connections from the client computer to the servers must be configured. This is basically done in three steps:

- A new client connection to the name server must be initially set up. This is only necessary, if the name server is newly started or the host computer of the name server was changed.
- The available RPS Servers must be detected from the RPS Client side.
- The user can select an arbitrary collection of the available servers for a distributed simulation.

All these tasks can be done very conveniently in the Client-Server tab of the preferences dialog. A sample view of this dialog can be seen in Fig. 5-13.

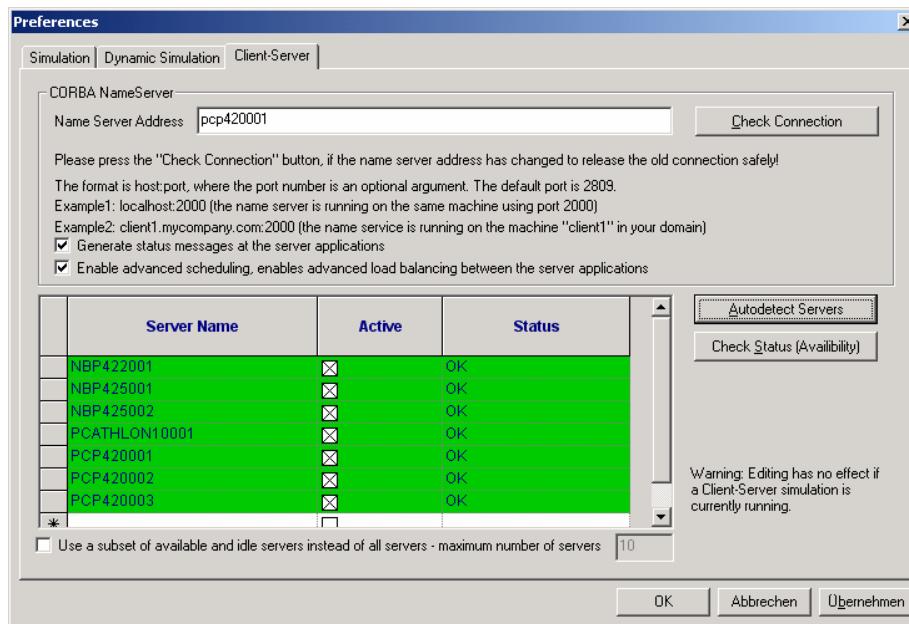


Fig. 5-13 Client-Server setup tab in the preferences dialog

In the upper text field Name Server Address the host name or IP address must be inserted. If a different port for the omninames name server than the default one (2809) is used, it

must be specified behind the host name delimited by a colon. If the host name was changed recently or the name server has just been started, the right button Check Connection must be pressed. This will configure the new connection and release a possible old one safely.

Next, the available RPS Servers in the TCP/IP network must be detected. This is accomplished by pressing the button Autodetect Servers. Then the workstations in the network are scanned for running RPS Server applications. After some seconds the complete list appears in the table to the left. The servers can have different states that are listed in Table 5-3.

Table 5-3 RPS Server states

| Status | Background Color | Description  |
|--------|------------------|--|
| OK     | green            | The server is idle and ready for simulation.   |
| busy   | yellow           | The server is currently busy. This means that it is used by another client for a simulation.   |
| Failed | red              | The server is not available any more. Possible reasons for this case can be that the server was stopped or that the connection of the corresponding workstation is broken. |

Automatically, all idle servers are checked as active for the current simulation. Nevertheless, it is in the user's choice which and how many servers to be used for a distributed simulation. Servers can simply be activated/disabled by toggling their active flag in the list using the left mouse button.

The maximal number of active servers is not restricted. The number of servers to be activated in practice depends on the LAN performance and - of course - on the complexity of the simulation. RPS also provides a convenient configuration to share a pool of available servers among several clients. This option can be chosen by checking the Use a subset of servers which is described in Subsection 5.4.1.4 below in greater detail.

Once the available servers have been detected, it is not necessary to do this step again for further simulations, assumed the network connections were not changed. Then it is sufficient to query the status of the distinct servers by pressing the button Check Status (Availability). This will update the presentation of the server status in the list for all servers contained in it.

If the check box Generate status messages at the server applications is active, the servers write some information during the simulation process to their console.

The option Enable advanced scheduling activates functions, that allow an advanced load balancing of all active servers. This option can be especially useful in LAN networks where the workstations have different computational power.

#### 5.4.1.4 Using a Server Pool

Up to here it was always assumed that a coverage simulation is distributed over several servers controlled by a single client. If several clients perform distributed simulations in the same TCP/IP network, it seems to be useful to share the pool of available servers among them. Especially in combination with the COM remote control interface of RPS this can be a very efficient way to predict the radio channel initiated from a Radio Network Planning (RNP) tool. Fig. 5-14 demonstrates this case by an example.

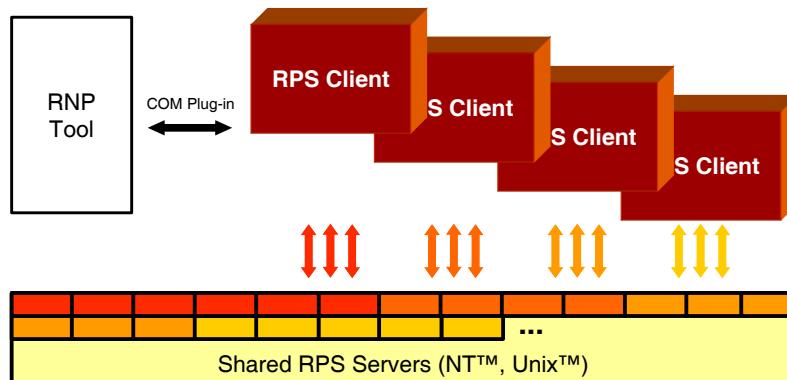


Fig. 5-14 Sharing a pool of RPS Servers

This principle can be used by checking the option Use a subset of servers in the Client-Server tab of the preferences dialog. In the corresponding text field a certain number of servers can be configured that will be activated for the current simulation, starting from the top of the server list. Busy or unavailable servers are ignored for the selection.

#### 5.4.2 Performing Parallelized Simulations

After the necessary CORBA connections were set up and an appropriate number of RPS Servers was selected, a parallelized ray tracing simulation can be started. The handling of the simulation process is basically the same as for local simulations.

##### 5.4.2.1 Simulator Invocation

It is generally recommended to have the Simulation -> Suggest Incremental Predictions flag checked. Then the user will be queried on every simulation start if he wants to exploit the incremental prediction mode where only obsolete transmitters are (re-)calculated. If this flag is not set, the complete radio channel will be predicted for all receivers irrespective of their obsolete status. Read more on the incremental prediction mode in Section 5.2.2.2.

For a parallelized radio channel simulation, the user has the choice between a 2.5D and a 3D ray tracing algorithm. The proper algorithm can be chosen by selecting one of the algorithms 2.5D Client-Server or 3D Client-Server from the standard toolbar or, alternatively, one of the menu entries Simulation -> 2.5D Client-Server or Simulation -> 3D Client-Server, respectively. The client-server simulation is started by simply clicking the icon in the standard toolbar or, alternatively, by choosing the Simulation -> Run menu entry. Immediately, the simulation will be started.

The server applications run in the background with lower priority. Thus they do not interfere with possible user interactions on the server workstations.

During the simulation, according statements in the message window inform about the progress of the calculations. In the beginning of a simulation, the environment data is submitted to each RPS Server. Then, the rays to be launched are distributed. Finally, the simulation results and detected rays (if option is active) are retrieved. Results can also be retrieved intermediately, just in the way as known from local simulations described in Section 5.3.3.

If a communication error occurs, a message is generated by RPS. When an RPS Server simulates, other clients get a busy message. A client has the capability to restart a busy

RPS Server when it wants to access them. Furthermore, a running server with a broken connection is timed out after 180s – then the simulation is stopped (but not the server itself). Then this server is idle again and can be used for another simulation.

The simulation can be cancelled at any time using the icon in the standard toolbar. The results calculated so far are retrieved then. Transmitters where the ray launching was not yet finished completely, are set obsolete. In another simulation they would be completely re-calculated.

#### 5.4.2.2 Load Balancing

Especially towards the end of a parallelized simulation the computation speed can be negatively affected because all servers have to wait for the slowest one. There are basically two reasons where a distributed simulation can become inefficient in the described manner:

- The server workstations have different computational power. Slower workstations take more time for the same number of operations.
- The ray launching approach could produce very unbalanced load on different servers. This depends heavily on the environment of the simulation.

In order to compensate for this effect, the RPS Client implements a load balancing function that distributes the necessary operations in such a way that faster servers get more work than slower servers (or more busy machines). This feature can be activated by checking the option Enable advanced scheduling. Fig. 5-15 shows an example of how advanced scheduling can improve the server load and thus reduce the total simulation time.

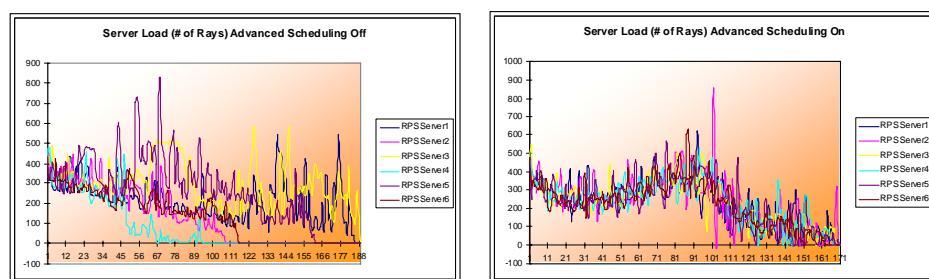


Fig. 5-15 Sample plot of the server load without (left) and with advanced scheduling (right)

The compute time for a parallelized simulation in a TCP/IP network can be considerably reduced by the usage of advanced load balancing between the servers.

The load balancing can be pre-configured in the `rps.ini` file by several parameters in the `[RPS]` section as described in Section 1.8.4.1. It is recommended to leave these settings as they are configured in the original RPS installation.

#### Important Notice

If the CORBA message size (parameter `giopMaxMsgSize` in the `rps.ini` file) for the Client-Server communication shall be adjusted, the user must make sure that the same value is also configured in the `startrpsserver.bat` file in the form

```
... -ORBgiopMaxMsgSize <value>
```

Otherwise the communication between the Client and the Servers would fail.

## 5.5 Propagation Model Plug-ins

Besides the built-in propagation models it is possible to apply user-defined plug-ins for empirical and semi-empirical propagation models. RPS provides a plug-in interface that can be easily used to implement and integrate custom propagation models written in Visual C++ or as MATLAB m-file. This Section describes how to use custom propagation models and how to implement your own semi-empirical models. The basics of the RPS plug-in technology are presented in Section 3.8.

### 5.5.1 Using Custom Propagation Models

Custom (empirical) propagation model plug-ins can be used for site-specific path loss calculations based on

- the base station (transmitter) position and elevation,
- the mobile station (receiver) position and elevation,
- the distance between base station and mobile station,
- the carrier frequency,
- the number of buildings or walls located between base station and mobile station, where each building/wall can contribute a different additional path loss depending on its size, thickness, material and incidence angle.

RPS provides plug-ins with source code for the following empirical and semi-empirical prediction models:

- Basic Multi Wall Model (m-file)
- COST 231 Multi Wall Model (DLL and m-file)
- COST 231 Walfish-Ikegami Model (DLL and m-file)
- Simple Path Loss Model / Free Space Path Loss Model (DLL and m-file)
- UMTS 30.03 V3.2 Vehicular Model (DLL and m-file)

All available models are listed in the Configuration tab of the tree window below the Plugins – Propagation Models node. An example view is given in Fig. 5-16.

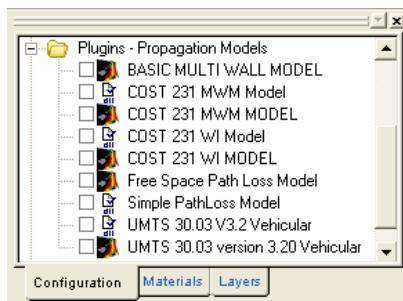


Fig. 5-16 Overview of custom propagation models in the tree window

Especially the Basic Multi Wall Model and the Simple / Free Space Path Loss models should demonstrate the easy implementation of user-specific empirical prediction models. The COST models are one of the most accurate empirical models (indoor and outdoor, respectively), under consideration of the underlying building database in detail (called semi-empirical in practice).

The UMTS model demonstrates the implementation of a specific prediction model, standardized in the UMTS 30.03 recommendation V3.2. Any other empirical and semi-empirical propagation formula can be used in propagation model plug-ins similar to the given examples.

Before a prediction can be performed, a custom propagation model plug-in must be selected in the Configuration tab of the tree window or in the plug-in dialog. Furthermore, one of the options Plugin Model (Outdoor Mode) or Plugin Model (Indoor Mode) must be selected either from the standard toolbar or from the Simulation menu in order to define the type of propagation model.

The basic difference between empirical *indoor* and *outdoor* simulations is the assumed ray path between transmitter and receiver. While in an indoor empirical model the ray propagates on a straight path directly from the transmitter to the receiver through all intermediate walls, the ray path in an outdoor empirical model is diffracted over the roofs of all buildings in the direct way from the transmitter to the receiver. See Section 5.1.2 for more details on the characteristics of these models.

Then, a simulation can be started by simply clicking the icon from the standard toolbar or by choosing the Simulation -> Run menu entry. During the simulation, for each transmitter-receiver combination the plug-in (DLL or m-file) is called from RPS. The simulation progress is indicated by some status messages in the message window.

Incremental prediction mode is also available using custom empirical models. Please refer to Section 5.2.2 for details.

### 5.5.2 Developing User-Defined Empirical Propagation Model Plug-ins

This Section explains how user-defined plug-ins can be developed using the RPS DLL interface or the use of m-files for empirical propagation models by writing only some lines of code. DLLs must be written and compiled in the MS Visual C++ 6 environment. Templates are available on the RPS CD-ROM. M-files are MATLAB scripts that can be generated with any text editor. The use of the built-in RPS Script Editor or the MATLAB Script Editor is recommended.

An empirical propagation model determines path loss values for any transmitter-to-receiver link. Several parameters, derived from the environment database, can be considered in the path loss formula.

As an input to the propagation model plug-in, RPS determines all obstacles located between the transmitter and the receiver in 2D case (an intersection point exists in the x-y plane). For each obstacle, the distance from the transmitter, its dimension in z-direction (i.e. its height) and thickness, its material characteristics, and a pre-calculated attenuation value is available. These input data are schematically shown in Fig. 5-17.

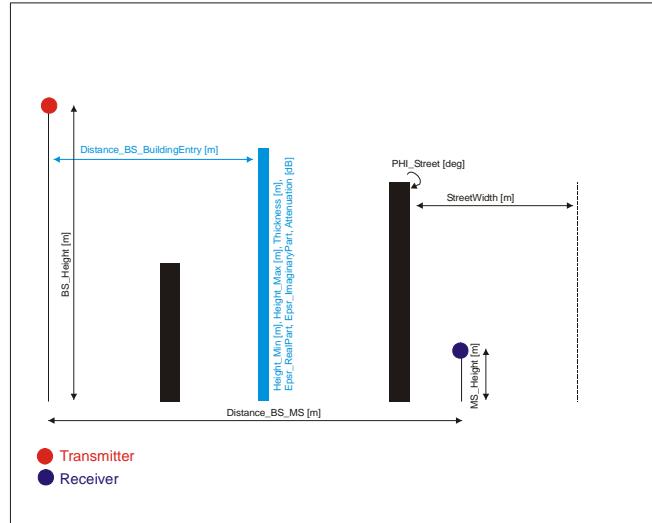


Fig. 5-17 Available input data for user-defined path loss predictions

In the following Table 5-4 the available parameters are summarized.

Table 5-4 Available parameters for custom propagation model plug-ins

| Parameter  | Description  |
|--|--|
| <b>General Parameters</b>  |  |
| Distance_BS_MS   | Distance between base station and mobile station [m]   |
| Frequency  | Carrier Frequency [GHz]  |
| BS_Height  | Height of the base station (z-coordinate) [m]  |
| MS_Height  | Height of the mobile station (z-coordinate) [m]  |
| StreetWidth  | Double distance from the last obstacle to the mobile station, i.e. it is assumed that the MS is always situated in the middle of the street.                   |
| PHI_Street   | Incidence angle PHI with respect to the orientation of the last obstacle in x-y plane [0...90 deg], an angle of 90 degrees is equal to perpendicular incidence |
| <b>Parameters for each wall / building entry (i.e. obstacle)</b> |  |
| Distance_BS_BuildingEntry  | Distance between the (2D) intersection point at the building and the base station position [m]   |
| Height_Min   | Lower z-coordinate of the building [m]   |
| Height_Max   | Upper z-coordinate of the building [m]   |
| Thickness  | Thickness of the building entry [m]  |
| Epsr_RealPart  | Real part of the material's relative permittivity  |
| Epsr_ImaginaryPart   | Imaginary part of the material's relative permittivity   |

| Parameter   | Description  |
|-------------|--|
| Attenuation | Penetration loss (Attenuation) [dB] of an electromagnetic wave; frequency, thickness, incidence angle, polarization state, and material characteristics are taken into account |

The interaction between the propagation plug-in and RPS works as follows:

1. RPS fills the parameter list for a certain transmitter-receiver pair.
2. RPS calls a custom function in the plug-in (a DLL or a MATLAB script).
3. RPS retrieves the path loss value for the transmitter-receiver pair.

#### 5.5.2.1 DLL Plug-in SDK for Custom Propagation Models

This type of plug-in is realized as a C++ class, called `CPlugin_RPS_PM`. It contains the parameter list and “Get/Set” functions for accessing them. In Fig. 5-18 the methods and attributes of the classes are summarized.

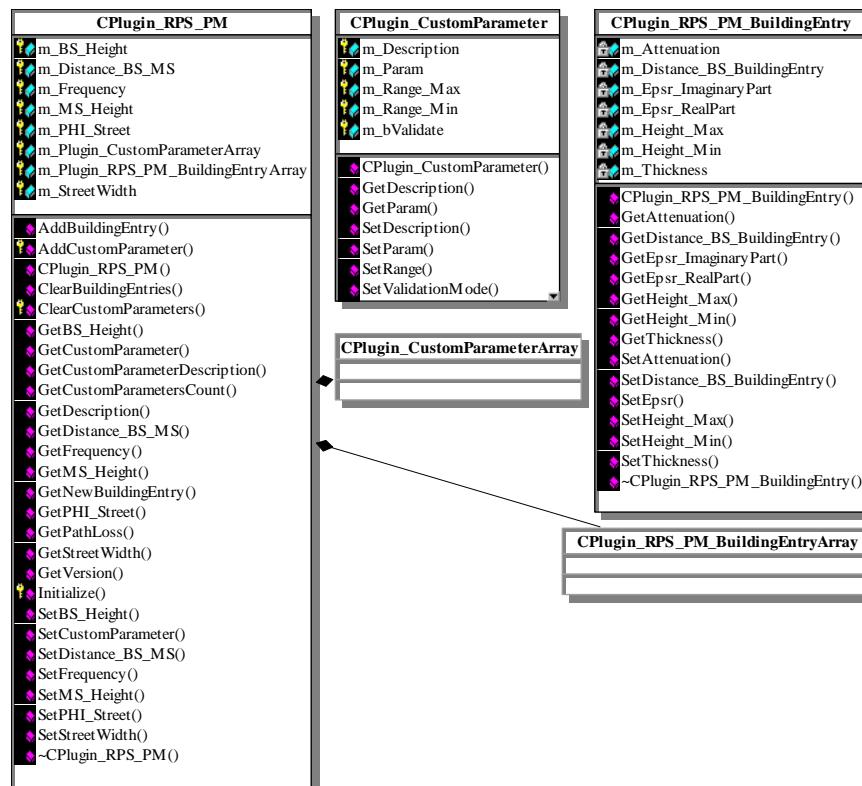


Fig. 5-18 Class view of the template for propagation plug-ins

In order to write a user-defined plug-in, the following three functions must be customized. The declarations of these functions can be found in the `plugin_customfunctions.cpp` file.

##### `CPlugin_RPS_PM::Initialize()`

This function is called by RPS when the plug-in is attached to the framework for the first time. Here, some initialization code can be added. In particular, custom

parameters can be added here. A custom parameter can be modified directly in the RPS Plug-in Dialog. A custom parameter contains a value and a description. Range limits can be applied optionally. For instance, a custom parameter for a path loss exponent with range limits of one and eight can be added by the following line of code:

```
AddCustomParameter( "2.3", "Path Loss Exponent", 1, 8, true);
```

The first argument defines the value or property of the parameter. It is stored commonly as a character string, since a string can be converted practically to any data type. The parameter value or property can be modified directly in the RPS Plug-in Dialog. The second argument is used for a brief description of the parameter. The last three fields are optional. The third and fourth arguments can be used to define range limits for the parameter. The last argument activates or deactivates the consideration of the range limits defined before.

```
char* CPlugin_RPS_PM::GetDescription()
```

This function must return a valid string that describes the plug-in briefly. This description is used for instance in the RPS Plug-in Dialog. It is independent of the DLL file name.

Example for an implementation:

```
char* CPlugin_RPS_PM::GetDescription()
{
    return "Simple PathLoss Model";
}

double CPlugin_RPS_PM::GetPathLoss()
```

This function is called for each base station – mobile station link. The parameter list is already updated by RPS before this function is called. The return value must contain a valid path loss value. The following example shows the implementation of a simple path loss equation:

```
double PL;
// convert the custom parameter with index 0 to a double value
double PathLossExponent = atof(GetCustomParameter(0));
// The simple path loss equation
PL = 32.5 + 10.0*PathLossExponent*log10(m_Distance_BS_MS) +
     20.0*log10(m_Frequency);
// Return a Path Loss Value in dB
return PL;
```

The source code for more empirical or semi-empirical indoor and outdoor models is available on the RPS CD-ROM in the SDK directory.

#### 5.5.2.2 MATLAB Plug-in SDK for Custom Propagation Models

A custom propagation model can be implemented as MATLAB script (m-file) as well. The input parameter list is stored in the MATLAB structure array **RPS\_Input**. It contains the following fields:

```
Frequency: [1x1 double]
Distance_BS_MS: [1x1 double]
BS_Height: [1x1 double]
MS_Height: [1x1 double]
```

```

StreetWidth: [1x1 double]
PHI_Street: [1x1 double]
Distance_BS_BuildingEntry: [Nx1 double]
Height_Min: [Nx1 double]
Height_Max: [Nx1 double]
Thickness: [Nx1 double]
Epsr: [Nx1 double]
Attenuation: [Nx1 double]

```

N is the number of obstacles, located between base station and receiver station. For each obstacle (or building entry), the parameters described above are stored into the row submatrices.

The path loss value must be stored into the **RetValue** field of the **RPS\_Output** MATLAB structure array. To give an example, the following code realizes a basic multi wall model:

```

% -----
% BASIC MULTI WALL MODEL - Propagation Model Plugin
%
%
% -----
%
% The basic multi wall model demonstrates a method, where the additional
% loss due to wall penetration is derived from the wall's thickness:
%
% PL = 32.5+20*log10(f)+10*PL_Exp*log10(d)+sum(Wall_Thickness)*W_Loss
% f: Frequency
% PL_Exp: Path Loss Exponent (2 in Free Space)
% d: Distance of BS and MS
% Wall_Thickness: Thickness of each Wall between BS and MS
% W_Loss: Wall Attenuation per Meter in dB
%
% -----
%
% 31.01.2001 Johannes Hübner, Radioplan GmbH Dresden, Germany
% -----
%
Wall_Attenuation_per_Meter = 20; %Loss in dB

PL_Exp = 2;
%LOS Attenuation (Free Space Loss)
PathLoss = 32.5 + 20.0*log10(RPS_Input.Frequency) +
PL_Exp*10.0*log10(RPS_Input.Distance_BS_MS);

% Basically add [Wall_Attenuation_per_Meter]dB/m additional attenuation
PathLoss = PathLoss + sum(Wall_Attenuation_per_Meter*RPS_Input.Thickness);

% Fill out RPSOutputParam Parameter 'PathLoss'
RPS_Output.PathLoss = PathLoss;

```

The header of a MATLAB script file is parsed by RPS, when the file is attached by the framework. This header information can contain the items as given in Table 5-5.

Table 5-5 Header Information of MATLAB scripts

| <i>Keyword</i> | <i>Description</i>                                 |
|----------------|--|
| TYPE           | type of plug-in                                    |
| VERSION        | version number, used for future version management |

| <i>Keyword</i>        | <i>Description</i>   |
|-----------------------|--|
| REVISION              | revision number, can be modified by the user   |
| DESCRIPTION           | short description of the plug-in   |
| MATRIXSUPPORT         | 0 – matrix mode is not supported by the plug-in<br>1 – matrix mode is supported by the plug-in |
| ENFORCEFULLMATRIXMODE | 0 – no full matrix mode<br>1 – enforce full matrix mode  |

TYPE specifies the type of plug-in. Two types are supported by RPS:

- PLUGIN\_RPS\_PM for custom propagation models
- PLUGIN\_RPS\_SP for custom surface plots.

The version number is used internally to distinguish future versions of the same plug-in type. The revision number can be used for internal reference. The DESCRIPTION keyword is followed by a short string that describes the plug-in briefly.

The field MATRIXSUPPORT tells RPS, if the plug-in supports matrix mode processing. Matrix mode is only relevant for custom surface plot plug-ins. In addition, the keyword ENFORCEFULLMATRIXMODE enforces the parallel processing of all output values in one step.

## 5.6 Dynamic Simulations

RPS offers a powerful set of functions for the generation of dynamic channel data in addition to pre-calculated channel data. The idea behind that is the combination of the wave propagation model with a mobility model. During a simulation a receiver moves through the environment with a fixed velocity along straight routes between different - so called - “attraction areas”. The dynamically generated channel impulse responses include angular information and can be stored then to ASCII- or MATLAB-readable files.

### 5.6.1 Mobility Model

The random walk mobility model allows completely random movements throughout the entire environment. It is modeled as a Markov chain. A mobile user moves randomly from one attraction area to another. The transitions between the attraction areas are defined by movement corridors and transition probabilities  $p(aa)$  and  $p(d)$ , see Fig. 5-19. RPS does not distinguish different types of transition probabilities.

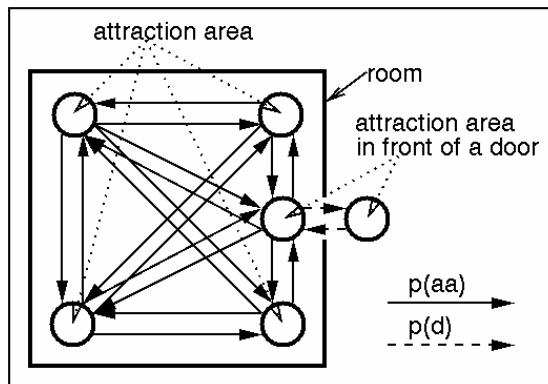


Fig. 5-19 Random walk mobility model as Markov chain

In order to create a mobility model, attraction areas and transitions between them must be defined. This can be done conveniently in the 2D view by using the following two icons from the paint toolbar as given in Table 5-6.

Table 5-6 Editing modes for the movement model

| Icon | Description  |
|------|--|
|      | Modify attraction areas                                  |
|      | Modify transition probabilities between attraction areas |

Attraction areas are defined by their center position  $p(x, y, z)$  and a radius  $r$ . When the drawing mode Edit attraction areas is activated, an attraction area can be added by a simple click using the left mouse button at the desired position within the environment. In the following dialog like Fig. 5-20, the position and the radius of the new (or selected) area can be modified or removed. Additionally, all existing attraction areas could also be removed.

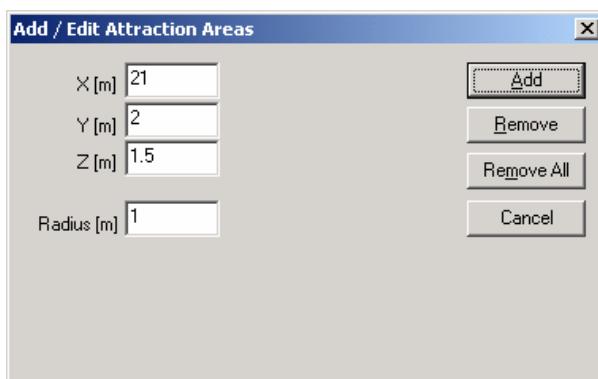


Fig. 5-20 Add / edit position and size of an attraction area

The transition probabilities between two attraction areas can easily be defined by simply dragging a connection between them using the left mouse button. In order to disconnect two areas, the mouse has to be moved from the originating area to the destination area again. It is also possible to connect two attraction areas more than once to increase their transition probability compared to other relations.

The moving corridors are defined by the transitions between attraction areas and the radii of these areas.

If an attraction area is selected by holding down the left mouse button, the connections to other areas are shown by thin blue lines.

An example of an environment with numerous attraction areas and movement corridors between them is given in Fig. 5-21.

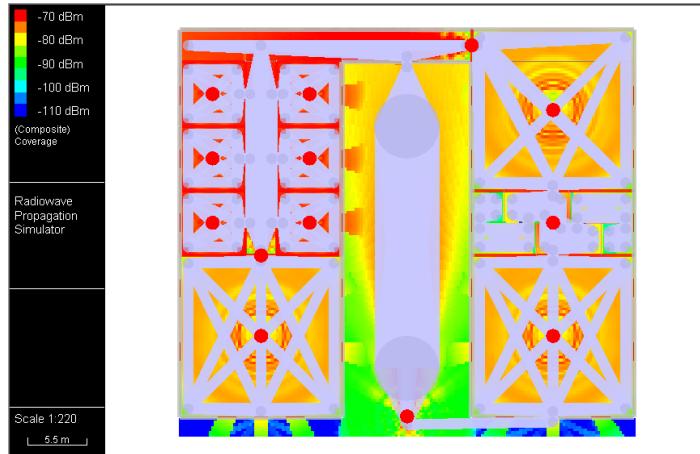


Fig. 5-21 Attraction areas and movement corridors defined in an office environment

### 5.6.2 Configuring a Dynamic Simulation

During a dynamic simulation, RPS moves a mobile receiver through the environment along the given movement corridors and with a configurable velocity  $v$ , where at discrete time instants  $\Delta t$  the channel data is stored. Please note that the simulation is not performed in real time!

Parameters for dynamic simulations can be defined in the [Dynamic Simulation](#) tab of the preferences dialog, shown in Fig. 5-22. The parameters are described in Table 5-7.

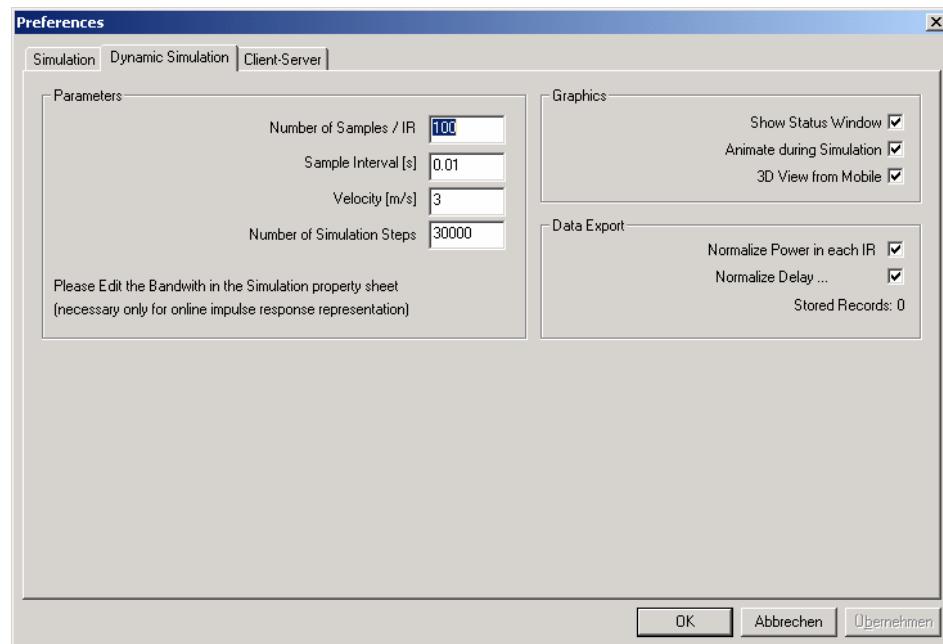


Fig. 5-22 Dynamic simulation configuration dialog

Table 5-7 Parameters for a dynamic simulation

| Parameter                  | Description   | Unit |
|----------------------------|---|------|
| <b>Parameters</b>          |   |      |
| Number of Samples / IR     | The number of samples of each impulse response. The relative temporal distance between two samples (i.e. temporal resolution) is given by the inverse of the bandwidth $B$ . The maximum excess delay results from the product $\text{Number\_of\_Samples} \cdot B$ .<br>This option is used only when the data is saved to disk! | —    |
| Sample Interval            | The interval $\Delta t$ in seconds, where channel parameters are determined and saved into memory.  | s    |
| Velocity                   | The velocity $v$ of the mobile receiver. Consequently, the spatial distance between two positions where impulse responses are obtained is $\text{Sample\_Interval} \cdot Velocity$ .  | m/s  |
| Number of Simulation Steps | The maximum number of impulse responses to be saved into memory.  | —    |
| <b>Graphics</b>            |   |      |
| Show Status Window         | If this flag is set, windows are displayed showing the angular and temporal impulse response at the current position of the mobile station during the dynamic simulation.   | —    |
| Animate during Simulation  | If this flag is set, the moving mobile station is shown in 2D or 3D view.   | —    |
| 3D View from Mobile        | If this flag is set and 3D view is used, the dynamic simulation is shown from the mobile station's perspective.   | —    |
| <b>Data Export</b>         |   |      |
| Normalize Power in each IR | If this flag is set, the power of each impulse response is normalized such that its square sum is equal to 1.<br><b>only used for storing dynamic simulation results</b>  | —    |

| Parameter       | Description  | Unit |
|-----------------|--|------|
| Normalize Delay | If this flag is set, the delay of the first path is normalized to 0s. All other paths are reduced in delay accordingly.<br><b>only used for storing dynamic simulation results</b> | —    |

During the simulation process it is useful to select a 2D or a 3D view to obtain animated pictures (if the option Animate during Simulation was activated). If the option Show Status Window was activated, the dynamic impulse responses and angles of arrival are presented in windows during the simulation.

If the box 3D View from Mobile was activated, and a 3D view is active, the environment around the mobile can be observed from the mobile's perspective.

### 5.6.3 Performing a Dynamic Simulation

A dynamic simulation is only useful if radio channel data have already been calculated previously because these channel data are used in the dynamic simulation. Likewise, it is advisable to ensure that the movement corridors are located in areas that are completely covered by receivers in order to have channel data available everywhere.

The simulation is started by using the menu command Simulation -> Run Dynamic Simulation. It can be finished by pressing the <ESC> key during the simulation process. If the option Animate during Simulation is active, the movement of the mobile station is shown in the 2D or 3D view (depending on which view was active at the beginning of the dynamic simulation).

A simulation without any attraction area is not possible. The initial receiver position is always the attraction area with index of zero. Starting at this position, a destination area is randomly chosen using the transition probabilities. Within the radius of the selected destination area a random position is calculated then that becomes the destination of the current motion. If that point is reached, a new attraction area is randomly chosen according to the transition probabilities towards other attraction areas, and the next destination position is calculated. Using this mobility model, the mobile receiver is moved only within the given corridor defined by straight lines connecting the radii of the source and the destination attraction area.

For a dynamic simulation, the radio channel data must have been predicted before. A dynamic simulation is finished when either the <ESC> key is pressed or the maximal number of simulation steps is reached.

### 5.6.4 Storing Dynamic Channel Data

RPS contains a comfortable MATLAB interface that allows to store dynamic channel data in a similar way as static channel data. As in the export functions for statically generated channel data, the matrices can be stored into separate ASCII files or into one MAT-File. The main difference to the static functions is that the dynamic channel data is modified to yield "more realistic" impulse responses - a process described more detailed in the following.

Please, note that dynamically generated data is not stored into the RPS files!

RPS determines the best serving transmitter for each receiver position. Signals from this transmitter station are considered by storing the channel data for this transmitter position. The corresponding transmitter ID (TxID) is stored into a separate matrix. For positions where no signals are detected, the TxID values are set to -1.

The impulse responses can be normalized both in magnitude and delay. If the option Normalize Power in each IR is active, the complex magnitudes are normalized such that the square sum of all samples yields 1 (no power loss). This option can be useful if the data shall be used in link level simulation tools where the channel is typically modeled as a FIR filter structure. If the option Normalize Delay is switched on, the delays of multi-path components are stored relative to the signal with the shortest delay (for instance a line of sight (LOS) path under LOS conditions). These normalization procedures are performed when the matrix files are saved to disk. They have no influence on the raw dynamic simulation data themselves stored into memory.

The azimuth angles at the receiver are recalculated such that they are relative to the signal with the highest complex magnitude. At the active transmitter, the angles of outgoing paths are normalized to the reference angle  $\varphi_{ref}$ . The reference angle is part of the antenna diagram and can be adjusted in the transmitter properties.

## 6 Working with RPS — Result Analysis

This Chapter covers the explanation of the result analysis opportunities offered by RPS. Starting from the raw channel impulse responses collected at each receiver, a lot of secondary results can be derived. Such results can be presented and analyzed in surface plots, chart graphs, and tables. Moreover, the analysis capabilities of RPS can be greatly enhanced by implementing user-defined surface plot plug-ins that pave the way to easily implement radio network planning functionality for result analysis.

Results can be further analyzed for a given position (point analysis), along a specified path (path analysis), or for a selected transmitter (Tx analysis). For these types of analysis functions, different presentation opportunities are provided as, e.g., tables, chart graphs, or histograms. Additionally, supplementary post-processing procedures can be implemented for them as MATLAB stand-alone plug-ins.

The Chapter is concluded by a description of all data import/export functionality that make RPS easily integrable into common network planning workflows.

### 6.1 Simulation Statistics

The statistics dialog is accessible from the menu entry Results -> Statistics. During a simulation, statistical measures about typical ray tracing parameters are collected in this dialog. When the simulation is finished, it contains a complete overview of all relevant simulation statistics. Fig. 6-1 shows an example of this dialog.

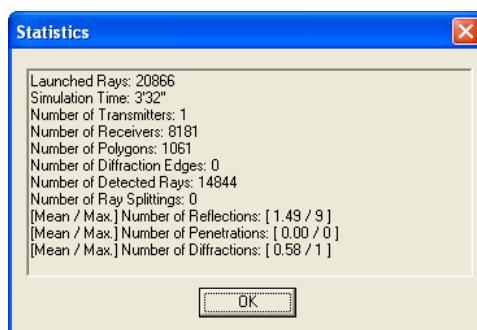


Fig. 6-1 Simulation statistics dialog

Note that the simulation statistics only embody additional information on the simulation process. If they are used for comparison purposes between different simulations or algorithms, they have to be seen always in combination with the simulation settings since e.g. the noise floor has a dramatic influence on the required simulation time.

## **6.2 Concept of Result Analysis**

When a radio channel prediction is finished, at each receiver position the obtained channel impulse response (including angles of arrival information) is available. Optionally, the complete information of all ray paths (including the intermediate deviating points) could also have been recorded. These results are the basis for all analysis and post-processing functions provided by RPS. Fig. 6-2 gives an overview of the available result analysis opportunities and data export options.

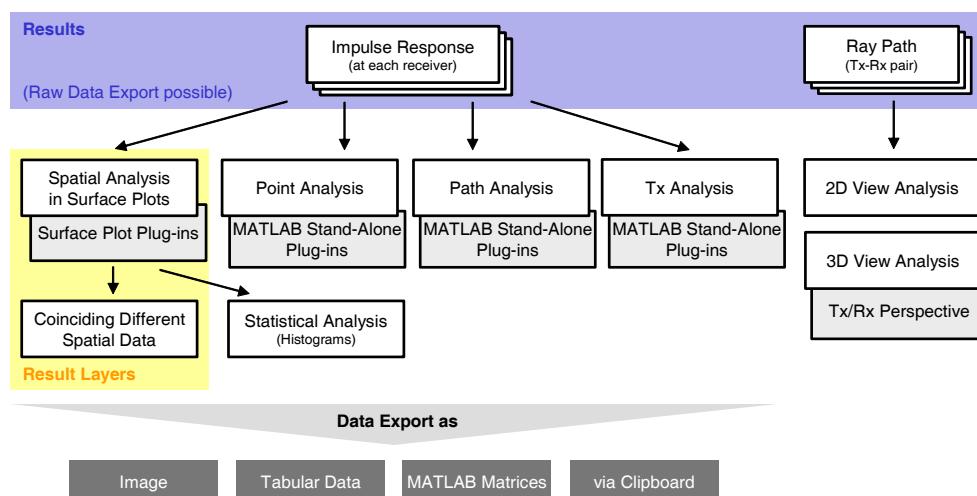


Fig. 6-2 Supported types of result analysis

Most of all parameters of interest can be derived from these basic results. Post-processed measures can be presented directly in the 2D and 3D views as surface plots, or chart diagrams and tables can be used for data analysis. The distinct analysis opportunities are introduced shortly below:

- **Spatial Analysis:** Several default plots (coverage, delay spread, etc.) can be derived from the impulse responses, each creating a new result layer. Moreover, customized network measures can be displayed in surface plot with the help of user-defined plug-ins. Different layers can be compared in various ways by coinciding them. Furthermore, from each surface plot a histogram can be derived.
  - **Point Analysis:** The impulse response at a point (i.e. a receiver) in the 2D view can be analyzed in detail incl. DoA analysis. Additionally, any kind of derived measures can be calculated from the receiver's impulse response by using user-defined plug-ins (MATLAB stand-alone applications).
  - **Path Analysis:** The received power and the delay spread can be analyzed along a polyline in the 2D view. All impulse responses of the receivers in the path of the polyline are taken into account. Additionally, any kind of derived measures can be calculated from these impulse responses by using user-defined plug-ins (MATLAB stand-alone applications).
  - **Tx Analysis:** The coverage range from a transmitter for different antenna tilts, antenna azimuth angles, as well as different transmit powers can be assessed by using specialized analysis functions. Additionally, any kind of derived measures can be calculated from all receiver's impulse responses reachable from the transmitter by using user-defined plug-ins (MATLAB stand-alone applications).

All the listed data evaluation functions offer to present data in different forms, e.g. as surface plot, table, histogram, or chart graph. Once such a presentation has been created, it is easily possible to convert it into another presentation as apparent from Fig. 6-3. Due to this high degree of flexibility, simulation data can be post-processed very efficiently, and results can be obtained quickly.

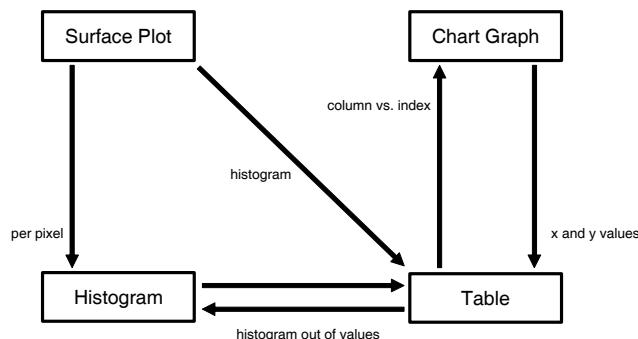


Fig. 6-3 Data transformation options supported by RPS

Moreover, the ray paths can be inspected in detail in the 2D or 3D view if the according simulation option (Save Ray Paths in the Simulation tab of the preferences dialog) has been selected. It is possible to display outgoing or incident rays at a transmitter/receiver (even from the respective network element's perspective), and all ray paths for a selected receiver can be shown. These analysis functions are especially useful for DoA analysis.

Both the raw impulse responses and all derived data can be exported in proper formats. Surface plots and diagrams can be saved as images. Generally, all data can be stored into ASCII files (tabular data) or as MATLAB matrices. Also, various clipboard copy functions are available.

### 6.3 Spatial Analysis

RPS provides built-in surface plot functions for the following, frequently used parameters derived from channel impulse responses that are presented in different result layers:

- coverage
- best serving cell ID
- signal-to-interference ratio
- delay spread (of best serving cell)
- angular spread (of best serving cell)

These result layers can be easily created by clicking the according icons from the standard toolbar or from the according entries in the Results -> Surface Plots submenu. Furthermore, arbitrary custom surface plot layers can be generated by using user-defined plug-ins. Surface plots are available in both 2D and 3D views. In 2D views, the value for the receiver under the mouse pointer is visualized by tooltips.

For a statistical analysis, it is possible to derive a histogram from the data contained in any kind of surface plot by using the method described in Subsection 6.3.10.

### 6.3.1 Result Surface Plot Layers

For every displayed result parameter a new layer is created. All available result layers are listed in the Layers tab of the tree window. For an efficient result analysis, all the usual layer evaluation functions can be used. In particular, the following options seem to be interesting:

- To display an already existing single layer, just double-click on it in the Layers tab of the tree window. An impressive way of comparing two layers is to alternately double-clicking on either of them, such that their presentation is constantly exchanged.
- Layers can be overlaid. To make a distinct layer visible, just activate the corresponding checkbox in the Layers tab of the tree window. In order to bring up or push down a certain layer, one can reorder the layers by the Move ... functions from that layer's context menu. Alternatively, a layer can be dragged graphically by the mouse to another level in the layers list.
- Layers can have individual graphical presentations. For every layer, the layer settings can be chosen independently. The properties of the currently active layer can be edited in the layer settings dialog by clicking into the black legend area to the left. Interesting graphical properties of result layers for a flexible analysis are, among others, the alpha blending factor, the color palette, the individual color space transparency, etc. For each layer type, these graphical presentation settings are persistently stored in the `legend.ini` file which is automatically created and maintained.
- If a layer is re-created (i.e. a layer of the same type already exists), normally the according layer item in the Layers tab of the tree window is reused for the new layer. The old surface plot is overridden then. If the user wishes to create a new layer item, the `<Shift>` key must be held down during creation of the layer.

A detailed description on the result layer concept can be found in Section 3.3.2.

### 6.3.2 Received Power (Coverage) Surface Plots



An often used analysis function is the site specific prediction of the received power.

RPS determines in the first step the best serving cell ID for each receiver position. In the second step, the square sum of all multipath components is calculated from the signals where the source is the serving cell. An example of a coverage plot in an outdoor environment is given in Fig. 6-4.

The logarithmic power values are translated then into colors, given by the current color map definition. The presentation properties of the current layer can be changed in the layer settings dialog (see Section 3.4.3).

All pixels with no reception power (i.e. where the power is zero) are left empty.

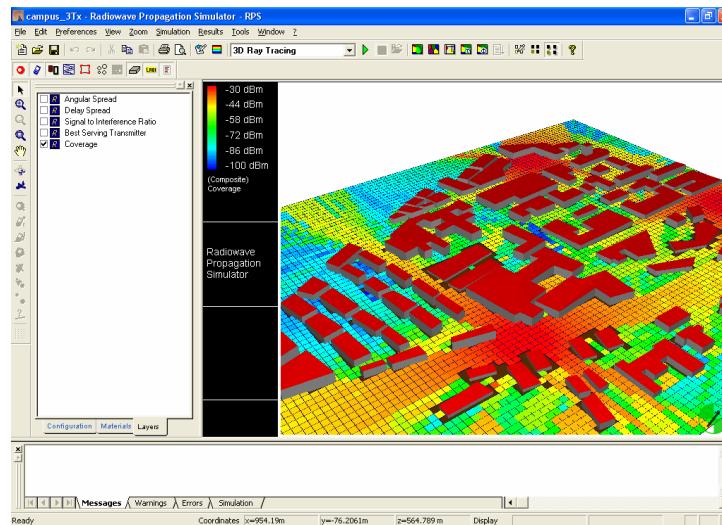


Fig. 6-4 Coverage distribution in an outdoor area (3D View)

### 6.3.3 Best Serving Cell Surface Plots



During the planning and design process of wireless communication systems it is necessary to get an overview which cell regions and cell sizes are associated with each of the transmitters. The best serving cell for each receiver position is calculated by determining that transmitter that contributes the highest field strength to the receiving signal at the respective receiver position (or least path loss).

The best serving cell algorithm draws the cell area for each transmitter in a different color. In 2D view, an additional legend can be shown that contains a list of all cells with their associated presentation colors. By default, the Discrete Colors flag in the layer settings dialog is set for this layer which assigns each cell a dedicated color automatically. However, this flag can also be deactivated. Then the specified color map would be applied directly. An example for a best serving cell plot in an outdoor environment is given in Fig. 6-5.

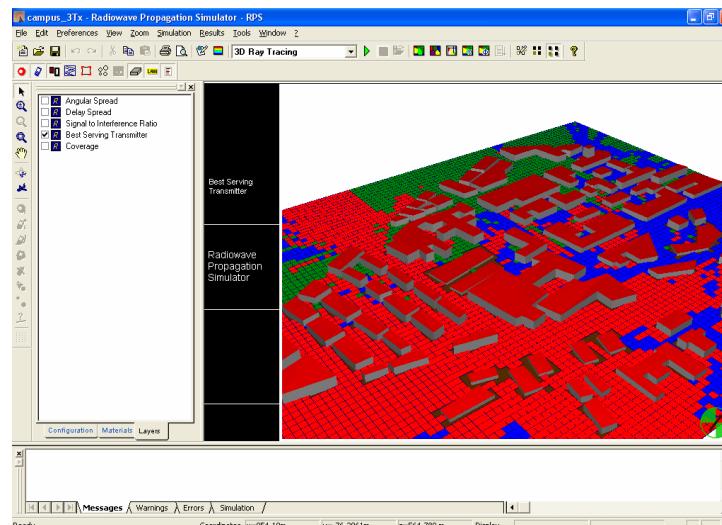


Fig. 6-5 Best serving cell plot in an outdoor environment (3D view)

### 6.3.4 Signal-to-Interference Ratio (SIR) Surface Plots

The signal-to-interference ratio is defined as the difference of the received power from the serving cell and the sum of all other (interference) signals in dB. For this type of surface plot, at least 2 transmitters must be available to calculate any SIR value.



Similar to the received power RPS determines in the first step the best serving cell ID for each receiver position. In the second step, the received power of the serving cell and the interference power of all other signals are calculated. The values are presented in dB. An example for an SIR plot in an outdoor environment is given in Fig. 6-6.

The signal-to-interference ratio values are translated then into colors, given by the current color map definition. The presentation properties of the current layer can be changed in the layer settings dialog (see Section 3.4.3).

Only those receiver positions are drawn with color where an SIR value was determined. On the contrary, areas that are not subject to interference from other cells despite the serving cell remain uncolored. Thus, potential interference regions can be identified easily.

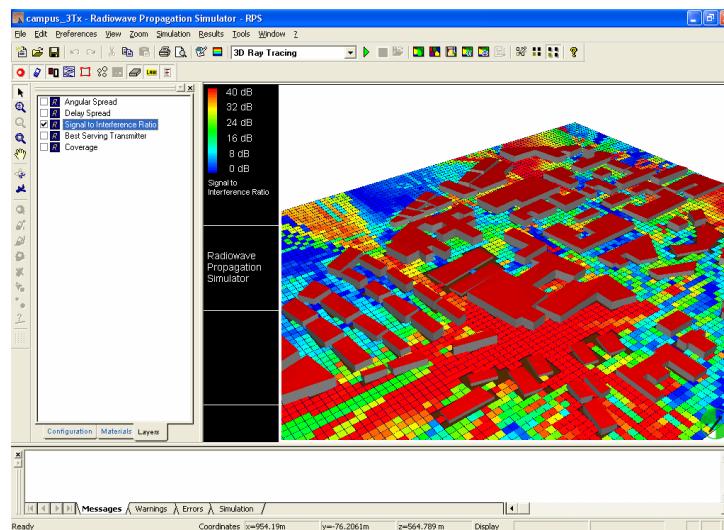


Fig. 6-6 Signal-to-Interference Ratio in an outdoor environment (3D view)

### 6.3.5 Delay Spread Surface Plots

The delay spread is the standard deviation of the mean excess delay of the received multipath components.



Similar to the received power RPS determines in the first step the best serving cell ID for each receiver position. In the second step, the delay spread of all multipath components is calculated from signals where the source is the serving cell. The values are presented in [ns]. An example of a delay spread plot in an outdoor environment is given in Fig. 6-7.

The delay spread values are translated then into colors, given by the current color map definition. The presentation properties of the current layer can be changed in the layer settings dialog (see Section 3.4.3).

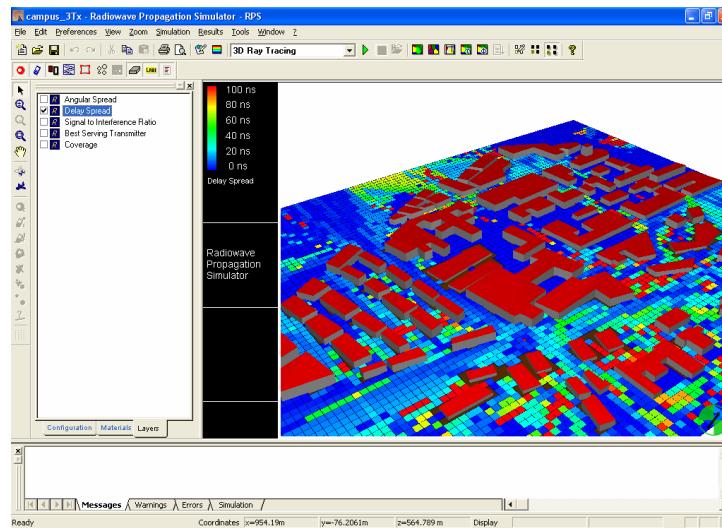


Fig. 6-7 Delay spread distribution in an outdoor area (3D view)

### 6.3.6 Angular Spread Surface Plots

The angular spread is the standard deviation of the angular distribution of the received multipath components from a certain transmitter in the horizontal plane.



Similar to the received power RPS determines in the first step the best serving cell ID for each receiver position. In the second step, the angular spread of all multipath components is calculated from signals where the source is the serving cell. The values are presented in [°]. An example of an angular spread plot in an outdoor environment is given in Fig. 6-8.

The angular spread values are translated then into colors, given by the current color map definition. The presentation properties of the current layer can be changed in the layer settings dialog (see Section 3.4.3).

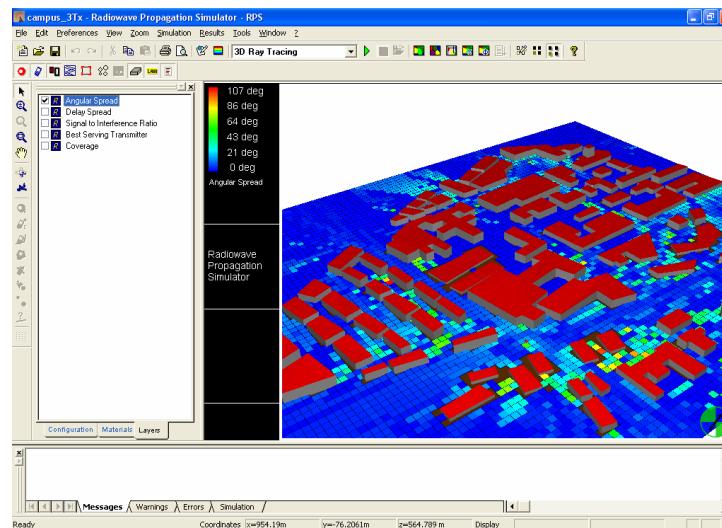


Fig. 6-8 Angular spread distribution in an outdoor area (3D view)

### 6.3.7 Custom Surface Plots

RPS provides an interface that allows the user to present further parameters, derived from channel impulse responses, directly in RPS. All available custom surface plot plug-ins are listed in the Configuration tab of the tree window. The currently active plug-in can be invoked by simply clicking the icon in the standard toolbar.

The customized calculation of the underlying data for surface plots is done by plug-ins. Working with surface plot plug-ins is discussed in detail in Section 6.4.

### 6.3.8 Contour Plots

In addition to the common surface plots where for each receiver a colored rectangle (pixel) is drawn, RPS can present each surface plot layer as a contour plot. A contour plot is comparable with an elevation map, where lines are drawn along regions with equal elevation over ground. This type of view can be used to represent outage regions, areas of good coverage etc. So this option is especially interesting when combining with a non-linear color map. An example is given in Fig. 6-9. The number of limits depends on the applied color palette.

The contour plot view for the currently active layer can be toggled in the layer settings dialog by selecting/deselecting the Draw Contour Lines flag.

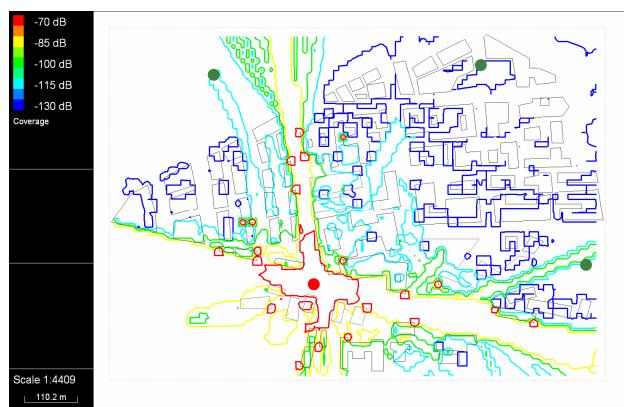


Fig. 6-9 Sample Contour Plot

Note that the contour plot option can only be displayed in 2D view.

### 6.3.9 Coinciding Surface Plot Layers

An interesting option is to investigate the possible relation between two different measures. The starting point are surface plot layers of the according data. Using the entry Coincide with other Layer... from a layer's context menu in the Layers tab of the tree window, a dialog is shown where another layer and different options for comparison can be chosen. This dialog is shown in Fig. 6-10. The elements of it are described in Table 6-1.

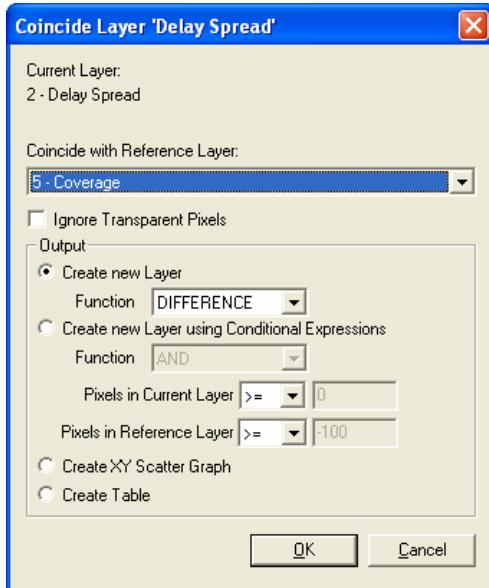


Fig. 6-10 Coincide layers dialog

Table 6-1 Parameters to coincide layers

| Parameter                                      | Description  |
|--|--|
| Current Layer                                  | The layer from which the dialog was called.  |
| Coincide with Reference Layer                  | The second layer for the comparison.   |
| Ignore Transparent Pixels                      | Transparent pixels are excluded from the combination.  |
| <b>Output</b>                                  |  |
| Create new Layer                               | A radio button to create a new surface plot layer from the given two layers, where the pixels of the layers are interrelated pair wise by a selectable function.   |
| Function                                       | A function to interrelate two layers, creating a new layer. Available functions are 'DIFFERENCE', 'SUM', 'QUOTIENT', 'PRODUCT', 'AND', 'OR', and 'XOR'.  |
| Create new Layer using Conditional Expressions | A radio button to create a new surface plot layer from the given two layers with additional threshold check. The result is a layer containing either of the values 100% and 0%. A pixel has the value 100% if the conditions is fulfilled; otherwise it is 0%. |
| Function                                       | The user can choose whether the conditions for both layers shall be fulfilled simultaneously ('AND') or separately ('OR').   |
| Pixels in Current Layer                        | The definition of the condition for the first layer. The user can specify the relation (either of the signs ' $\geq$ ', ' $=$ ' and ' $\leq$ ') and the comparison threshold.  |
| Pixels in Reference Layer                      | The definition of the condition for the reference layer. The user can specify the relation (either of the signs ' $\geq$ ', ' $=$ ' and ' $\leq$ ') and the comparison threshold.  |
| Create XY Scatter Graph                        | A radio button to create a scatter graph from the given two layers. For each pixel pair of the layers, a dot is set in the resulting graph. The first layer is mapped on the y-axis, and the second layer is mapped on the x-axis.                             |
| Create Table                                   | A radio button to create a table from the given two layers. The data in the table is the same as is created by the XY Scatter Graph function.  |

At first it shall be demonstrated how this function can be exploited to easily show the difference in coverage when an antenna is turned. For this showcase first a coverage plot needs to be created for the initial network setup. This is done by clicking the icon in the standard toolbar. Note that the result layer just created is also listed in the Layers tab of the tree window.

Then a certain antenna can be turned in azimuth which certainly will change the coverage situation. Note that it is not necessary to re-calculate the coverage! For the comparison another coverage plot layer is needed which can now be created by again clicking the icon while simultaneously holding down the **<Shift>** key. If the **<Shift>** key would not be held down, the previous coverage layer would be overridden.

Now the two coverage layers can be compared by creating a new layer representing the difference of the others. To accomplish this, choose the entry Coincide with other Layer from the latest created layer's context menu in the Layers tab of the tree window. The coincide layers dialog will be opened where the other layer should be chosen for comparison. Furthermore, select the option Create new Layer and choose the function '**DIFFERENCE**'. The resulting comparison layer representing the coverage change after the antenna turn is immediately created. An example for such a derived layer is shown in Fig. 6-11.

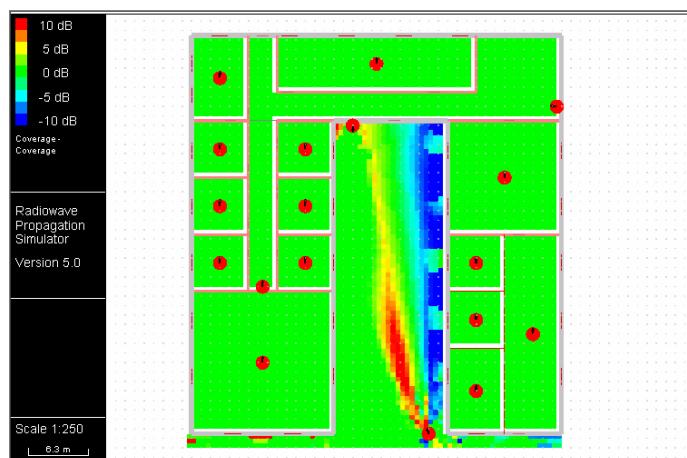


Fig. 6-11 Coverage change after turning an antenna

If, as another example, the relation between the signal-to-interference ratio over the mean pathloss (i.e. coverage) shall be investigated, the coincide layers function can be used to create a chart graph from the values of these two layers mapped onto the x- and y-axes of the graph (XY scatter graph). This scatter graph is shown in a separate window. Additionally, it can be useful to check the Ignore Transparent Pixels flag to exclude those pixels from the comparison that are not defined in either of the underlying surface plots. A possible result is shown in Fig. 6-12.

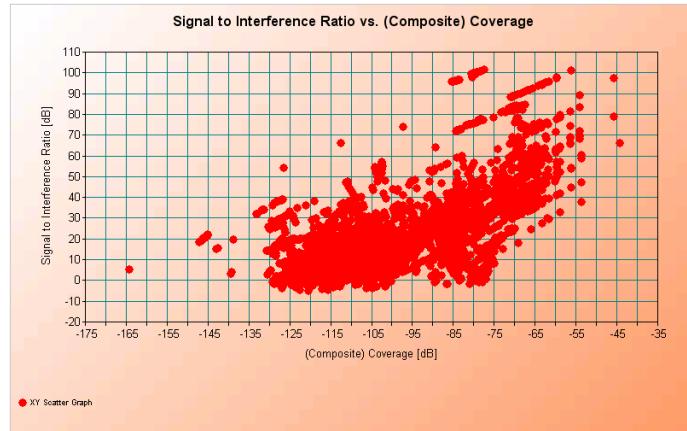


Fig. 6-12 Sample chart graph of two coincided layers

See Section 6.10.1 for further data post-processing and transformation options of chart graphs.

### 6.3.10 Deriving Histograms from Surface Plots

The data of the currently active surface plot layer can be easily translated into a histogram view. Using the menu items Results -> Chart Diagrams -> Histogram from Surface Plot as Chart or Results -> Chart Diagrams -> Histogram from Surface Plot as Table, the underlying pixel data of the surface plot is translated into a statistical distribution shown in a chart or a table, respectively. The same functions can be reached from the context menu of result data layers which can be displayed by right-clicking into a surface plot of that result parameter. The histogram or the table are presented in a separate window. An example of the distribution of the coverage (in terms of received power) is given in Fig. 6-13.

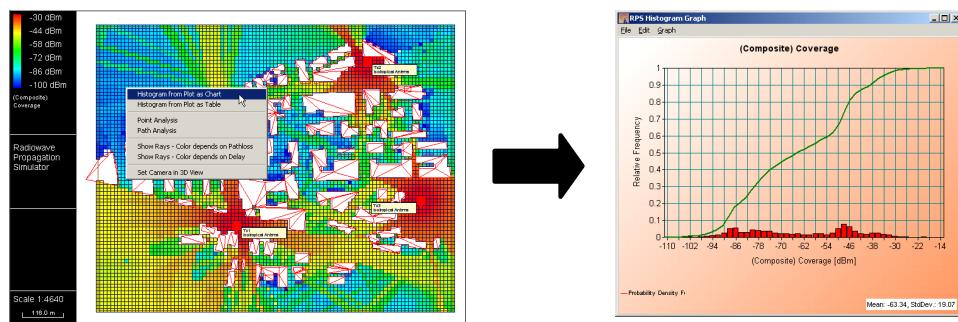


Fig. 6-13 An example of creating a histogram from a surface plot

See Section 6.10.2 for further data post-processing and transformation options of histograms.

### 6.3.11 Exporting Surface Plots Layers

The matrix contents of a surface plot layer can be saved into a TIFF file by using the Save as TIFF image... entry in the context menu of a layer item in the Layers tab of the tree window, irrespective of the current presentation of the layer in the newing area. If the contained values in the layer shall be best retained in the resulting TIFF file, it is recommended to use a grey color palette for the layer.

Independent of the before mentioned direct storage of layers in an image file, the current contents shown in the viewing area, both in 2D and 3D view, can be copied to the clipboard using <Ctrl-C>, or by clicking the  icon from the standard toolbar, or by choosing the menu entry Edit -> Copy. This copy action could also be applied to graphs and tables in the viewing area. If the viewing area shall be copied as a pure image, the menu entry Edit -> Copy Plot as Graphic should be used instead.

### 6.3.12 Printing Surface Plots

Generally, every 2D or 3D view of surface plots can be printed in true scale. A comfortable print preview is also offered to control the printing result in advance.

The print preview is displayed by clicking the  icon (tooltip Print Preview) from the standard toolbar or, alternatively, using the menu entry File -> Print Preview..... A new window is opened that shows the print preview and has some additional buttons to e.g. start the printing process.

In order to print the current presentation in the viewing area, the menu entry File -> Print... can be used. Then the print dialog is opened where all printing settings can be made. If the proper printer driver has already been selected before, the printing process can directly be started by clicking the  icon (tooltip Print) from the standard toolbar.

## 6.4 Surface Plot Plug-ins

Besides the built-in surface plot functions it is possible to create user-defined surface plot layers by implementing customized plug-ins. RPS provides a plug-in interface that can be easily used to implement and integrate custom surface plot functions written in Visual C++ or as MATLAB m-file. This Section describes how to use custom surface plot plug-ins and how to implement your own surface plot function. The basics of the RPS plug-in technology are presented in Section 3.8.

### 6.4.1 Using Custom Surface Plots

Custom surface plots allow the presentation of user-defined data, obtained from field strength prediction results. RPS provides a couple of plug-ins including source code for presentations additionally to built-in surface plots, among them:

- Coverage (Highest Rx Power) (DLL and m-file)
- Delay Spread (DLL and m-file)
- Number of Receivable Base Stations (DLL and m-file)
- Least Path Loss (m-file)
- Angular Spread PHI- and THETA-plane at BS and MS (m-file)

All available surface plot models are listed in the Configuration tab of the tree window below the Plugins – Surface Plots node. An example view is given in Fig. 6-14. There they can be selected by clicking their checkbox.

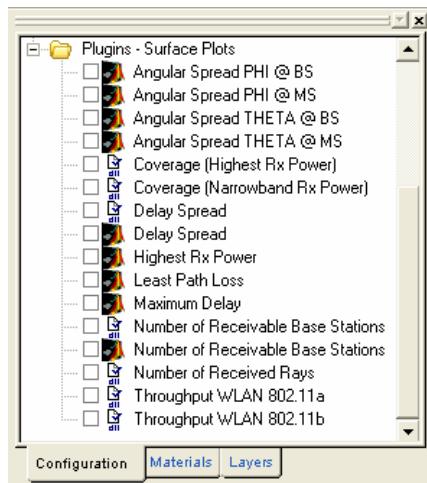


Fig. 6-14 Overview of user-defined surface plot models in the tree window

The coverage and delay spread plug-ins demonstrate how RPS determines these parameters internally. The plug-in Number of Receivable Base Stations determines the count of “visible” base stations (Tx) at a mobile station (Rx), where the strength of the signals are higher than an adjustable sensitivity value.



A custom surface plot using the currently selected plug-in can be generated by the menu entry Results -> Surface Plots -> Plugin for Surface Plots or using the corresponding icon from the standard toolbar.

The range limits of the surface plot can be adjusted in the layer settings dialog (see Section 3.4.3) after the plot is generated.

RPS is able to hide non-defined values in a plot. This is accomplished by NaN values in m-files, or by setting these values to NaN's using `fmod(0, 0)` function in DLLs for instance. This presentation mode can be useful to highlight problem zones, or present only regions that are inside a certain range limit.

Currently, m-file matrix mode processing is supported for custom surface plots only.

#### 6.4.2 Developing User-Defined Surface Plot Plug-ins

This Section explains how user-defined plug-ins can be developed using the RPS DLL interface or the use of m-files for surface plots by writing only some lines of code. DLLs must be written and compiled in the MS Visual C++ 6 environment. Templates are available on the RPS CD-ROM. M-files are MATLAB scripts that can be generated with any text editor. The use of the built-in RPS Script Editor or the MATLAB Script Editor is recommended.

User-defined surface plots allow the customized post-processing and presentation of data, derived from site-specific channel impulse responses. Plug-ins translate the raw channel impulse response data into user-defined results. These results are presented then as a colorized surface plot in 2D or 3D views. The available input parameters are summarized in Table 6-2.

Table 6-2 Available parameters for user-defined surface plot plug-ins

| Parameter  | Description  |
|--|--|
| <b>General Parameters</b>  |  |
| RxPosition   | position(x [m], y [m], z [m]) of the current mobile station                |
| Frequency  | carrier frequency [GHz]  |
| Bandwidth  | bandwidth [GHz]  |
| NoiseFloor   | noise floor parameter [dBm] or unitless in dB, respectively                |
| Range_Min  | lower range limit for surface plot color map                               |
| Range_Max  | upper range limit for surface plot color map                               |
| <b>Parameters for base stations (N = number of available base stations)</b>  |  |
| TxPositionArray  | vector of N Positions(x [m], y [m], z [m])                                 |
| TxPowerArray   | vector of N TxPower values [dBm] or unitless in [dB]                       |
| <b>Parameters for channel impulse responses (M = number of available multipath components; equal to the number of detected rays)</b> |  |
| Magnitude  | magnitude of the multipath   |
| Tau  | total excess delay [ns]  |
| Rx_PHI   | angle of arrival PHI at the mobile station [deg]                           |
| Rx_THETA   | angle of arrival THETA at the mobile station [deg]                         |
| Tx_PHI   | outgoing angle PHI at the base station [deg]                               |
| Tx_THETA   | outgoing angle THETA at the base station [deg]                             |
| TxID   | zero-based index of the transmitter ID (source of the multipath component) |

The interaction between the surface plot plug-in and RPS works as follows:

1. RPS fills the parameter list for one or more mobile station positions.
2. RPS calls a custom function in the plug-in (a DLL or a MATLAB script).
3. RPS retrieves the data for one or more mobile station positions.

The parallel translation of data for multiple mobile station positions in one step is called matrix mode here. Currently this mode is supported by plug-ins provided as MATLAB scripts.

#### 6.4.2.1 DLL Plug-in SDK for Custom Surface Plots

This type of plug-in is realized as a C++ class, called `CPlugin_RPS_SP`. It contains the parameter list and “Get/Set” functions for accessing them. Fig. 6-15 summarizes the methods and attributes of the classes.

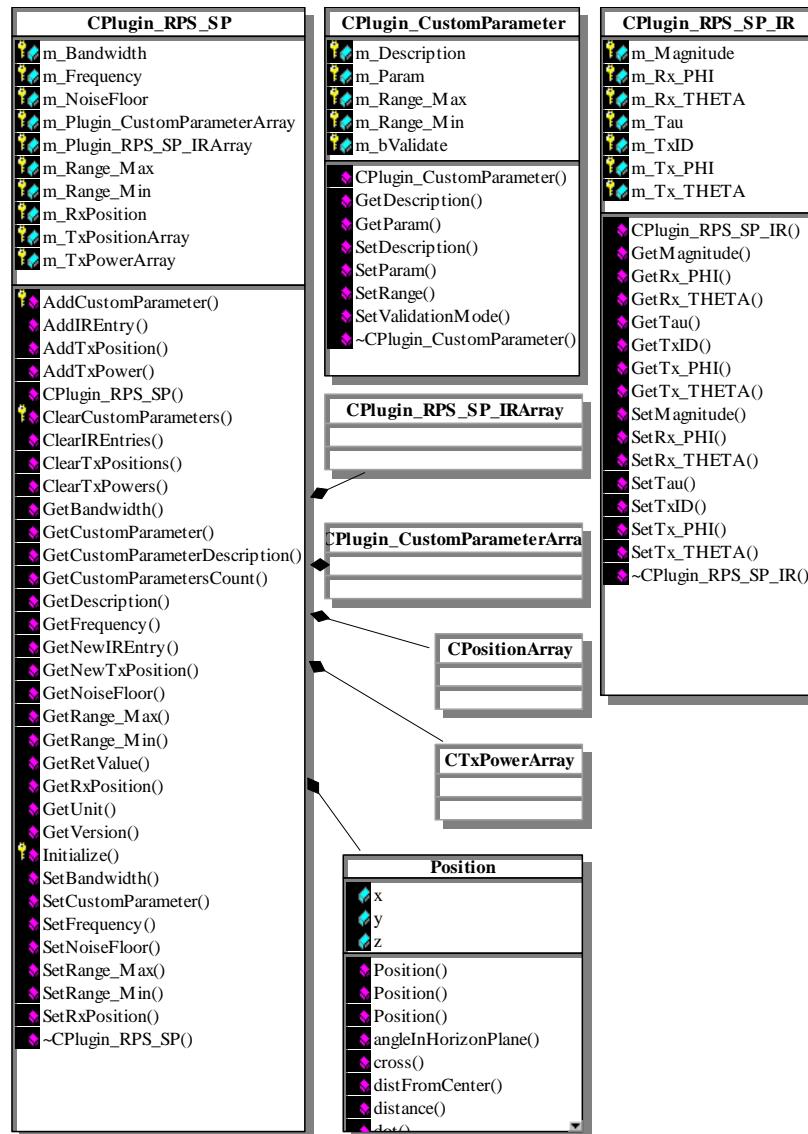


Fig. 6-15 Class view of the custom surface plot library

In order to write a user-defined plug-in, the following four functions should be customized. The declarations of these functions can be found in the `plugin_customfunctions.cpp` file.

```
void CPlugin_RPS_SP::Initialize()
```

This function is called by RPS when the plug-in is attached by the framework. Some additional initialization code can be added here. In the following example, we use custom parameters for range limit definitions. See the explanation in the DLL plug-in SDK for custom propagation models given in Section 5.5.2.1 on how to work with customizable parameters. An example with regard to user-defined surface plots is given here:

```
m_Frequency = 1.0;
m_Bandwidth = 1.0;
m_NoiseFloor = -130.0;
m_Range_Min = 0;
m_Range_Max = 1;
```

```
// Parameters: Value, Parameter Description, Lower Limit, Upper Limit, ...
AddCustomParameter("-130", "Range Min");
AddCustomParameter("-70", "Range Max");
```

#### **char\* CPlugin\_RPS\_SP::GetDescription()**

This function should return a string that describes the plug-in briefly. This string is used in the RPS plug-in dialog. It is independent of the DLL filename.

#### **char\* CPlugin\_RPS\_SP::GetUnit()**

This function should return a short string, containing a unit for the data to be presented, for instance:

```
return "dBm";
```

#### **double CPlugin\_RPS\_SP::GetRetValue()**

This function is called for each mobile station position once. The parameter list is already updated by RPS before this function is called. The return value (double) is stored at the corresponding Rx position. The following example shows the implementation of a function that determines the highest Rx power at a receiver position:

```
double RetValue = 0;
int i;

// Obtain the best serving cell id
double maxMag = 0;
int txID = -1;
for (i = 0; i < m_Plugin_RPS_SP_IRArray.size(); i++)
{
    CPlugin_RPS_SP_IR* ir = m_Plugin_RPS_SP_IRArray.at(i);
    if (std::norm(ir->GetMagnitude())>=maxMag)
    {
        maxMag = std::norm(ir->GetMagnitude());
        txID = ir->GetTxID();
    }
}

for (i = 0; i < m_Plugin_RPS_SP_IRArray.size(); i++)
{
    CPlugin_RPS_SP_IR* ir = m_Plugin_RPS_SP_IRArray.at(i);
    if (txID == ir->GetTxID())
        RetValue += std::norm(ir->GetMagnitude());
}

RetValue = 10.0*log10(RetValue);

m_Range_Min = atof(GetCustomParameter(0));
m_Range_Max = atof(GetCustomParameter(1));

return RetValue;
```

The source code for more custom surface plot plug-ins is available on the RPS CD-ROM in the SDK directory.

#### 6.4.2.2 MATLAB Plug-in SDK for Custom Surface Plots

A custom surface plot model can be implemented as MATLAB script (m-file) as well. The input parameter list is stored in the MATLAB structure array **RPS\_Input**. It contains the following fields:

```

Frequency: [1x1 double]
Bandwidth: [1x1 double]
NoiseFloor: [1x1 double]
RxPosition: [3xM double] % position(x,y,z)
TxPosition: [3xN double] % position(x,y,z)
TxPowerdB: [1xN double]
    IR: Magnitude: [CxM double]
        Tau: [CxM double]
        Rx_PHI: [CxM double]
        Rx_THETA: [CxM double]
        Tx_PHI: [CxM double]
        Tx_THETA: [CxM double]
        TxID: [CxM double]

```

M: Number of mobile station positions computed in parallel

N: Number of available base stations

C: Maximal number of channel impulse response entries

In general, for each mobile BS-MS link MATLAB is called and the plug-in file is processed once. While this procedure may require a lot of computational overhead, a matrix mode is introduced that allows the computations for multiple base station positions to be performed in a single MATLAB call. This procedure is referred to as matrix processing mode here. The maximum number of items processed in parallel can be adjusted in the RPS plug-in dialog. Furthermore, a plug-in can support matrix processing or not. It is declared in the header part of the script file. More details about header information can be found in the MATLAB plug-in SDK in Section 5.5.2.2. In non-matrix mode, the number of mobile station positions M computed in parallel is always one.

C is defined by the channel impulse response with the highest number of entries. Since the number of entries of impulse responses for different mobile station positions can be different, the unused fields in the IR sub array are set to NaN's. For instance, if the first impulse response of the first mobile station contains 5 entries, but the IR for the second position has only 3 entries, the Matrix of TxID has a structure like the following example (links to three transmitters with TxID's 0, 1, and 2 are available):

```

TxID: 0 0
      1 0
      1 2
      2 NaN
      0 NaN

```

The evaluated results must be stored into the field **RetValue** in the **RPS\_Output** data structure as a column vector (or as a single value in non-matrix mode).

Temporary data can be stored into the **UserData** field of **RPS\_Output**. It can be used to save temporary results that would be necessary for the evaluation of data for the next mobile station position(s). Please take a look into the examples for more details.

#### 6.4.2.3 Support Functions for MATLAB Custom Surface Plot Plug-ins

In the **RPS\_Input** matrix the whole channel impulse response data is stored. In many cases, the highest Rx power, the best serving cell, or the link loss to each (receivable) base station is needed as an input argument. Therefore, support functions are provided that can be used to extract frequently used basic parameters from the **RPS\_Input** data structure.

The following functions are available:

- **RPSInput2RxPower**: transforms the **RPS\_Input** data structure into received powers
- **RPSInput2PathLoss**: transforms the **RPS\_Input** data structure into path loss
- **RPSInput2LeastPL**: transforms the **RPS\_Input** data structure into least path loss and returns the corresponding transmitter ID's of the best serving cell

These auxiliary MATLAB functions are located in the *MATLAB User Library Path*. It can be modified in the bottom line of the RPS plug-in dialog. The user is free to store additional support functions into this directory.

For more detailed information, please refer to the comments in the m-files. You can also type *help \$function* directly in the MATLAB command window to obtain a detailed description about the function.

## 6.5 Point Analysis

In the 2D view, the calculated impulse response can be analyzed in detail at any point where data are available. For such a point analysis, several analysis functions are provided such as

- impulse response diagram
- horizontal/vertical DoA diagram
- tabular data of impulse response

These functions are accessible by clicking with the right mouse button into the 2D view at a dedicated position and choosing Point Analysis from the appearing context menu. Then a dialog is shown that is presented in Fig. 6-16. The usage of this dialog is described in Table 6-3.

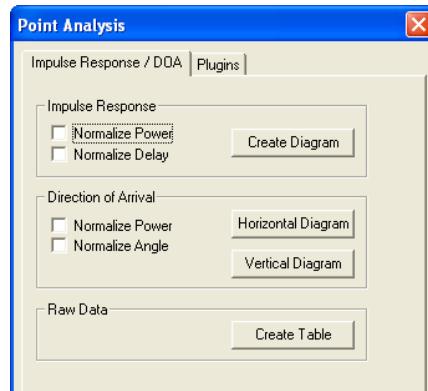


Fig. 6-16 Point analysis dialog

Table 6-3 Usage of the Impulse Response / DoA tab of the point analysis dialog

| Parameter                   | Description   |
|-----------------------------|---|
| <b>Impulse Response</b>     |   |
| Normalize Power             | If this flag is set, the impulse response is scaled in a way that the accumulated power of all paths received from a certain transmitter is equal to 0 dB/dBm.  |
| Normalize Delay             | If this flag is set, the impulse response is shifted to the left such that the first received path is aligned at $\tau = 0$ .   |
| <u>Create Diagram</u>       | This button creates a window with a diagram of the impulse response at the chosen position according to the settings of the normalize flags. For each received transmitter a separate graph is drawn in the diagram.                  |
| <b>Direction of Arrival</b> |   |
| Normalize Power             | If this flag is set, the polar impulse response is scaled in a way that the accumulated power of all paths received from a certain transmitter is equal to 0 dB/dBm.  |
| Normalize Angle             | If this flag is set, the polar impulse response is turned in a way that the LOS direction from a certain transmitter towards the receiver is aligned at $0^\circ$ . Note that the LOS path need not exist immediately.                |
| <u>Horizontal Diagram</u>   | This button creates a window with a polar diagram of the horizontal impulse response at the chosen position according to the settings of the normalize flags. For each received transmitter a separate graph is drawn in the diagram. |
| <u>Vertical Diagram</u>     | This button creates a window with a polar diagram of the vertical impulse response at the chosen position according to the settings of the normalize flags. For each received transmitter a separate graph is drawn in the diagram.   |
| <b>Raw Data</b>             |   |
| <u>Create Table</u>         | This button creates a window with a table grid containing the raw data of the impulse response. Each row represents a multipath components from a certain transmitter.  |

For each option a new window is opened. These result windows are non-modal such that different results can be compared easily by positioning the windows side by side. The contents of the windows can also be copied to the clipboard for further post-processing in other applications (e.g. MS Excel). More functionality of the diagram and table windows is given in Section 6.10.

The built-in point analysis functions can be extended by applying user-defined plug-ins that can be chosen in the second tab of the point analysis dialog. These plug-ins must be located in the subfolder `rps_pointanalysis` of the plug-in directory in order to be listed in the dialog. Read more about the implementation and usage of point analysis plug-ins in Section 6.8.

### 6.5.1 Site-specific Channel Impulse Response

The channel impulse responses are the fundamental results in time domain generated by the ray tracing algorithms. Most of all channel parameters can be derived from them. A channel impulse response contains discrete Dirac impulses with a given delay, and a complex magnitude. Depending on the system bandwidth, some impulses can be separated and others cannot. RPS adds all those complex values coherently that cannot be distinguished on the time axis within one delay bin. Empty points in the diagram are set to the configured noise floor (-110 dBm in Fig. 6-17). For each received transmitter a separate

graph is drawn in the diagram. A window with the impulse response at a dedicated position can be opened by clicking the [Create Diagram](#) button in the point analysis dialog.

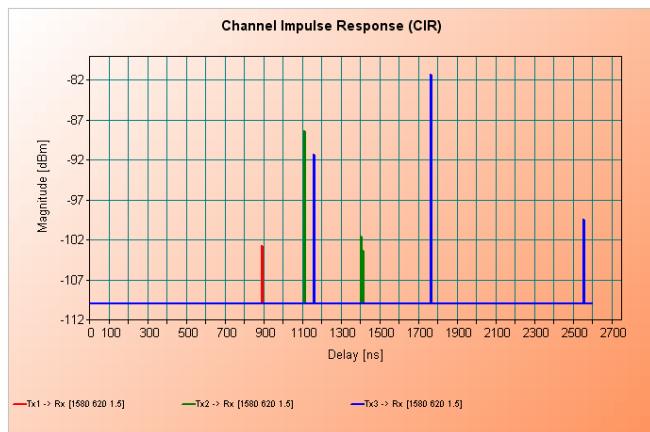


Fig. 6-17 Channel impulse responses at a receiver position from several transmitters

The raw data of an impulse response at a certain position can also be inspected in a tabular form. This can be done by clicking the [Create Table](#) button in the point analysis dialog which will open a separate window with a table grid containing the impulse response data. Each row represents a single multipath component. It is indicated from which transmitter the respective path was emitted. Besides the magnitude and phase information, also the outgoing and incident angles at transmitter and receiver, respectively, are given. An example of raw impulse response data is shown in Fig. 6-18.

The screenshot shows a Windows-style application window titled "RPS Data Table - Point Analysis". The window has a menu bar with "File" and "Edit" options. The main area is a table grid with the following data:

|   | Tx Name | Tau [ns] | Magnitude [dB] | Phase [°] | Tx Phi [°] | Tx Theta [°] | Rx Phi [°] | Rx Theta [°] |
|---|---------|----------|----------------|-----------|------------|--------------|------------|--------------|
| 1 | Tx3     | 1164.987 | -91.21431      | -44.95443 | 288.4349   | -6.123922    | 108.435    | -18.22531    |
| 2 | Tx3     | 1769.831 | -81.19226      | -53.87418 | 231.99     | 1            | 214.439    | 1.000054     |
| 3 | Tx3     | 2555.595 | -99.33302      | -131.2637 | 319.99     | 0            | 126.4768   | 0            |
| 4 | Tx2     | 1113.915 | -88.34227      | -24.08647 | 212.7352   | -4.149332    | 32.73523   | -8.961275    |
| 5 | Tx2     | 1405.963 | -101.5461      | -53.7726  | 229.99     | 1            | 358.4181   | -0.9999905   |
| 6 | Tx2     | 1419.715 | -103.3103      | 34.03375  | 229.99     | 1            | 339.3977   | -1.000045    |
| 7 | Tx1     | 894.966  | -102.6502      | -60.32672 | 46.5482    | -11.5979     | 226.548    | -44.6541     |

Fig. 6-18 Sample data of an impulse response

### 6.5.2 Site-specific Directions of Arrival

Every ray hits a receiver with a specific incidence angle, called direction of arrival (DoA) or angle of arrival (AoA). On the other hand, the outgoing angles at the transmitter can also be of interest. RPS stores directions of arrival at both transmitter and receiver objects in the impulse responses. The spatial resolution in the related charts is set to one degree. All magnitudes of rays that cannot be separated in space are added incoherently. In analogy to the channel impulse response chart, all empty points are replaced by the noise floor value. For each received transmitter a separate graph is drawn in the diagram. A window with the polar impulse response at a dedicated position can be opened by clicking the [Horizontal Diagram](#) or [Vertical Diagram](#) buttons, respectively, in the point analysis dialog. An example for both the horizontal and the vertical impulse response diagrams are given in Fig. 6-19.

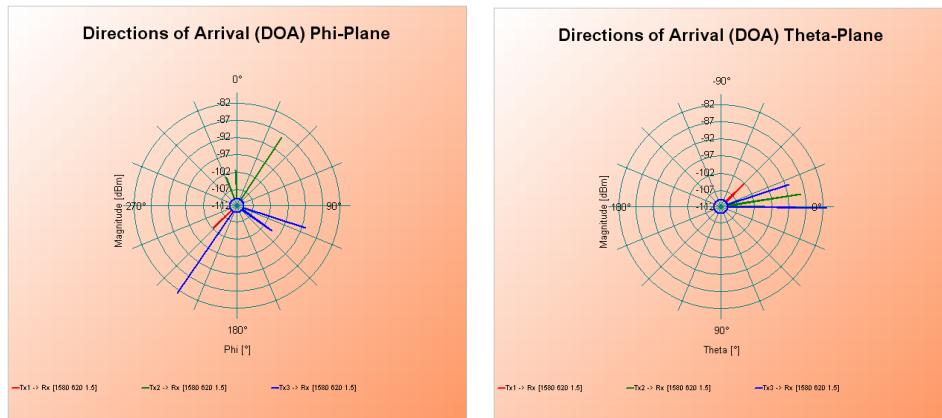


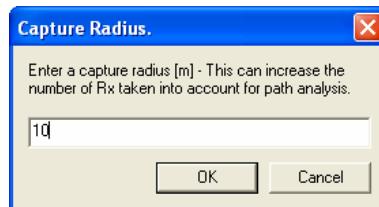
Fig. 6-19 Angles of arrival at a receiver positions for several received transmitters in  $\varphi$ -plane (horizontal diagram; left) and  $\theta$ -plane (vertical diagram; right)

## 6.6 Path Analysis

In many cases, a result analysis along a path (polyline) is very useful to understand e.g. the LOS/NLOS behavior in a certain environment. RPS has built-in path analysis functions to investigate

- the received power and
- the delay spread

along a given path in the 2D view. In order to specify the path, the path analysis mode must be entered first. Either choose Path Analysis from a receiver's context menu in the 2D view or enter this mode by clicking the icon in the paint toolbar to the left. Then a polyline can be drawn by setting the vertices of the path with the left mouse button. The line drawing can be stopped by clicking the right mouse button. Then a dialog appears where a capture radius around the polyline can be specified in order to reasonably catch receivers that are slightly displaced from the drawn path.



Next the path analysis dialog is shown that is presented in Fig. 6-20. The usage of this dialog is described in Table 6-4.

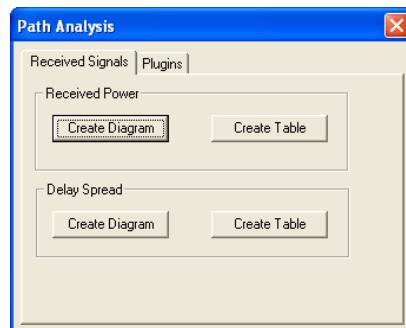


Fig. 6-20 Path analysis dialog

Table 6-4 Usage of the Received Signals tab of the path analysis dialog

| <i>Parameter</i>      | <i>Description</i>   |
|-----------------------|--|
| <b>Received Power</b> |  |
| <u>Create Diagram</u> | This button creates a diagram of the received <i>wideband</i> power along a specified path.<br>With <Shift> key held down: Creates a diagram of the received <i>narrowband</i> power along a specified path.                             |
| <u>Create Table</u>   | This button creates a table with the values of the received <i>wideband</i> power along a specified path.<br>With <Shift> key held down: Creates a table with the values of the received <i>narrowband</i> power along a specified path. |
| <b>Delay Spread</b>   |  |
| <u>Create Diagram</u> | This button creates a diagram of the delay spread along a specified path.  |
| <u>Create Table</u>   | This button creates a table with the values of the delay spread along a specified path.  |

For each option a new window is opened. These result windows are non-modal such that different results can be compared easily by positioning the windows side by side. The contents of the windows can also be copied to the clipboard for further post-processing in other applications (e.g. MS Excel). More functionality of the diagram and table windows is given in Section 6.10.

The built-in path analysis functions can be extended by applying user-defined plug-ins that can be chosen in the second tab of the path analysis dialog. These plug-ins must be located in the subfolder `rps_pathanalysis` of the plug-in directory in order to be listed in the dialog. Read more about the implementation and usage of path analysis plug-ins in Section 6.8.

### 6.6.1 Received Power along a Path

As an important measure, the received power along an arbitrary polyline in the investigation environment can be analyzed. Then, within a certain range (specified by a capture radius) around the drawn polyline, all available impulse response data are collected to calculate the received power. The related distance is measured along the path in the direction it was drawn.

The diagram of the wideband received power (addition of power values of all signals at a certain position) can be shown in a separate window by clicking the Create Diagram button

in the Received Power section of the path analysis dialog. The **<Shift>** key must be held down while clicking the same button, if the narrowband received power (incoherent addition of received signals) shall be displayed in the window. An example of the wideband and narrowband powers along a path is given in Fig. 6-21.

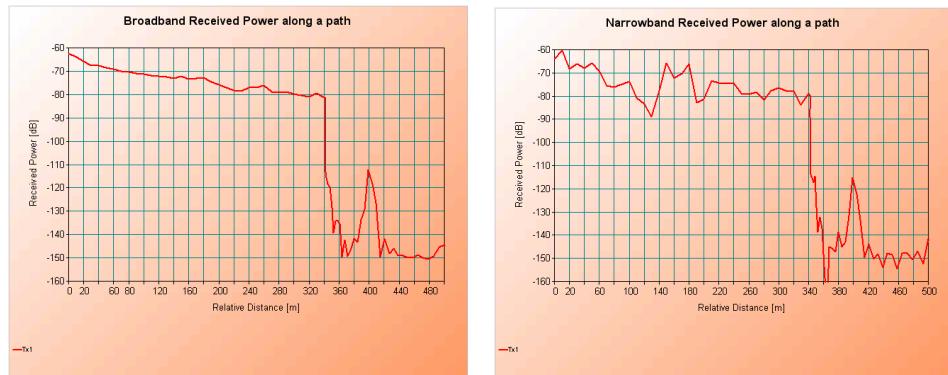


Fig. 6-21 Coverage along a path (left: wideband received power; right: narrowband received power)

The received power data can also be displayed in a table. This can be accomplished by clicking the Create Table button in the Received Power section of the path analysis dialog. Again, a distinction between the wideband and the narrowband power can be done by using the **<Shift>** key in the latter case.

### 6.6.2 Delay Spread along a Path

Likewise, the delay spread along a drawn path can be displayed in separate windows as diagram or table. This can be achieved by using the according buttons in the Delay Spread section of the path analysis dialog. Note that the delay spread is generally calculated based on the wideband approach – a distinction with the **<Shift>** key is not possible. See Fig. 6-22 for an example of a delay spread diagram along a path.

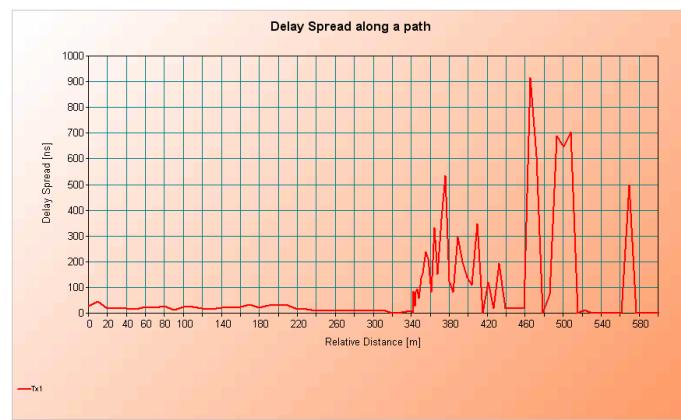


Fig. 6-22 Delay spread along a path

## 6.7 Transmitter Analysis

While the point and path analysis functions allow for an analysis of direct and derived measures based on impulse responses observed at receivers, there is also a Tx analysis to investigate different results related to a transmitter. In order to use the functions of the Tx

analysis, the Tx Edit Mode must be entered first by clicking the icon in the paint toolbar to the left. Then the context menu of a transmitter offers the following options:

- **Show LOS Preview Plot:** Gives a rough overview of the cell coverage area by highlighting the LOS area of that transmitter.
- **Statistical Analysis:** Charts over the coverage range of the transmitter with regard to antenna tilt or transmit power can be shown. The resulting diagrams contain several curves according to the configured min and max values and the step size.
- **DoA in PHI/THETA Plane:** Shows polar charts of the angles of outgoing rays in the PHI or THETA planes at the transmitter.

Some of these options are also available from the context menu of the transmitter items in the Configuration tab of the tree window.

### 6.7.1 LOS Preview Plot

A quick preview of the estimated coverage area of a cell can be easily created by using the [LOS Preview Plot](#) function from a transmitter's context menu. Then all receivers that have an LOS path to that transmitter and which are reachable above the configured noise floor are colored in red. See Section 4.2.2.5 for a more detailed explanation.

### 6.7.2 Statistical Analysis

For a more elaborated investigation of the coverage from a transmitter and for testing different transmitter configurations, a comprehensive statistical analysis for transmitters is provided. It can be invoked by choosing the entry [Statistical Analysis](#) from the respective transmitter's context menu in the 2D view. Then a dialog is opened that offers several tabs for different analysis options. They are explained in more detail in the following Subsections.

For each option a new result window is opened. These windows are non-modal such that different results can be compared easily by positioning the windows side by side. The contents of the windows can also be copied to the clipboard for further post-processing in other applications (e.g. MS Excel). More functionality of the diagram and table windows is given in Section 6.10.

The built-in path analysis functions can be extended by applying user-defined plug-ins that can be chosen in the fourth tab of the Tx analysis dialog. These plug-ins must be located in the subfolder `rps_txanalysis` of the plug-in directory in order to be listed in the dialog. In contrast to the other plug-ins for propagation models and surface plots, this type of plug-ins must be stand-alone executables which preferably should be generated by compiling MATLAB m-files that implement a certain data interface. Read more about the implementation and usage of Tx analysis plug-ins in Section 6.8.

#### 6.7.2.1 Antenna Tilt Analysis

The first option is an analysis of the coverage range for various antenna tilt angles. This function exploits the fact that the antenna orientation can also be changed after a coverage simulation without any recalculation of the channel. Of course, this function is only useful if the transmitter has an antenna with a certain vertical directivity – an isotropic antenna would yield the same results for all tilts. The Antenna Tilt tab of the statistical analysis dialog is shown in Fig. 6-23. The corresponding parameters are described in Table 6-5.

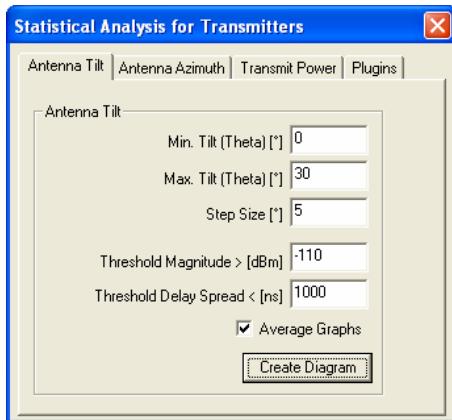


Fig. 6-23 Statistical Tx analysis dialog for antenna tilt investigations

Table 6-5 Parameters of the antenna tilt analysis at a transmitter

| Parameter                | Description  | Unit |
|--------------------------|--|------|
| Min. Tilt (Theta)        | The starting elevation angle (Theta) for the antenna tilt analysis.  | °    |
| Max. Tilt (Theta)        | The ending elevation angle (Theta) for the antenna tilt analysis.  | °    |
| Step Size                | The step size for the variation of the elevation angle of the antenna.   | °    |
| Threshold Magnitude >    | A lower threshold for the received signal strength that is considered for the assessment of the coverage range of the transmitter.                 | dBm  |
| Threshold Delay Spread < | An upper threshold for the delay spread of received signals that are still considered for the assessment of the coverage range of the transmitter. | ns   |
| Average Graphs           | If this flag is set, the resulting graphs will be averaged which makes the curves smoother.  | —    |
| <u>Create Diagram</u>    | This button creates a diagram with a separate coverage graph for each iterated antenna tilt angle.   | —    |

The antenna tilt analysis works as follows:

- The algorithm steps through the antenna tilt within the configured angular range at the given step size. In each step, a coverage curve will be generated at the end.
- For each antenna tilt angle, the algorithm steps through concentric circles around the transmitter with increasing radius. At each circle, a coverage percentage value is calculated for the current curve.
- All receivers lying on a circle are checked for the threshold conditions. If the signal strength at the receiver position is above the magnitude threshold and the delay spread is still below the delay spread threshold, this receiver is counted as valid; otherwise it would be counted as invalid. Finally, the coverage percentage for the current distance (i.e. radius of the circle) to the transmitter is determined as the portion of the valid receivers in relation to all receivers on the circle.

The resulting graphs are collected in the diagram window to be opened. Due to the complexity of the calculations and depending on the number of tilt steps, it can take a moment until the diagram is displayed. An example for an antenna tilt analysis is shown in Fig. 6-24.

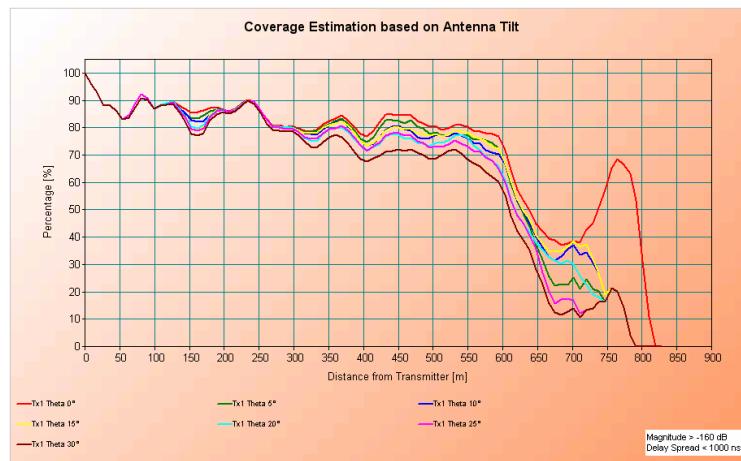


Fig. 6-24 Example of an antenna tilt analysis

Note that the antenna tilt analysis can only consider the existing receivers in the network configuration. If an area is not completely covered with receivers, the results would be different than if the entire area around the transmitter could be considered.

#### 6.7.2.2 Antenna Azimuth Angle Analysis

Similar to the antenna tilt analysis, an analysis of the coverage range for various antenna azimuth angles can be performed. This function also exploits the fact that the antenna orientation can be changed after a coverage simulation without any recalculation of the channel. Of course, this function is only useful if the transmitter has an antenna with a certain horizontal directivity – an isotropic antenna would yield the same results for all azimuth angles. The Antenna Azimuth tab of the statistical analysis dialog is shown in Fig. 6-25. The corresponding parameters are described in Table 6-6.

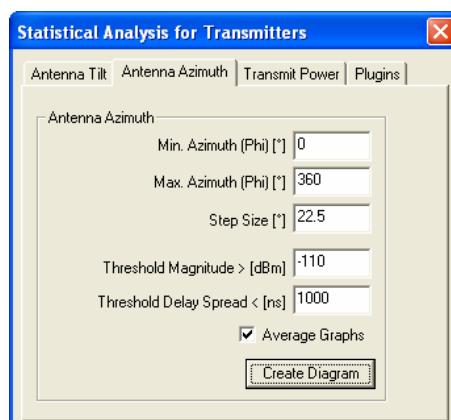


Fig. 6-25 Statistical Tx analysis dialog for antenna azimuth angle investigations

Table 6-6 Parameters of the antenna azimuth analysis at a transmitter

| Parameter                | Description  | Unit |
|--------------------------|--|------|
| Min. Azimuth (Phi)       | The starting azimuth angle (Phi) for the antenna azimuth analysis.   | °    |
| Max. Azimuth (Phi)       | The ending azimuth angle (Phi) for the antenna azimuth analysis.   | °    |
| Step Size                | The step size for the variation of the azimuth angle of the antenna.   | °    |
| Threshold Magnitude >    | A lower threshold for the received signal strength that is considered for the assessment of the coverage range of the transmitter.                 | dBm  |
| Threshold Delay Spread < | An upper threshold for the delay spread of received signals that are still considered for the assessment of the coverage range of the transmitter. | ns   |
| Average Graphs           | If this flag is set, the resulting graphs will be averaged which makes the curves smoother.  | —    |
| <u>Create Diagram</u>    | This button creates a diagram with a separate coverage graph for each iterated antenna azimuth angle.  | —    |

The algorithm of the antenna azimuth analysis works in a similar way as the antenna tilt analysis. However, here the antenna orientation is varied in the azimuth, not in the elevation. Each of the resulting graphs represents the coverage percentage for a certain antenna azimuth angle.

The resulting graphs are collected in the diagram window to be opened. Due to the complexity of the calculations and depending on the number of azimuth steps, it can take a moment until the diagram is displayed. An example for an antenna tilt analysis is shown in Fig. 6-26.

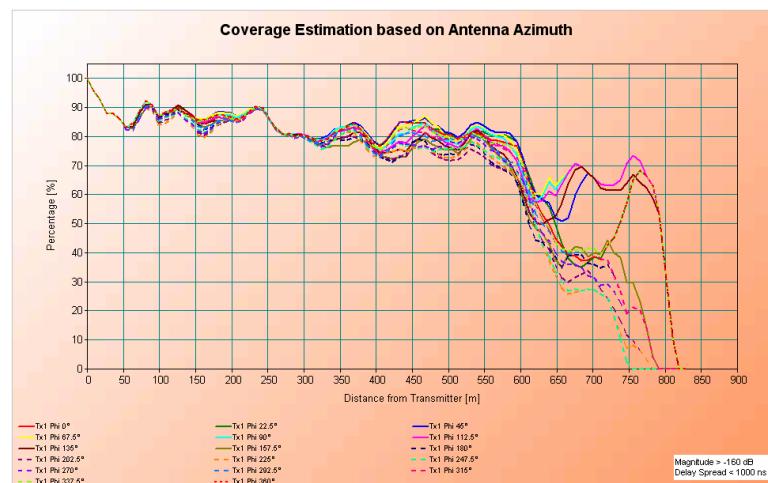


Fig. 6-26 Example of an antenna azimuth analysis

Note that the antenna azimuth analysis can only consider the existing receivers in the network configuration. If an area is not completely covered with receivers, the results would be different than if the entire area around the transmitter could be considered.

### 6.7.2.3 Transmit Power Analysis

Eventually, a coverage analysis due to different transmit powers can be performed. Even in this case, the channel is not simulated in each iteration; however, the upper power threshold should not be too high in relation to the originally configured transmit power. Otherwise the accuracy of the results could suffer slightly. The Transmit Power tab of the statistical analysis dialog is shown in Fig. 6-27. The corresponding parameters are described in Table 6-7.

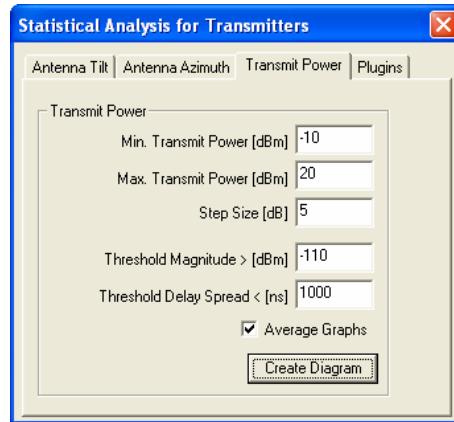


Fig. 6-27 Statistical Tx analysis dialog for transmit power investigations

Table 6-7 Parameters of the transmit power analysis at a transmitter

| Parameter                | Description  | Unit |
|--------------------------|--|------|
| Min. Transmit Power      | The starting transmit power for the transmit power analysis.   | dBm  |
| Max. Transmit Power      | The ending transmit power for the transmit power analysis.   | dBm  |
| Step Size                | The step size for the variation of the transmit power.   | dB   |
| Threshold Magnitude >    | A lower threshold for the received signal strength that is considered for the assessment of the coverage range of the transmitter.                 | dBm  |
| Threshold Delay Spread < | An upper threshold for the delay spread of received signals that are still considered for the assessment of the coverage range of the transmitter. | ns   |
| Average Graphs           | If this flag is set, the resulting graphs will be averaged which makes the curves smoother.  | —    |
| <u>Create Diagram</u>    | This button creates a diagram with a separate coverage graph for each iterated transmit power.   | —    |

Like the two other statistical analysis algorithms, the algorithm of the transmit power analysis iterates through several values of a variable which is the transmit power in this case. Each of the resulting graphs represents the coverage percentage for a certain transmit power.

The resulting graphs are collected in the diagram window to be opened. Due to the complexity of the calculations and depending on the number of transmit power steps, it can take a moment until the diagram is displayed. An example for a transmit power analysis is shown in Fig. 6-28.

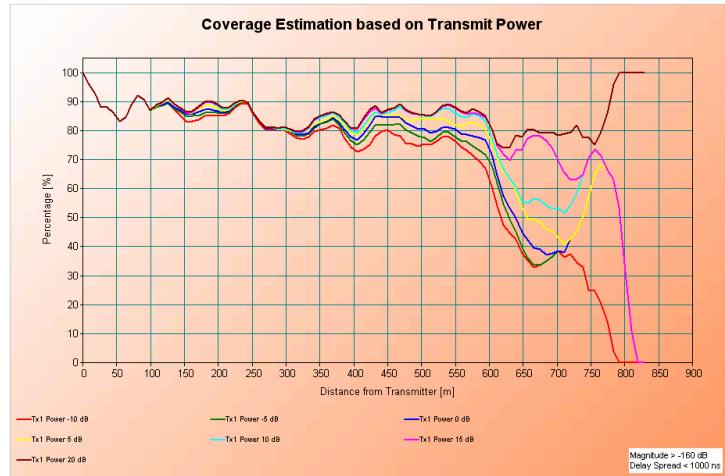


Fig. 6-28 Example of a transmit power analysis

Note that the transmit power analysis can only consider the existing receivers in the network configuration. If an area is not completely covered with receivers, the results would be different than if the entire area around the transmitter could be considered.

### 6.7.3 DoA Analysis

Similar to the DoA analysis at a receiver, also the angles of the outgoing rays at a transmitter can be investigated in the horizontal and vertical planes. The entries DOA PHI-Plane and DOA THETA-Plane from a transmitter's context menu can be used to open separate windows with polar diagrams of the horizontal or vertical angles of all outgoing rays, respectively. These graphs are comparable to the overall transmitted power contributing to any received power at all receiver positions. In analogy to the channel impulse response charts, all empty points are replaced by the noise floor value. Fig. 6-29 shows sample graphs of the horizontal and vertical DoA analysis at a transmitter.

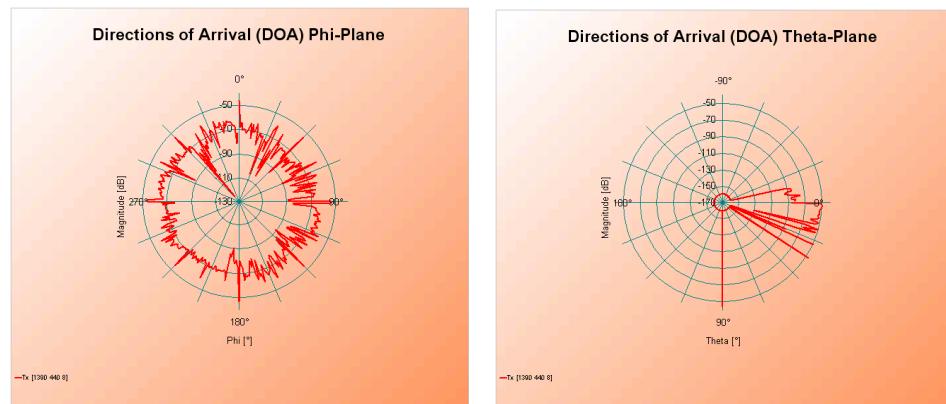


Fig. 6-29 Angles of outgoing rays at a transmitter (left: horizontal plane; right: vertical plane)

## 6.8 Post-Processing Plug-ins as Stand-Alone Applications

The post-processing functions of point, path, and Tx analysis can be arbitrarily extended by user-defined plug-ins. In contrast to the other plug-in types (propagation model and surface

plot plug-ins), the post-processing plug-ins must be stand-alone applications, i.e. executables. A comfortable way to implement such plug-ins is to write them as MATLAB m-files and then compile them using the MATLAB compiler.

MATLAB stand-alone plug-ins are independent executables and can thus be used in RPS as is. However, in order to implement and compile the underlying m-files, MATLAB is needed which can be purchased from The Mathworks Inc. ([www.mathworks.com](http://www.mathworks.com)) or through any local reseller.

### 6.8.1 Using Custom Post-Processing Plug-ins

Custom post-processing plug-ins allow for a further analysis of channel prediction results related to a certain position, along a path, or related to a transmitter. As an example, RPS provides a path analysis plug-in that presents the three-dimensional impulse response (over delay and path) along a given polyline as a MATLAB stand-alone application.

All post-processing plug-ins must be saved accordingly in one of the subfolders `rps_pathanalysis`, `rps_pointanalysis`, or `rps_txanalysis` in the plug-in directory in order to be recognized by RPS. All available post-processing plug-ins are listed in the Plugins tab of the corresponding dialog for point, path, or Tx analysis. For each plug-in, an analysis bandwidth can be specified. An example view of the provided path analysis plug-in is given in Fig. 6-30. If there are several plug-ins, they are all listed in these dialogs and can be selected there.

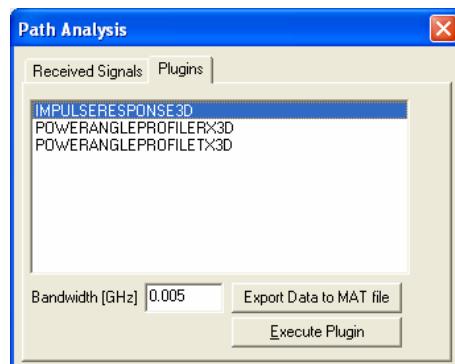


Fig. 6-30 User-defined post-processing plug-ins in an analysis dialog

The selected plug-in is executed by pressing the Execute Plugin button in the dialog. If there are several receivable transmitters for a point or path analysis plug-in, then for each of them a separate result window is generated. Fig. 6-31 presents sample results with the “IMPULSERESPONSE3D” plug-in which calculates the three-dimensional impulse response along a path.

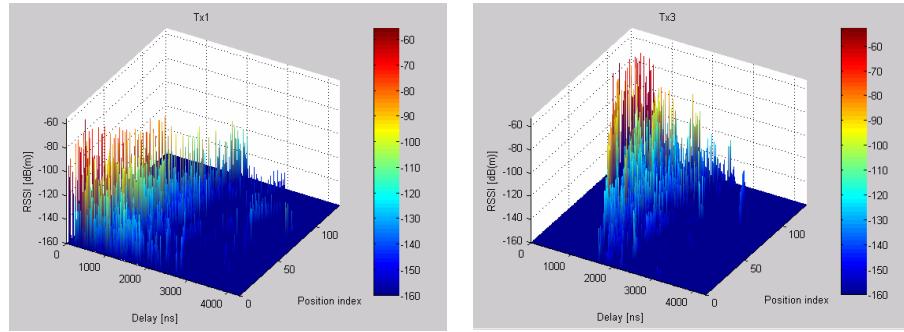


Fig. 6-31 Impulse responses along the same path, seen from two different transmitters

The channel data along the defined path could also be saved to a MATLAB file in a data structure. This function can be invoked by using the button [Export Data to MAT file](#) in the Plugin tab of the path analysis dialog as shown in Fig. 6-30 above. More details on this option can be found in Section 6.11.3.4.

## 6.8.2 Developing User-Defined Post-Processing Plug-ins

Post-processing plug-ins can translate the raw channel impulse response data into user-defined results. The usual way of implementation is to write a MATLAB m-file that is compiled then to a stand-alone application. The interface is very similar to the surface plot plug-in interface. In contrast to that, here the results can be presented in an arbitrary way, e.g. as MATLAB diagrams.

When a post-processing plug-in is executed, RPS saves a `data.mat` file that is stored in the plug-in directory first. Secondly, the plug-in is executed and the path and filename of the `data.mat` file is provided to the stand-alone plug-in as an argument. Within the `data.mat` file, a structure array with the name “RPS” is created that contains the information for the previously selected point, path, or transmitter according to Table 6-8.

Table 6-8 Plug-in interface for MATLAB stand-alone applications

| Parameter   | Description  |
|---|--|
| <b>General Parameters (N = number of available mobile stations)</b>         |  |
| RxPosition  | position(x [m], y [m], z [m]) of the current mobile station, 3 columns, N rows   |
| Frequency   | carrier frequency [GHz]  |
| Bandwidth   | bandwidth [GHz]  |
| NoiseFloor  | noise floor parameter [dBm] or unitless in dB, respectively  |
| <b>Parameters for base stations (N = number of available base stations)</b> |  |
| TxPosition  | vector of N Positions(x [m], y [m], z [m])   |
| TxPowerdB   | vector of N TxPower values [dBm] or unitless in [dB]   |
| TxName  | cell array with the names (strings) of the transmitters  |
| SelectedTxID  | An index for Tx analysis plug-ins that indicates which transmitter index (beginning from zero) was chosen. If SelectedTxID is equal to -1, a point analysis or path analysis dialog was chosen to create the <code>data.mat</code> file. |

| Parameter  | Description   |
|--|---|
| <b>Parameters for channel impulse responses (M = number of available multipath components; equal to the number of detected rays)</b> |   |
| Magnitude  | magnitude of the multipath  |
| Tau  | total excess delay [ns]   |
| Rx_PHI   | angle of arrival PHI at the mobile station [deg]  |
| Rx_THETA   | angle of arrival THETA at the mobile station [deg]  |
| Tx_PHI   | outgoing angle PHI at the base station [deg]  |
| Tx_THETA   | outgoing angle THETA at the base station [deg]  |
| TxID   | zero-based index of the transmitter ID (source of the multipath component)                      |
| IRCount  | Number of available items (columns) in this row. All other rows are initialized by a NaN value. |

The developer is responsible for using the contents in the `data.mat` file for any user-defined plots, charts, or tables created by the MATLAB stand-alone plug-in.

Due to the mentioned similarity, the reader is referred to the description of the surface plot plug-in interface for MATLAB m-files in Section 6.4.2.2.

## 6.9 Ray Path Analysis

In the previous Sections, the post-processing analysis of the ray impulse response data was described in detail. This Section is dedicated to the explanation of the graphical analysis of the ray paths in the 2D and 3D views. Note that a ray path analysis is only possible if the rays were saved during a simulation. This option can be selected in the simulation settings dialog, see Section 5.2.3.

The general presentation of the rays can be toggled using the icon in the standard toolbar. Rays are shown in different colors. The color of the rays can either depend on the pathloss or the delay. By default, the color is set according to the pathloss.

### 6.9.1 Inspecting Ray Paths

For a more detailed ray path analysis, it seems to be useful to only show the rays that impinge at a single receiver. This function is available in the 2D view by choosing the entries Show Rays – Color depends on Pathloss or Show Rays – Color depends on Delay, respectively, from a certain receiver's context menu. Then the rays can be inspected in both the 2D and the 3D views, colored according to the chosen option. An example for the ray paths impinging at a receiver is given in Fig. 6-32.

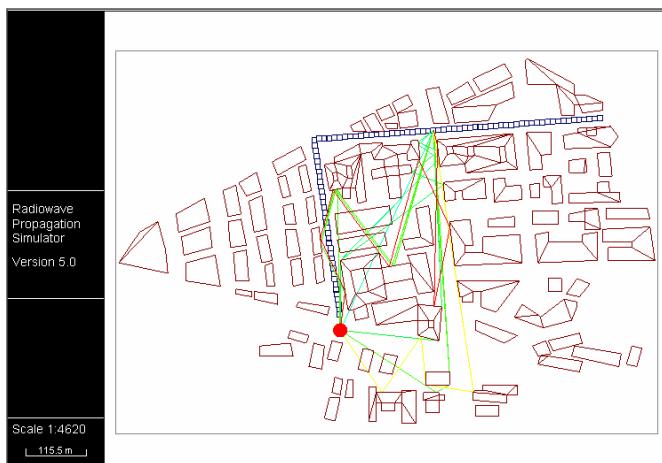


Fig. 6-32 Ray paths for a dedicated receiver position, colored according to their delays

In the latter case, the key navigation commands (see Section 3.6.4.3) help to easily move through the 3D environment. Moreover, the 3D camera position can be directly set to a receiver position by choosing Set Camera in 3D View from a receiver's context menu in the 2D view.

In order to return to the normal ray view mode where the rays at all receivers are shown, the View -> Ray Options -> Rays for all Receivers menu entry can be used.

### 6.9.2 Inspecting Launched Rays

For an in-depth DoA analysis at the transmitter, it is helpful to inspect the directions of outgoing rays in a 2D or 3D view. For this purpose it is possible to only show the first segment of the rays (i.e. those that are directly launched from a transmitter). To enable this option, the View -> Ray Options -> First Ray Segment menu entry must be checked.

Again, the analysis can be done in the 2D or 3D view. Especially for the 3D view, the camera position can be set to the position of the investigated transmitter by using the entry Set Camera in 3D View from that transmitter's context menu in the 2D view or in the Configuration tab of the tree window. Since then the camera is exactly in the center of the transmitter, the perspective should possibly be turned appropriately and also be moved slightly back using the key navigation commands (Cursor Down ↓). Furthermore, the presentation of the transmitter sphere should be switched off by toggling the icon in the standard toolbar. Otherwise the rays would be mostly disguised by the transmitter sphere itself. An example view of the resulting outgoing ray segments form a transmitter is shown in Fig. 6-33.

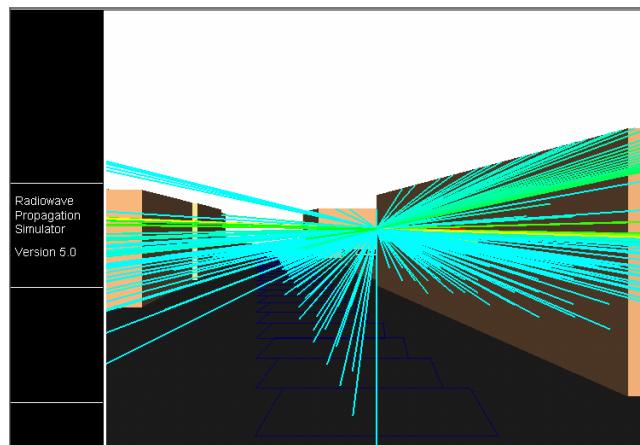


Fig. 6-33 Launched rays from a transmitter

Of course, the presentation of the first ray segments can also be combined with showing the rays for a single receiver only. Such a modification of the ray presentation could be reset by using the [View -> Ray Options -> Rays for all Receivers](#) menu entry. In order to return to the full ray presentation instead of the first ray segments only, the [View -> Ray Options -> First Ray Segment](#) menu entry must be clicked again.

#### 6.9.3 Inspecting Incident Rays

Likewise, the incident rays at a receiver can be inspected in detail in the 2D and 3D views. For this purpose, the last ray segments can be shown by selecting the [View -> Ray Options -> Last Ray Segment](#) menu entry. In order to show the incident rays at a dedicated receiver, this option can be combined with showing the rays for this receiver only as demonstrated in Section 6.9.1.

Also here, the analysis can be done both in the 2D or 3D view, respectively. Especially for the 3D view, the camera position can be set to the position of the investigated receiver by using the entry [Set Camera in 3D View](#) from that receiver's context menu in the 2D view. An example of a top view in the 3D view at an investigated receiver position is shown in Fig. 6-34.



Fig. 6-34 Top view of the incident rays at a receiver

In order to return to the full ray presentation instead of the last ray segments only, the [View -> Ray Options -> Last Ray Segment](#) menu entry must be clicked again.

## 6.10 Handling of Chart Graphs, Histograms, and Tables

Chart graphs, histograms, and tables represent an important instrument for the analysis of simulation results in RPS. They are shown in separate windows that can be resized and freely moved on the screen. Since they are non-modal, it is possible to have several of them open simultaneously which allows for an efficient comparison of results. Furthermore, the windows have a menu bar that provides several functions for modification and data export. The handling of these work elements are explained in the following.

### 6.10.1 Using Chart Graphs

RPS has sophisticated XY- and polar charts. The design, scaling, title, etc. can be adjusted by the user in a wide range. Chart graphs are presented in separate windows with an own menu bar. The functions available in the menus are described in Table 6-9.

Table 6-9 Functions of the menu in a chart graph window

| Menu Entry                 | Function  |
|----------------------------|---|
| <b>File Menu</b>           |   |
| Print                      | Opens the print dialog, see Subsection 6.10.1.5.  |
| Save as Image...           | Shows a file save dialog to save the diagram as an image.   |
| Close                      | Closes the chart graph window.  |
| <b>Edit Menu</b>           |   |
| Copy                       | Copies the chart to the Windows clipboard.  |
| <b>Graph Menu</b>          |   |
| Axis Scale and Settings... | Opens the axis scale and diagram settings dialog to modify the graph presentation, see Subsection 6.10.1.1. |
| Create Table from Graph    | Creates a table with two columns representing the x- and y values of the graph in a separate window.        |
| Reset to default styles    | Restores the default diagram settings.  |

#### 6.10.1.1 Chart Components

A chart consists of components. Each component can be modified separately using the axis scale and diagram settings dialog, accessible from the Graph -> Axis Scale and Settings... menu entry. Components are a *background* component, a *title*, the *display* (the graph itself), a *legend* and several *labels* commonly used as axis descriptors. The axis scale and diagram settings dialog is shown in Fig. 6-35.

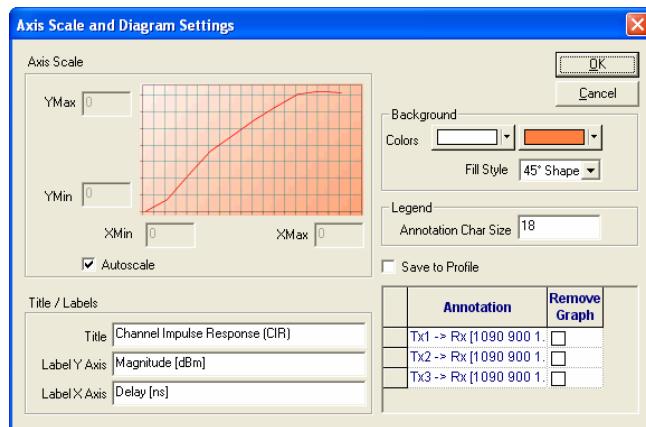


Fig. 6-35 Axis scale and diagram settings dialog

Any modification of the presentation style of chart graphs can be reset by choosing the menu entry Graph -> Reset to default styles.

#### 6.10.1.2 Scaling and Zooming

Commonly a graph is scaled automatically, thus assuring that all values are visible. It can be rescaled by the integrated zoom functions. Simply mark the area of interest in the graph with the left mouse button – and the chart is rescaled according to the selection as apparent from Fig. 6-36. This method also works in polar diagrams as, e.g., antenna patterns. Reverse zooming can be done by clicking the right mouse button.

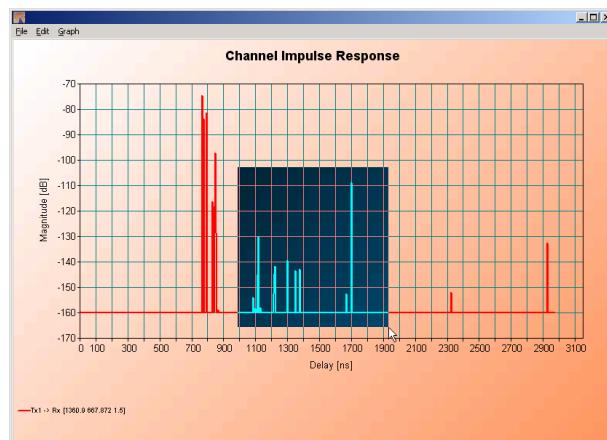


Fig. 6-36 Zooming in a chart

The axis scaling can also be set to dedicated values in the axis scaling and diagram settings dialog, see the previous Subsection.

#### 6.10.1.3 Creating a Table from a Graph

The x- and y values of a chart graph can be extracted into a table with two columns. This is accomplished by using the Graph -> Create Table from Graph menu entry. The table data can be further post-processed by means of the statistical evaluation functions available in tables, or in an external application.

#### 6.10.1.4 Exporting Chart Data

The graphical presentation of the diagram can be copied to the clipboard using the Edit -> Copy menu entry. Then it can be used in other applications. Furthermore, the diagram can be saved as an image by using the File -> Save as Image... menu entry. Possible file formats are \* .jpg and \* .dib (device independent bitmap).

Note that when exporting a diagram via the clipboard, the current presentation resolution of the chart is used. In order to achieve a high graphical resolution of the copied diagram, it is recommended to maximize the window before copying it.

Most of the data in diagrams can also be presented in tables. They allow a direct export of the data to ASCII files or via the clipboard to other applications.

#### 6.10.1.5 Printing a Chart Graph

In order to print a diagram, the File -> Print menu entry can be used. Then a page setup dialog appears where the printable size can be configured, refer to Fig. 6-37. After these settings have been made, the actual printer setup dialog is shown where the printer driver can be configured.

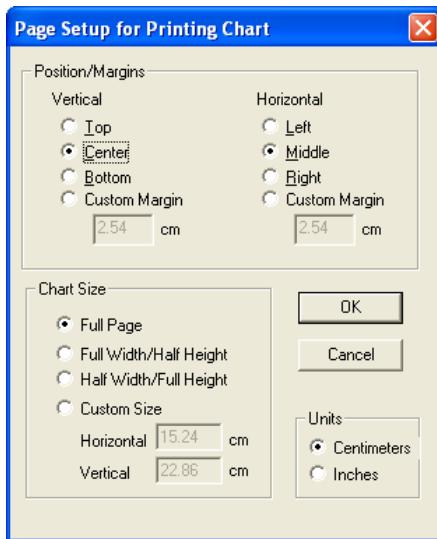


Fig. 6-37 Chart page setup dialog

### 6.10.2 Using Histograms

Histograms are important to statistically evaluate network performance data in order to obtain further results. Basically, a histogram represents the probability distribution of a parameter. Besides the actual histogram (quantized version of the probability density function), also the CDF as well as the mean value and the standard deviation of the selected parameter can be inspected.

Histograms are presented in separate windows with an own menu bar. The functions available in the menus are described in Table 6-10.

Table 6-10 Functions of the menu in a histogram window

| <i>Menu Entry</i>          | <i>Function</i>  |
|----------------------------|--|
| <b>File Menu</b>           |  |
| Print                      | Opens the print dialog.  |
| Save as Image...           | Shows a file save dialog to save the histogram as an image.  |
| Close                      | Closes the histogram window.   |
| <b>Edit Menu</b>           |  |
| Copy                       | Copies the histogram to the Windows clipboard.   |
| <b>Graph Menu</b>          |  |
| Histogram Settings...      | Shows the histogram settings dialog to modify bin size, limits, etc., see Subsection 6.10.2.2.                                 |
| Axis Scale and Settings... | Opens the axis scale and diagram settings dialog to modify the graph presentation.   |
| Create Table from Graph    | Creates a table with three columns representing the x values and the PDF and CDF values of the histogram in a separate window. |
| Reset to default styles    | Restores the default histogram settings.   |

#### 6.10.2.1 Creating a Table from a Histogram

The x values as well as the PDF and CDF values of a histogram can be extracted into a table with three columns. This is accomplished by using the Graph -> Create Table from Graph menu entry. The table data can be further post-processed by means of the statistical evaluation functions available in a table (refer to Subsection 6.10.3.2), or in an external application.

#### 6.10.2.2 Customization of Histograms

The initial scaling of a histogram is determined automatically. These settings can also be modified afterwards in the histogram settings dialog as shown in Fig. 6-38. Here the histogram limits and the bin resolution (either in terms of a bin size or the number of bins) can be specified. If the option Skip Values outside Limits is selected, all values out of the configured limits are not considered in the creation of the histogram. Additionally, the visibility of the CDF, the mean value, and the standard deviation can be controlled. If the option Mean and StdDev. as Logarithmic Values (dB / dBm) is chosen, the according characteristics are calculated based on logarithmic values, typically given in dB.

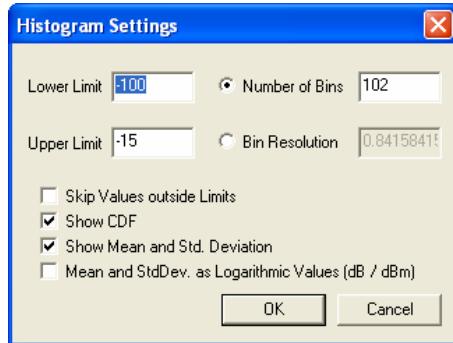


Fig. 6-38 Histogram settings dialog

Furthermore, using the axis scale dialog can be used to set the axis limits manually in a way similar to chart graphs. This dialog can be opened using the Graph -> Axis Scale and Settings... item from the menu bar. In a separate histogram window, this menu entry is also available. See Subsection 6.10.1.1 for the usage of this dialog.

#### 6.10.2.3 Printing a Histogram

Histograms can also be printed. The size and the orientation of the histogram on the page can be adjusted using the page setup dialog, accessible from the Graph -> Page Setup... menu item. In separate histogram windows the same dialog can be accessed from File -> Print. The usage is comparable to the page setup for chart graphs, see Subsection 6.10.1.5 for details.

#### 6.10.3 Using Tables

Tables contain data in tabular form. The rows and columns can be resized, single table fields or groups of fields can be selected and copied to the clipboard. Such data can be efficiently post-processed in other applications as, e.g., MS Excel or MATLAB. An example of an impulse response in tabular form is given in Fig. 6-39.

|    | Tx Name | Tau [ns] | Magnitude [dB] | Phase [°] | Tx Phi [°] | Tx Theta [°] | Rx Phi [°] | Rx Theta [°] |
|----|---------|----------|----------------|-----------|------------|--------------|------------|--------------|
| 1  | Tx1     | 766.258  | -79.70947      | 95.72327  | 352.99     | 2            | 172.99     | -2           |
| 2  | Tx1     | 766.744  | -81.80833      | 91.09448  | 352.99     | 3            | 172.99     | 3            |
| 3  | Tx1     | 777.969  | -83.79742      | 10.34628  | 358.99     | 3            | 158.39     | 3            |
| 4  | Tx1     | 798.759  | -86.26872      | -90.77863 | 5.99       | 2            | 192.514    | -2           |
| 5  | Tx1     | 799.212  | -88.35415      | -107.2388 | 5.99       | 3            | 192.514    | 3            |
| 6  | Tx1     | 830.603  | -121.1257      | -4.908373 | 36.99      | 2            | 167.028    | -2           |
| 7  | Tx1     | 831.07   | -123.2545      | -16.43164 | 36.99      | 3            | 167.028    | 3            |
| 8  | Tx1     | 835.467  | -148.3358      | -101.7324 | 36.99      | 3            | 158.689    | 3            |
| 9  | Tx1     | 848.603  | -123.2259      | 172.2681  | 36.99      | 2            | 189.551    | -2           |
| 10 | Tx1     | 849.08   | -125.3382      | 164.4296  | 36.99      | 3            | 189.551    | 3            |
| 11 | Tx1     | 850.829  | -97.19217      | -66.61076 | 13.99      | 3            | 145.236    | 3            |
| 12 | Tx1     | 866.173  | -158.8162      | 145.194   | 330.99     | -6           | 218.474    | -4.00001     |
| 13 | Tx1     | 1085.02  | -157.1149      | 89.00015  | 68.99      | 1            | 167.028    | -1           |
| 14 | Tx1     | 1085.32  | -159.5541      | 17.92389  | 68.99      | 2            | 167.028    | 2            |
| 15 | Tx1     | 1103.19  | -158.4181      | -29.95956 | 68.99      | 1            | 189.551    | -1           |
| 16 | Tx1     | 1119.47  | -129.9768      | -55.09519 | 356.99     | 2            | 37.7403    | 1.75         |
| 17 | Tx1     | 1134.81  | -158.1304      | 19.30088  | 297.99     | 1            | 218.566    | -1           |
| 18 | Tx1     | 1219.95  | -144.9818      | -40.00814 | 335.99     | 2            | 132.22     | 1.75         |
| 19 | Tx1     | 1221.69  | -141.625       | -159.3329 | 36.99      | 1            | 37.9207    | -1.25        |
| 20 | Tx1     | 1300.21  | -139.3355      | 18.36791  | 358.99     | 1            | 68.0759    | -1           |
| 21 | Tx1     | 1352.98  | -143.5513      | -65.64626 | 7.99       | 1            | 68.0759    | -1.25        |
| 22 | Tx1     | 1375.59  | -142.885       | -153.7328 | 335.99     | 2            | 116.021    | 1.75         |
| 23 | Tx1     | 1671.03  | -152.4879      | 134.7294  | 185.99     | 0            | 193.237    | -1.25001     |
| 24 | Tx1     | 1700.95  | -109.5851      | -27.3327  | 30.99      | 1            | 165.222    | -0.75        |
| 25 | Tx1     | 1701.14  | -110.4525      | -137.3491 | 30.99      | 1            | 165.222    | 1.25         |
| 26 | Tx1     | 2326.03  | -152.0394      | -37.5718  | 117.99     | 1            | 196.345    | -0.625       |
| 27 | Tx1     | 2928.37  | -133.8832      | -66.8375  | 359.99     | 1            | 198.685    | -0.625       |
| 28 | Tx1     | 2928.6   | -136.4318      | -165.0244 | 359.99     | 1            | 198.685    | 0.875        |

Fig. 6-39 An impulse response in tabular form

Tables are presented in separate windows with an own menu bar. The functions available in the menus are described in Table 6-11.

Table 6-11 Functions of the menu in a table window

| <i>Menu Entry</i> | <i>Function</i>  |
|-------------------|--|
| <b>File Menu</b>  |  |
| Close             | Closes the table window.   |
| <b>Edit Menu</b>  |  |
| Copy Table        | Copies the table data to the Windows clipboard.                      |
| Find...           | Opens a find dialog for the current selection in the table.          |
| Replace...        | Opens a replace dialog valid for the current selection in the table. |

#### 6.10.3.1 Exporting Table Data

The complete table data can be copied to the clipboard by simply using the Edit -> Copy Table menu entry of the table window. If only a part of the table shall be copied, the respective part must be selected first with the mouse. Then the data can be copied by pressing the <Ctrl-C> key combination. The copied data can then be further analyzed in other applications. The BIFF-8 format (MS Excel 97/2000) is supported that translates locale differences, in particular the decimal delimiter (,” or “.” problem).

#### 6.10.3.2 Statistical Data Evaluation in Tables

Numerical values in tables can be evaluated in various ways. The data evaluation functions are accessible from the context menu of a table field. In order to use the functions, first a relevant selection of values in the table must be made. Then the chosen evaluation function is applied to the selected data. The result value is shown in a dialog. The workflow of statistically evaluating a table selection is demonstrated in Fig. 6-40. The available table evaluation functions are described in Table 6-12.

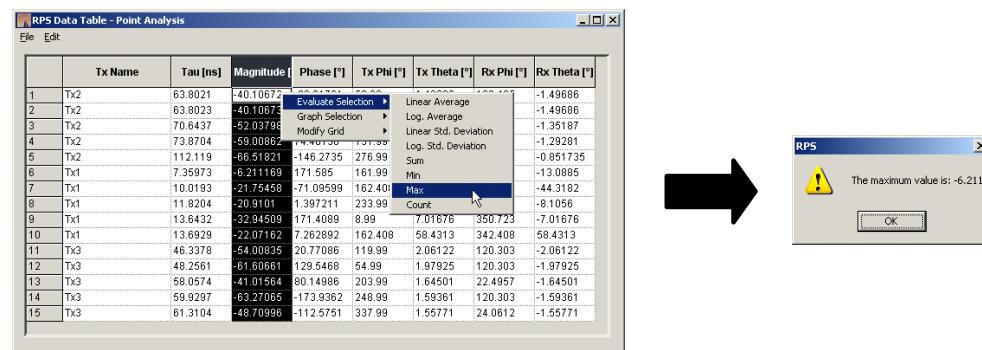


Fig. 6-40 Statistical evaluation of numerical table data

Table 6-12 Statistical evaluation functions in a table

| <i>Context Menu Entry</i> | <i>Function</i>   |
|---------------------------|---|
| <b>Evaluate Selection</b> |   |
| Linear Average            | Calculates the linear average of the selected numerical values and shows the result in a dialog.  |
| Log. Average              | Calculates the logarithmic average of the selected numerical values and shows the result in a dialog. This function should be used for values given in dB.            |
| Linear Std. Deviation     | Calculates the linear standard deviation of the selected numerical values and shows the result in a dialog.   |
| Log. Std. Deviation       | Calculates the logarithmic standard deviation of the selected numerical values and shows the result in a dialog. This function should be used for values given in dB. |
| Sum                       | Calculates the sum of the selected numerical values and shows the result in a dialog.   |
| Min                       | Determines the minimum of the selected numerical values and shows the result in a dialog.   |
| Max                       | Determines the maximum of the selected numerical values and shows the result in a dialog.   |
| Count                     | Determines the number of the selected numerical values and shows the result in a dialog.  |

#### 6.10.3.3 Transforming Table Data into Graphical Presentations

Selected values in a table can be transformed into graphical presentations as chart graph, histogram, or surface plot by using the entries in the Graph Selection submenu of the table grid context menu. In case of a diagram/histogram, a separate window is opened with the chart or histogram; in case of a surface plot, a new result layer is created. The available presentation functions are described in Table 6-13.

Table 6-13 Graphical presentation functions from a table

| <i>Context Menu Entry</i> | <i>Function</i>  |
|---------------------------|--|
| <b>Graph Selection</b>    |  |
| Chart View                | Shows a curve of a selected column of numerical values vs. their index in a separate chart graph window. |
| Histogram                 | Determines the histogram of a selected column of numerical values in a separate histogram window.        |

The workflow of creating a graphical presentation from a table selection is demonstrated in Fig. 6-41 with a histogram as example.

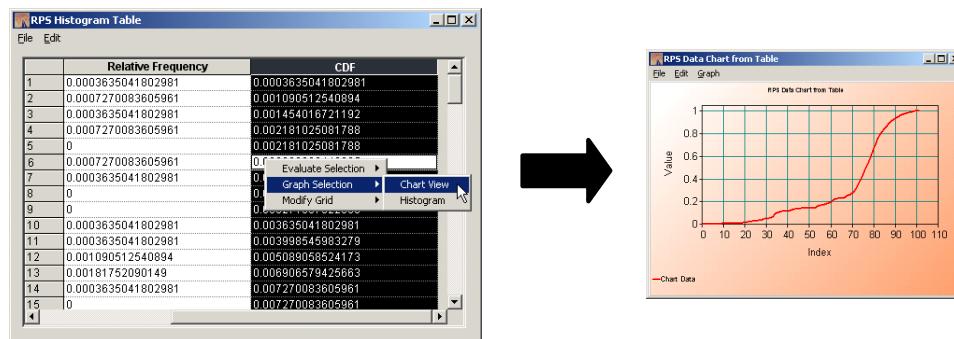


Fig. 6-41 Graphical presentation of numerical table data

#### 6.10.3.4 Customization of Tables

The initial scaling of the rows and columns in a table is determined automatically. However, sometimes a column may be too narrow thus hiding a part of the field contents, or a multi-line head row of a table only shows a part of the header. That is why the width of table rows and columns can be modified afterwards by simply dragging the border lines between rows/columns with the mouse pointer.

The data in the table can be ordered with respect to a certain row or column. This is accomplished by double-clicking the respective header field of the row/column. Note that such an ordering operation generally causes a permutation of columns or rows.

For flexible table handling, rows and columns can be inserted or deleted. These functions are available from the Modify Grid submenu in the table context menu. Their functionality is self-explanatory.

In order to modify the data in a table, Find and Replace functions are available. These features can be initiated from the Edit menu of a table window and are applied to the current selection in the table. If there is no active selection, these functions are applied to the entire table.

## 6.11 Data Import and Export

RPS supports many data export and import formats that enable a convenient way for setting up an investigation scenario and saving the simulation results in external data files for a further post-processing in other tools. The comprehensive data import and export capabilities incl. the supported data formats are described in this Section.

### 6.11.1 Import of Antenna Diagrams

An antenna can only be assigned to a transmitter or to the available receivers, if the antenna has been imported into the project before. All available antennas in the project are represented in the tree view. One or more antennas can be imported in one step by using the menu item File -> Import -> Import Antenna... or by clicking on the Antenna root item in the tree view with the right mouse button and using the entry in the appearing context menu.

Next, the available antenna header files are shown in a file selection dialog. One or more \*.cfg files can be chosen as apparent from the example dialog in Fig. 6-42.

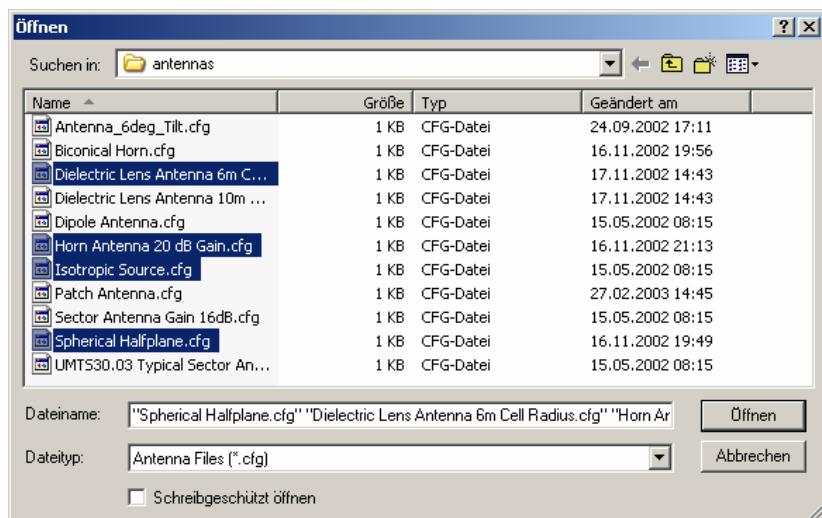


Fig. 6-42 Antenna import file selection dialog

RPS adds all the selected antenna patterns to the project in one step finally.

A description of the antenna file format can be found in Appendix 10.6.

Note: The import of an antenna does not automatically mean that the antenna is really used in simulations. Available antennas in the project can be assigned to one or more transmitters or to all receivers. The relative orientation of the antenna is configurable for each transmitter separately, so that the antennas can be shared by several transmitters.

### 6.11.2 Import and Export of Environment Data

RPS has a built-in environment editor that can handle complete AutoCAD drawings which guarantees the most advanced compatibility in importing, exporting, or modifying vector drawing projects. The original DXF and DWG AutoCAD formats were published by Autodesk. They can represent various geometrical shapes. In the next Subsections, only those types of supported formats and the types of building entities that can be handled by the ray tracer will be discussed.

#### 6.11.2.1 Importing DXF/DWG Files

RPS can import objects from Autodesk DXF or DWG files. DXF is the abbreviation of *Drawing Exchange Format* and it is widely supported by many CAD applications. The following file format versions are supported:

- DWG files from AutoCAD release 2.5 to 2000,
- DXF files from AutoCAD release 10 to 2000.

The environment editor can handle all common types of entities inclusive blocks. Since the ray tracing engine requires a 3D environment, only three-dimensional entities will be transferred to the ray tracing engine internally. The following entities – also defined within blocks – are supported:

- 3D faces,

- polyfaces,
- blocks, and
- inserts.

Polylines and other 3D entities can easily be converted into 3D faces with the rich editing capabilities of the environment editor.

A DXF or DWG file can be imported with the File -> Import -> Import DXF or DWG environment data menu item. All existing data will be removed from the environment, if a new environment is imported.

Note: If an environment shall be added to an existing environment without deleting existing building data, the Insert Block command can be used as an alternative. The new DXF or DWG file that has been imported as a block can be exploded then to enable the editing of the previously imported environment.

#### 6.11.2.2 Import Environment Data from OBL Files

RPS supports an option to import environment data from OBL files. The name OBL is derived from Object List. It can be used to import 3D polygons that can have references to several different material layers. The OBL format specification is given in Appendix 10.5.

For the import of environment data from an OBL file the menu item File -> Import -> Import Object List... is available.

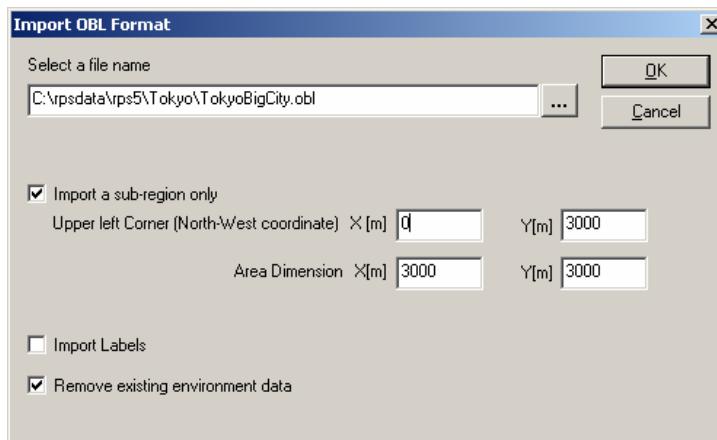


Fig. 6-43 Import OBL Format Dialog

In the appearing dialog as shown in Fig. 6-43 a filename must be chosen first. Optionally, it is possible to define a sub-region to limit the import of polygons to a part of the environment data. If this feature is switched on by activating the check box Import a sub-region only, the upper left corner (North-West corner) of a rectangular area and its extents in X and Y directions given in meters must be entered in Cartesian coordinates (projection coordinate system, see Section 3.2).

Finally, it can be chosen if the label fields shall be imported together with the polygons, and if already existing data shall be removed from the environment first by switching on or off the respective check boxes available in the lower part of the dialog.

After confirmation, the data will be imported and will be converted into entities (polyfaces and 3D faces). The progress is shown in the status bar. Any line that cannot be imported will be presented in the message window. The import process can be canceled by pressing the <ESC> key.

#### *6.11.2.3 Exporting Objects to DXF or DWG Files*

The environment data can easily be exported into DWG or DXF files. The following file format versions are supported:

- DWG files from AutoCAD release 2.5 to 2000,
- DXF files from AutoCAD release 10 to 2000.

The file format can be chosen in the file save dialog that can be opened by using the File -> Export -> Export Environment Data as DXF or DWG menu item from the file type selection box.

#### *6.11.2.4 Exporting Objects to OBL Files*

Three-dimensional vertices of polygons supported by the ray tracing engine can also be stored into the proprietary OBL file format. The menu item File -> Export -> Export Environment Data as Object List opens a common file save dialog that enables to define a directory and a file name where the data shall be saved.

Note that only those objects are stored in the OBL file the material layers of which are activated. Objects of inactive layers are ignored during the OBL export.

### **6.11.3 Exporting Simulation Results**

Results of a simulation can be stored in several tab-delimited ASCII file formats as well as in binary \*.MAT files readable with MATLAB for further post-processing and data analysis. This enables, e.g., creating statistics about angular distributions, power distributions, or other data processing exceeding the integrated analysis functions of RPS.

Zero power values (magnitudes) cannot be converted into logarithmic dB-values by default because the  $\log_{10}(0)$  is -INF. RPS replaces all non-numerical values with a not-a-number identifier “NaN” in ASCII files.

#### *6.11.3.1 Exporting Received Power and Delay Spread*

The function File -> Export -> Export Received Power and Delay Spread generates an ASCII data file where each receiver position is associated with the wideband received power and the delay spread as a total value and per transmitter:

---

```
%Data generated with RPS
%Number of Dataset Records = 6138
%Center Frequency = 1 GHz
%Bandwidth = 0.2 GHz
```

| %X[m] | Y[m] | Z[m] | TotalReceivedPower [dBm] | DelaySpread[ns] | PowerFromTx 'Tx1' | DelaySpreadFromTx 'Tx1' | ... |
|-------|------|------|--------------------------|-----------------|-------------------|-------------------------|-----|
| 0.00  | 0.00 | 1.5  | -100.99                  | 0.00            | NaN               | 0.00                    | ... |
| 0.00  | 0.50 | 1.5  | -100.87                  | 0.00            | NaN               | 0.00                    | ... |
| 0.00  | 1.00 | 1.5  | -100.98                  | 0.00            | NaN               | 0.00                    | ... |
| 0.50  | 0.00 | 1.5  | -101.20                  | 0.00            | NaN               | 0.00                    | ... |
| 0.50  | 0.50 | 1.5  | -100.82                  | 0.00            | NaN               | 0.00                    | ... |
| 0.50  | 1.00 | 1.5  | -100.94                  | 0.00            | NaN               | 0.00                    | ... |
| .     | .    | .    | .                        | .               | .                 | .                       | .   |
| .     | .    | .    | .                        | .               | .                 | .                       | .   |
| .     | .    | .    | .                        | .               | .                 | .                       | .   |

---

The first lines beginning with the character '%' contain comments about the amount of data records, center frequency etc. Below, the total received power and the delay spread calculated from signals coming from all transmitters, and the received power and delay spread value calculated for each transmitter signal separately are given for each receiver position.

#### 6.11.3.2 Exporting Data as an ASCII File

The export function **File -> Export -> Export Data as ASCII-File** is useful if the complete raw data of each multi-path component arrived at any receiver is required. Each line represents a multi-path component (ray path) that has arrived at a receiver position given in the comment line above the data:

| %Data generated with RPS            |        |                 |          |           |             |           |             |
|-------------------------------------|--------|-----------------|----------|-----------|-------------|-----------|-------------|
| %Tx@X=1390m, Y=440m, Z=8m, BSID=Tx1 |        |                 |          |           |             |           |             |
| %Tx@X=1770m, Y=890m, Z=8m, BSID=Tx2 |        |                 |          |           |             |           |             |
| %Tx@X=1930m, Y=520m, Z=8m, BSID=Tx3 |        |                 |          |           |             |           |             |
| %Number of Dataset Records = 1      |        |                 |          |           |             |           |             |
| %Center Frequency = 1.5 GHz         |        |                 |          |           |             |           |             |
| %Rx@X=8.00m, Y=5.00m, Z=1.50m       |        |                 |          |           |             |           |             |
| %Tx                                 | TAU    | Magnitude [dBm] | Phase[°] | PHI@Tx[°] | THETA@Tx[°] | PHI@Rx[°] | THETA@Rx[°] |
| Tx1                                 | 167.35 | -106.94         | -156.70  | 87.20     | 91.05       | 353.00    | 90.46       |
| Tx1                                 | 27.65  | -80.08          | -109.70  | 272.01    | 96.66       | 151.16    | 108.27      |
| Tx2                                 | 129.84 | -109.07         | 78.10    | 142.86    | 87.12       | 351.00    | 90.61       |
| .                                   | .      | .               | .        | .         | .           | .         | .           |
| .                                   | .      | .               | .        | .         | .           | .         | .           |
| .                                   | .      | .               | .        | .         | .           | .         | .           |

---

In the body of the data file, each comment line beginning with the character '%' indicates the start of a new Rx section, containing the full impulse response from all transmitters at the respective receiver. Below this comment lines, the name of the transmitter source, absolute delay, magnitude, phase and angles at the receiver, as well as the outgoing angles from the transmitter are saved.

#### 6.11.3.3 Exporting DoA as an ASCII File

For computations based on the angle of arrival (AoA is similar to the widely used abbreviation DoA, direction of arrival) distribution, this export function provides a cumulative storage of the angular distribution of angles of arrival for uplink and downlink analysis:

---

```
%Data generated with RPS
%Number of Dataset Records = 1
%Center Frequency = 1.500 GHz
%Rx@X=8.00m,Y=5.00m,Z=1.50m
```

| %PHI [ ° ] | Power@Rx[dB/dBm] | Power@Tx[dB/dBm] (uplink) |
|------------|------------------|---------------------------|
| 0          | NaN              | -52.65                    |
| 1          | NaN              | NaN                       |
| 2          | NaN              | NaN                       |
| 3          | NaN              | NaN                       |
| 4          | -99.49           | NaN                       |
| 5          | NaN              | NaN                       |
| 6          | -86.96           | NaN                       |
| 7          | NaN              | NaN                       |
| 8          | -69.47           | NaN                       |
| 9          | -88.41           | NaN                       |
| .          | .                | .                         |
| .          | .                | .                         |
| .          | .                | .                         |

---

For each discrete angle from 0° to 360°, the total received power for downlink (angles at the receiver) and for uplink (angles at the transmitter) are saved. The file pattern is stored for each receiver position separately. Fields where no signals are available will be filled with NaN's.

#### 6.11.3.4 Exporting Path Analysis Results into a MATLAB file

The results of a path analysis can be stored into a MATLAB file. After having defined a polyline for path analysis and specified the capture radius for it as described in Section 6.6, the path analysis dialog appears which is shown in Fig. 6-44 with its Plugin tab for convenience.

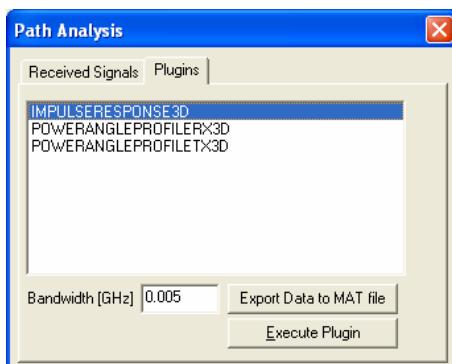


Fig. 6-44 Plugin tab of the path analysis dialog

The raw data along the defined path can be stored into a MATLAB file by clicking the button Export Data to MAT file. In the appearing file selector, the filename of the MATLAB file has to be specified.

In the resulting MATLAB file a struct array called 'RPS' is saved. It has the following contents:

- Frequency: The default center frequency in GHz.

- Bandwidth: The default bandwidth in GHz.
- NoiseFloor: The noise floor in dBm.
- RxPosition: A matrix with all Rx positions.
- TxPosition: A matrix with all Tx positions.
- TxPowerdB: The transmit powers of all Tx's given in dBm.
- TxName: A row vector containing all Tx names.
- SelectedTxID: [not used]
- IR: A structure containing all impulse responses with the following items:
  - Magnitude: A matrix of the magnitudes of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - Tau: A matrix of the path delays of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - Rx\_PHI: A matrix of the azimuth angles at the Rx of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - Rx\_THETA: A matrix of the elevation angles at the Rx of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - Tx\_PHI: A matrix of the azimuth angles at the Tx of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - Tx\_THETA: A matrix of the elevation angles at the Tx of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - TxID: A matrix of the Tx IDs per path of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.
  - IRCount: A column vector of the number of paths of all impulse responses (at each receiver) with a number of bins determined by the selection of the bandwidth.

#### 6.11.4 Exporting Static Channel Data

The channel data calculated for each receiver position can also be stored into matrices. These matrices can be stored into a set of ASCII files, or into one MATLAB file, respectively.

#### 6.11.4.1 Static Channel Data Export into ASCII Matrices

The menu command File -> Export -> Export Static Data as ASCII-Matrices in Multiple Files can be used to store a set of matrices into several ASCII files. Each matrix is stored into a separate file which helps to simplify the post-processing in most cases. The names of these files are derived from a root name that is configurable in the File Save As Dialog similar to Fig. 6-45. In the lower right part a Bandwidth given in GHz and a Number of Samples / IR can be specified. The bandwidth defines the temporal resolution of the data stored in the files. The number of samples per impulse response declares the maximum delay of multipath components stored in the matrices. The maximum delay given in ns can easily be calculated by dividing the number of samples by the bandwidth. The bandwidth parameter that can be entered here is only used temporarily for the data export.

After confirmation RPS creates the files with the following contents, as specified in Table 6-14.

Table 6-14 Matrix files generated by RPS

| <i>File Name:</i>  | <i>File Contents</i>   |
|--|--|
| <i>XXX</i> is the root name  |  |
| <i>YYY</i> is the TxID, if the number<br>of transmitters is greater than<br>one, ( <i>YYY</i> = <i>Tx0</i> , <i>Tx1</i> , ...) |  |
| XXX.ascii  | This file contains comments about simulation parameters, for instance center frequency, bandwidth, noise floor, etc. It is not readable by MATLAB.                         |
| XXX_real_YYY.ascii   | This file contains the real parts of the complex magnitudes of the received signals.   |
| XXX_imag_YYY.ascii   | This file contains the imaginary parts of the complex magnitudes of the received signals.  |
| XXX_tx_name_YYY.ascii  | This file contains the names of all available transmitters.  |
| XXX_numpaths_YYY.ascii   | This file contains the number of resolvable paths. The values depend on the bandwidth.   |
| XXX_rx_position_YYY.ascii  | This file contains the positions for all receivers.  |
| XXX_tx_position_YYY.ascii  | This file contains the positions of all transmitters.  |
| XXX_tx_phi_YYY.ascii   | This file contains the azimuth angular information for rays at the transmitter for all receiver positions.   |
| XXX_rx_phi_YYY.ascii   | This file contains the azimuth angular information for rays at the receivers.  |
| XXX_tx_theta_YYY.ascii   | This file contains the elevation angular information for rays at the transmitter for all receiver positions, the angles are in absolute values (0 deg = horizontal plane). |
| XXX_rx_theta_YYY.ascii   | This file contains the elevation angular information for rays at the receivers, the angles are in absolute values (0 deg = horizontal plane).                              |

If the number of transmitters within the scenario is greater than one, for each transmitter a separate matrix is generated. The file names are extended by the suffix "\_Tx0", "\_Tx1" and so on.

#### 6.11.4.2 Static Channel Data Export into a MATLAB File

The simulation results can also be stored as matrices into binary \*.MAT files. MATLAB can read these files and the data can be processed very conveniently and efficiently then.

RPS has incorporated export filters to create MAT files directly. Both the MATLAB V4 and the newer format that was introduced from MATLAB Release 11 (V5/V6) can be generated. A MATLAB Data Export dialog can be opened with the menu command File -> Export -> Export static data as matrices readable with Matlab(TM). In this dialog the target file name as well as some further options can be configured. First of all, it can be chosen if the matrices that can potentially contain many zeros will be created as normal matrices or as sparse matrices. The latter opportunity can reduce the memory and file size necessary for creating and storing the matrices significantly. Further, MATLAB has the capability to handle sparse matrices directly in a very efficient manner. Therefore, it is recommended to use the Store Data into Sparse Matrices option. Secondly, an option is available to save the matrices in an old \*.MAT file format that can be read also with MATLAB V4.

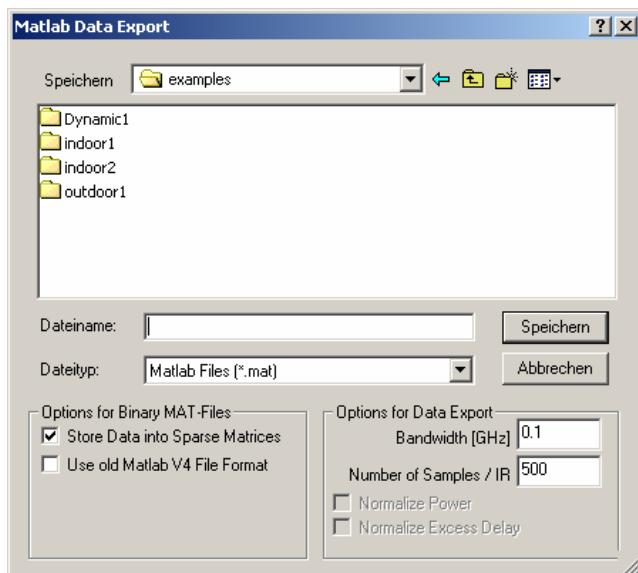


Fig. 6-45 Options in the matrix file export dialog

In the lower right part of the MATLAB export dialog, a Bandwidth given in GHz and a Number of Samples / Impulse Response can be specified. The bandwidth defines the temporal resolution of the data stored in the MATLAB file. The number of samples per impulse response declares the maximum delay of multipath components stored in the matrices. The maximum delay given in ns can easily be calculated by dividing the number of samples by the bandwidth. The bandwidth parameter that can be entered here is only used temporarily for the data export.

After confirmation a MAT file will be created with the following matrices inside as apparent from Table 6-15.

Table 6-15 Matrix names stored into the MAT file

| Matrix Name                   | Values  |
|-------------------------------|---|
| RPS_Magnitude(N, M)           | complex magnitudes of the channel impulse responses<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response<br>Max. Excess Delay [ns]: M/(RPS_Bandwidth [GHz])<br><br>Notice: Values are linear complex magnitudes ( $a_{dB} = 20\log(a)$ ). |
| RPS_NumResolvable Paths(N, 1) | Number of resolvable paths (depends on the bandwidth of the radio channel)<br>N: Total number of receiver positions   |
| RPS_Rx_Position(N, 3)         | Position [X Y Z] of each receiver position<br>N: Total number of receiver positions   |
| RPS_Tx_Position(T, 3)         | Position [X Y Z] of each transmitter position<br>T: Total number of transmitters  |
| RPS_Rx_PHI(N, M)              | Azimuth angles of paths in degree [0...360°] at receivers<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response  |
| RPS_Tx_PHI(N, M)              | Azimuth angles of paths in degree [0...360°] at transmitters<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response   |
| RPS_Rx_THETA(N, M)            | Elevation angles of paths in degree [-90...90°] at receivers<br>N: Total number of positions<br>M: Number of samples for each impulse response  |
| RPS_Tx_THETA(N, M)            | Elevation angles of paths in degree [-90...90°] at transmitters<br>N: Total number of positions<br>M: Number of samples for each impulse response   |
| RPS_Tx_Names(T)               | Names of the transmitters as a cell array<br>T: Total number of transmitters  |
| RPS_CenterFrequency           | Center frequency [GHz]  |
| RPS_Bandwidth                 | Bandwidth [GHz]   |
| RPS_NoiseFloor                | Noise Floor [dB]  |

If the number of transmitters within the scenario is greater than one, for each transmitter a separate matrix is generated. The names are extended by the suffix "\_Tx0", "\_Tx1" and so on.

#### 6.11.4.3 Data Post-Processing in MATLAB

The distribution of RPS provides a set of m-files that shall demonstrate the import and graphical representation of result data in MATLAB. This shall help for getting started with RPS data post-processing in MATLAB. The scripts can be used for loading the data from both ASCII and \*.MAT data files. These sample m-files can be found in the `matlab_scripts` directory in the RPS installation directory. Note that the lines between "Begin ASCII-Files import" and "End of ASCII Import" must be commented out if the ASCII files shall be used as data sources.

Fig. 6-46 to Fig. 6-49 show sample results generated with MATLAB and the provided m-files for static result analysis.

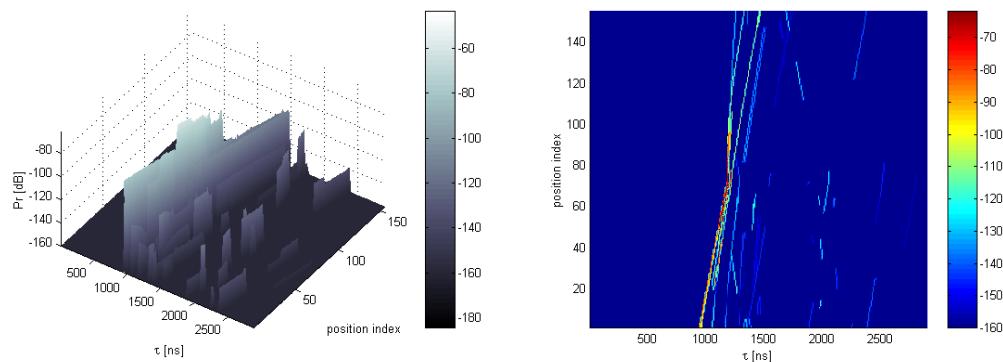


Fig. 6-46 Channel impulse response generated with RPS and visualized using the MATLAB interface

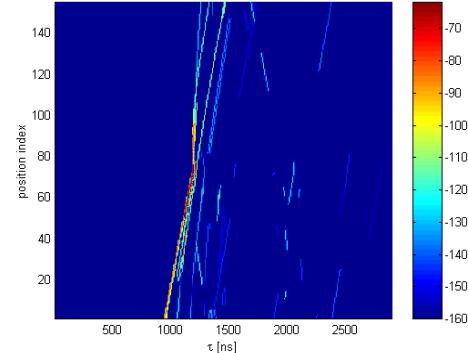


Fig. 6-47 Footprint of the channel impulse response shown in Fig. 6-46

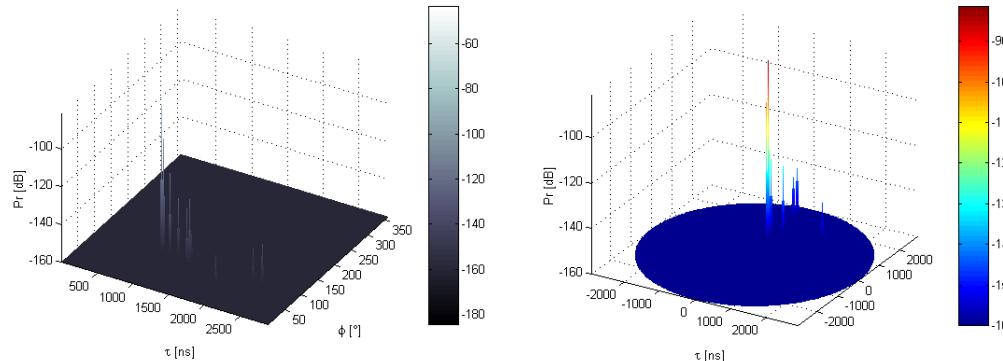


Fig. 6-48 Angles of arrival for a selected position in Cartesian view

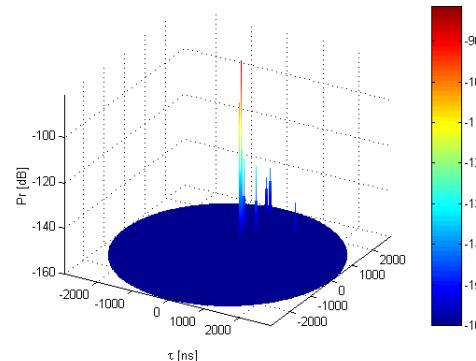


Fig. 6-49 Directions of arrival for a selected position in angular view

### 6.11.5 Exporting Dynamic Channel Data

Channel data that has been obtained by dynamic simulations can also be stored into ASCII and MATLAB files. The storage procedure is similar to the saving of static data as described in the previous Section. Thus, only the differences to the static data storage method are pointed out in the following Sub-sections.

#### 6.11.5.1 Dynamic Channel Data Export as ASCII Matrices

The menu command File -> Export -> Export Dynamic Data as ASCII-Matrices in Multiple Files can be used to create a set of ASCII files. The names of these files are derived from a root name, which is defined in the file save dialog. RPS then generates the files as specified in Table 6-16.

Table 6-16 ASCII files with dynamic data generated by RPS

| <i>Filename:<br/>XXX is the root name</i> | <i>File Contents</i>  |
|---|---|
| XXX.ascii                                 | Comments on simulation parameters, for instance center frequency, bandwidth, noise floor etc. This file is not readable by MATLAB.  |
| XXX_real.ascii                            | Real parts of the complex magnitudes of the received signals.   |
| XXX_imag.ascii                            | Imaginary parts of the complex magnitudes of the received signals.  |
| XXX_numpaths.ascii                        | Number of resolvable paths.   |
| XXX_rx_position.ascii                     | Positions [x, y, z] of the receivers for each matrix row  |
| XXX_tx_name.ascii                         | This file contains the names of all available transmitters.   |
| XXX_tx_position.ascii                     | Positions [x, y, z] of the transmitters   |
| XXX_tx_phi.ascii                          | Azimuth angular information for rays launched from the transmitters for all receiver positions; the angles are normalized relative to the signal with the highest magnitude |
| XXX_rx_phi.ascii                          | Azimuth angular information for rays detected at the receivers; the angles are normalized relative to the signal with the highest magnitude.                                |
| XXX_tx_theta.ascii                        | Elevation angular information for rays launched from the transmitter for all receiver positions within [-90, 90 deg], where 0 deg is the horizontal plane                   |
| XXX_rx_theta.ascii                        | Elevation angular information for rays detected at the receivers within [-90, 90 deg], where 0 deg is the horizontal plane  |
| XXX_activetxid.ascii                      | Active transmitter ID TxID (a zero based index which identifies the best serving cells) for each impulse response   |

In addition to the bandwidth and number of samples per impulse response settings, it can be chosen if the channel impulse responses shall be normalized with respect to its total power (squared sum of magnitudes) and / or to the signal with the shortest delay.

#### 6.11.5.2 Dynamic Channel Data Export into a MATLAB File

RPS can store the matrices into one MAT file using the function File -> Export -> Export Dynamic Data as Matrices readable with Matlab(TM). The MAT file contains the matrices and values as given in Table 6-17.

Table 6-17 Matrix names stored into the MAT file and their contents

| <i>Matrix Name</i>                          | <i>Contents</i>  |
|---|--|
| RPS_Magnitude(N, M)<br>N = rows<br>M = cols | Complex magnitudes of the channel impulse responses<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response<br>Max. Excess Delay [ns]: M/(RPS_Bandwidth [GHz])<br><br>Note: values are linear complex magnitudes ( $a_{dB} = 20\log(a)$ ) |
| RPS_NumResolvable Paths(N, 1)               | Number of resolvable paths (depends on the bandwidth of the radio channel or signal)<br>N: Total number of receiver positions  |
| RPS_Rx_Position(N, 3)                       | Coordinates [X Y Z] of each receiver position<br>N: Total number of positions  |
| RPS_Tx_Position(T, 3)                       | Coordinates [X Y Z] of each transmitter position<br>T: Total number of transmitters  |

| <i>Matrix Name</i>   | <i>Contents</i>  |
|----------------------|--|
| RPS_Rx_PHI(N, M)     | Azimuth angles of the paths in degree [0...360°] at the receivers<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response       |
| RPS_Tx_PHI(N, M)     | Azimuth angles of the paths in degree [0...360°] at the transmitters<br>N: Total number of positions<br>M: Number of samples for each impulse response             |
| RPS_Rx_THETA(N, M)   | Elevation angles of the paths in degree [-90...90°] at the receivers<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response    |
| RPS_Tx_THETA(N, M)   | Elevation angles of the paths in degree [-90...90°] at the transmitters<br>N: Total number of receiver positions<br>M: Number of samples for each impulse response |
| RPS_ActiveTxID(N, 1) | Active transmitter ID (TxID) as a zero-based index<br>N: Total number of receiver positions  |
| RPS_Tx_Names(T)      | The names of the transmitters as a cell array<br>T: Total number of available transmitters   |
| RPS_CenterFrequency  | Center frequency [GHz]   |
| RPS_Bandwidth        | Bandwidth [GHz]  |
| RPS_NoiseFloor       | Noise Floor [dB]   |
| RPS_Velocity         | Velocity [m/s]   |
| RPS_TimeInterval     | Time Interval between two positions [s]  |

In addition to the bandwidth and number of samples per impulse response settings, it can be chosen if the channel impulse responses shall be normalized with respect to its total power (squared sum of magnitudes) and / or to the signal with the shortest delay.

#### 6.11.5.3 Data Post-Processing in MATLAB

The distribution of RPS provides also a set of m-files that shall demonstrate the import and graphical representation of the dynamically created data in MATLAB. This shall help for getting started with RPS data post-processing in MATLAB. The scripts can be used for loading the data from both ASCII and \*.MAT data files. The m-files for the dynamic result analysis can be found in the `matlab_scripts` directory in the RPS installation path. Refer to section 6.11.4.3 for some comments on loading the data from ASCII files.

## 7 RPS COM Automation Interface

### 7.1 Capabilities using an Automation Interface

RPS Enterprise provides a COM interface that enables a convenient way to use RPS as a channel prediction engine within a simulation, for example of a complete radio network. Clients can control the complete RPS framework remotely. This means that the configuration, simulation, and result analysis can be done using COM. RPS acts as a COM server application that responds to requests from any client that supports COM. MATLAB, all MS Office applications, Visual Basic, Visual C++, or Corel are products that support COM. The principle of using RPS as a highly accurate channel prediction engine embedded into the radio network planning workflow is depicted in Fig. 7-1. In combination with the parallelization of computations in a LAN that is offered by RPS Enterprise, this system represents an extremely powerful RNP platform.

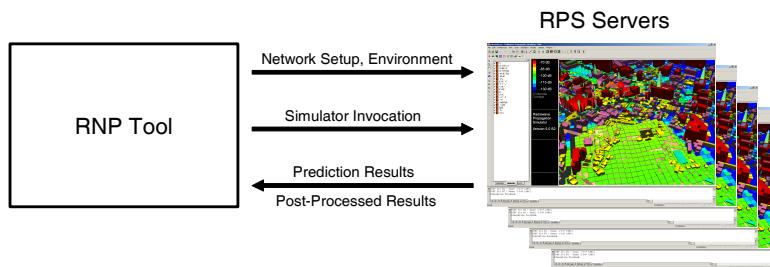


Fig. 7-1 Radio network planning with highly accurate channel prediction

In addition to the embedding of RPS into another simulator, the COM interface can be used to perform simulations with different parameter settings automatically. For instance, the optimization of a radio network is possible using some lines of Visual Basic code running on Excel.

The public methods of the underlying COM interface are declared by an IDL interface. This interface was designed to provide a minimal number of concise, yet powerful methods. The parameters of the type strings are case sensitive.

The next Section describes the methods of the interface and its parameters. Afterwards, an example demonstrates how simple it is to use RPS as a COM server.

## 7.2 COM Interface Specification

Table 7-1 Overview

|         |   |
|---------|---|
| boolean | <b>Open(BSTR str_FileName)</b>  |
| boolean | <b>Save(BSTR str_FileName)</b>  |
| boolean | <b>Quit()</b>   |
| boolean | <b>Run()</b>  |
| boolean | <b>Stop()</b>   |
| boolean | <b>IsRunning()</b>  |
| boolean | <b>RetrieveIntermediateResults()</b>  |
| VARIANT | <b>GetStatisticsParam(BSTR str_Param)</b>   |
| boolean | <b>SelectCustomPropagationModel(BSTR str_description)</b>   |
| boolean | <b>SetSimulationMode(BSTR str_SimulationMode)</b>   |
| boolean | <b>SetPreferenceParam(BSTR str_Param, VARIANT var_Value)</b>  |
| VARIANT | <b>GetPreferenceParam(BSTR str_Param)</b>   |
| long    | <b>GetBSCount()</b>   |
| boolean | <b>SetBSPParam(long long_ID, BSTR str_Param, VARIANT var_Value)</b>   |
| VARIANT | <b>GetBSPParam(long long_ID, BSTR str_Param)</b>  |
| long    | <b>GetBSIDFromIndex(long long_index)</b>  |
| long    | <b>GetBSIDFromName(BSTR str_Name)</b>   |
| long    | <b>GetMSCCount()</b>  |
| boolean | <b>SetMSPParam(long long_Index, BSTR str_Param, VARIANT var_Value)</b>  |
| VARIANT | <b>GetMSPParam(long long_Index, BSTR str_Param)</b>   |
| VARIANT | <b>GetMSResult(long long_Index, BSTR str_Param, long long_BSID)</b>   |
| long    | <b>AddMS(double double_Position_X, double double_Position_Y, double double_Position_Z, double double_Dimension_X, double double_Dimension_Y, long long_rxgroupid)</b> |
| boolean | <b>RemoveMS(long long_Index)</b>  |
| boolean | <b>RemoveMSAll()</b>  |
| boolean | <b>SetMSAntenna(long long_id)</b>   |
| long    | <b>GetMSGROUPCount()</b>  |
| long    | <b>AddMSGROUP(BSTR str_Name)</b>  |
| boolean | <b>RemoveMSGROUP(long long_ID)</b>  |
| long    | <b>GetMSGROUPIDFromIndex(long long_index)</b>   |
| long    | <b>GetMSGROUPIDFromName(BSTR str_Name)</b>  |
| boolean | <b>SetMSGROUPParam(long long_ID, BSTR str_Param, VARIANT var_Value)</b>   |

|         |  |
|---------|--|
| VARIANT | <b>GetMSGroupParam</b> (long long_ID, BSTR str_Param)                    |
| long    | <b>GetAntennaCount()</b>   |
| long    | <b>GetAntennalIDFromIndex</b> (long long_index)                          |
| long    | <b>GetAntennalIDFromName</b> (BSTR str_Name)                             |
| boolean | <b>SetAntennaParam</b> (long long_ID, BSTR str_Param, VARIANT var_Value) |
| VARIANT | <b>GetAntennaParam</b> (long long_ID, BSTR str_Param)                    |
| long    | <b>GetAttractionAreaCount()</b>  |
| VARIANT | <b>GetAttractionAreaParam</b> (long long_Index, BSTR str_Param)          |
| void    | <b>SetApplicationVisible</b> (boolean bool_state)                        |
| void    | <b>Refresh()</b>   |

Table 7-2 Method Details

|               |   |
|---------------|---|
| boolean       | <b>Open</b> (BSTR str_FileName)                           |
| Description:  | Opens an RPS project.                                     |
| Parameter:    | str_FileName: path and file name                          |
| Return value: | true: operation was successful<br>false: operation failed |

|               |   |
|---------------|---|
| boolean       | <b>Save</b> (BSTR str_FileName)                           |
| Description:  | Saves an RPS project.                                     |
| Parameter:    | str_FileName: path and file name                          |
| Return value: | true: operation was successful<br>false: operation failed |

|               |   |
|---------------|---|
| boolean       | <b>Quit</b> ()                                |
| Description:  | Quits the RPS application and the connection. |
| Parameter:    | n/a   |
| Return value: | always true                                   |

|               |   |
|---------------|---|
| boolean       | <b>Run</b> ()   |
| Description:  | Starts the simulation with the current settings.          |
| Parameter:    | n/a   |
| Return value: | true: operation was successful<br>false: operation failed |

|                |   |       |                          |        |                  |
|----------------|---|-------|--------------------------|--------|------------------|
| <b>boolean</b> | <b>Stop()</b>   |       |                          |        |                  |
| Description:   | Stops the running simulation.   |       |                          |        |                  |
| Parameter:     | n/a   |       |                          |        |                  |
| Return value:  | <table> <tr> <td>true:</td><td>operation was successful</td></tr> <tr> <td>false:</td><td>operation failed</td></tr> </table> | true: | operation was successful | false: | operation failed |
| true:          | operation was successful  |       |                          |        |                  |
| false:         | operation failed  |       |                          |        |                  |

|                |   |       |                             |        |                           |
|----------------|---|-------|-----------------------------|--------|---------------------------|
| <b>boolean</b> | <b>IsRunning()</b>  |       |                             |        |                           |
| Description:   | Checks the status of a running simulation.  |       |                             |        |                           |
| Parameter:     | n/a   |       |                             |        |                           |
| Return value:  | <table> <tr> <td>true:</td><td>simulation is still running</td></tr> <tr> <td>false:</td><td>simulation is not running</td></tr> </table> | true: | simulation is still running | false: | simulation is not running |
| true:          | simulation is still running   |       |                             |        |                           |
| false:         | simulation is not running   |       |                             |        |                           |

|                |  |       |                          |        |                  |
|----------------|--|-------|--------------------------|--------|------------------|
| <b>boolean</b> | <b>RetrieveIntermediateResults()</b>   |       |                          |        |                  |
| Description:   | Queries the results from the Raytracing engine during a running simulation. Thus all results obtained up to that moment can be analyzed. |       |                          |        |                  |
| Parameter:     | n/a  |       |                          |        |                  |
| Return value:  | <table> <tr> <td>true:</td><td>operation was successful</td></tr> <tr> <td>false:</td><td>operation failed</td></tr> </table>            | true: | operation was successful | false: | operation failed |
| true:          | operation was successful   |       |                          |        |                  |
| false:         | operation failed   |       |                          |        |                  |

|                                   |   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
|-----------------------------------|---|-----------------------------------|--------------------------|--------------------------|---------------------|-------------------------|-------------------------|----------------------|----------------------|---------------------|-----------------------|---------------------|----------------------------|---------------------|------------------------|---------------------|-----------------------------|----------------------|------------------------|----------------------|-----------------------------|----------------------|-----------------------------|
| <b>VARIANT</b>                    | <b>GetStatisticsParam(BSTR str_Param)</b>   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| Description:                      | Returns properties of the statistics class.   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| Parameter:                        | str_Param: parameter (string)   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| Return value:                     | <table> <tr> <td>parameter value:</td><td>operation was successful</td></tr> <tr> <td>null:</td><td>operation failed</td></tr> </table>   | parameter value:                  | operation was successful | null:                    | operation failed    |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| parameter value:                  | operation was successful  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| null:                             | operation failed  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| Parameter,<br>Return value        | <table> <tr> <td>SimulationStatusPercent (double):</td><td>completed fraction [%]</td></tr> <tr> <td>SimulationTime (double):</td><td>simulation time [s]</td></tr> <tr> <td>NumLaunchedRays (long):</td><td>number of launched rays</td></tr> <tr> <td>NumSplitRays (long):</td><td>number of split rays</td></tr> <tr> <td>RefCountSum (long):</td><td>number of reflections</td></tr> <tr> <td>RefCountMax (long):</td><td>max. number of reflections</td></tr> <tr> <td>PenCountSum (long):</td><td>number of penetrations</td></tr> <tr> <td>PenCountMax (long):</td><td>max. number of penetrations</td></tr> <tr> <td>DiffCountSum (long):</td><td>number of diffractions</td></tr> <tr> <td>DiffCountMax (long):</td><td>max. number of diffractions</td></tr> <tr> <td>IRCountTotal (long):</td><td>number of impulse responses</td></tr> </table> | SimulationStatusPercent (double): | completed fraction [%]   | SimulationTime (double): | simulation time [s] | NumLaunchedRays (long): | number of launched rays | NumSplitRays (long): | number of split rays | RefCountSum (long): | number of reflections | RefCountMax (long): | max. number of reflections | PenCountSum (long): | number of penetrations | PenCountMax (long): | max. number of penetrations | DiffCountSum (long): | number of diffractions | DiffCountMax (long): | max. number of diffractions | IRCountTotal (long): | number of impulse responses |
| SimulationStatusPercent (double): | completed fraction [%]  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| SimulationTime (double):          | simulation time [s]   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| NumLaunchedRays (long):           | number of launched rays   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| NumSplitRays (long):              | number of split rays  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| RefCountSum (long):               | number of reflections   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| RefCountMax (long):               | max. number of reflections  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| PenCountSum (long):               | number of penetrations  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| PenCountMax (long):               | max. number of penetrations   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| DiffCountSum (long):              | number of diffractions  |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| DiffCountMax (long):              | max. number of diffractions   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |
| IRCountTotal (long):              | number of impulse responses   |                                   |                          |                          |                     |                         |                         |                      |                      |                     |                       |                     |                            |                     |                        |                     |                             |                      |                        |                      |                             |                      |                             |

|                |   |
|----------------|---|
| <b>boolean</b> | <b>SelectCustomPropagationModel(BSTR str_description)</b> |
| Description:   | Selects a custom propagation model according to its name. |
| Parameter:     | str_description: plug-in name (string)                    |

|               |   |                          |
|---------------|---|--------------------------|
| Return value: | true:   | operation was successful |
|               | false:  | operation failed         |
| Comments:     | if more than one plug-in with equal descriptions are available, the first of them is selected |                          |

|                   |  |  |
|-------------------|--|--|
| boolean           | <b>SetSimulationMode(BSTR str_SimulationMode)</b>  |  |
| Description:      | Sets the simulation mode to be used when running a simulation.   |  |
| Parameter:        | str_SimulationMode: simulation mode specifier (string)   |  |
| Return value:     | true: operation was successful<br>false: operation failed  |  |
| Parameter values: | RT3D: raytracing 3D<br>RT25D: raytracing 2.5D<br>RT3DCS: raytracing 3D client server<br>RT25DCS: raytracing 2.5D client server<br>COST231WI: COST 231 Walfisch-Ikegami model (outdoor)<br>COST231MWM: COST 231 Multi Wall Model (indoor)<br>CUSTOMINDOOR: custom plug-in indoor propagation model<br>CUSTOMOUTDOOR: custom plug-in outdoor propagation model |  |

|                                 |  |  |
|---------------------------------|--|--|
| boolean                         | <b>SetPreferenceParam(BSTR str_Param, VARIANT var_Value)</b>   |  |
| Description:                    | Changes settings for the simulation parameters.  |  |
| Parameter:                      | str_Param: parameter (string)<br>var_Value: parameter value (variant)  |  |
| Return value:                   | true: operation was successful<br>false: operation failed  |  |
| Parameter,<br>Variant data type | Bandwidth: double [GHz]<br>Frequency: double [GHz]<br>NoiseFloor: double [dB/dBm]<br>Penetration: boolean<br>RestrictPenetration: boolean<br>PenetrationCount: long<br>Reflection: boolean<br>RestrictReflection: boolean<br>ReflectionCount: long<br>Diffraction: boolean<br>RestrictDiffraction: boolean<br>DiffractionCount: long<br>DiffractionOnlyInShadowRegions: boolean<br>DiffractionReduceResolution: boolean<br>25DOverRooftopPropagation: boolean<br>25DSingleReflectionOnHorizontalPlanes: boolean<br>3DOverRooftopPropagation: boolean<br>RaySplitting: boolean<br>RaySplittingMaxTubeSize: double [m] |  |

|                                  |              |
|----------------------------------|--------------|
| RestrictDelay:                   | boolean      |
| DelayMax:                        | double [ns]  |
| ThreadCount:                     | long         |
| AngularRestrictionPHIBegin:      | double [deg] |
| AngularRestrictionPHIEnd:        | double [deg] |
| AngularRestrictionPHIStepSize:   | double [deg] |
| AngularRestrictionTHETABegin:    | double [deg] |
| AngularRestrictionTHETAEnd:      | double [deg] |
| AngularRestrictionTHETAStepSize: | double [deg] |
| SaveRays:                        | boolean      |
| EnableIncrementalPredictions:    | boolean      |

|               |   |
|---------------|---|
| VARIANT       | <b>GetPreferenceParam(BSTR str_Param)</b>                           |
| Description:  | Returns the value for a simulation parameter.                       |
| Parameter:    | str_Param: parameter (string), see SetPreferenceParam               |
| Return value: | parameter value: operation was successful<br>null: operation failed |

|               |                                      |
|---------------|--------------------------------------|
| long          | <b>GetBSCount()</b>                  |
| Description:  | Returns the number of base stations. |
| Parameter:    | n/a                                  |
| Return value: | number of base stations              |

|                                 |  |
|---------------------------------|--|
| boolean                         | <b>SetBSPParam(long long_ID, BSTR str_Param, VARIANT var_Value)</b>  |
| Description:                    | Changes settings for the BS parameters.  |
| Parameter:                      | long_Index: GUID of the BS (long)<br>str_Param: parameter (string)<br>var_Value: parameter value (variant)   |
| Return value:                   | true: operation was successful<br>false: operation failed  |
| Parameter,<br>Variant data type | TransmitPowerDB: double [dB]<br>ActiveFlag: boolean<br>ObsoleteFlag: boolean<br>Position.X: double [m]<br>Position.Y: double [m]<br>Position.Z: double [m]<br>Altitude: double [m]<br>Height: double [m]<br>Name: string<br>AntennaID: long: the antenna GUID<br>ReferenceAnglePHI: double [deg]<br>ReferenceAngleTHETA: double [deg]<br>Description: string |

|               |  |
|---------------|--|
| CableLoss_DL: | double [dB]  |
| CableLoss_UL: | double [dB]  |
| NoiseFigure:  | double [dB]  |
| Comments:     | ObsoleteFlag is changed to true, if a new antenna is applied to the BS, or the position is set |

|               |   |
|---------------|---|
| VARIANT       | <b>GetBSPParam</b> (long long_ID, BSTR str_Param)                                   |
| Description:  | Returns properties of a BS.   |
| Parameter:    | long_Index: GUID of the BS (long)<br>str_Param: parameter (string), see SetBSPParam |
| Return value: | parameter value: operation was successful<br>null: operation failed                 |

|               |   |
|---------------|---|
| long          | <b>GetBSIDFromIndex</b> (long long_index)                     |
| Description:  | Returns the GUID of the base station by the index.            |
| Parameter:    | long_index: index (zero based) of the BS (long)               |
| Return value: | parameter value: operation successful<br>-1: operation failed |

|               |   |
|---------------|---|
| long          | <b>GetBSIDFromName</b> (BSTR str_Name)                        |
| Description:  | Returns the GUID of the base station by name.                 |
| Parameter:    | str_Name: name of the BS (string)                             |
| Return value: | parameter value: operation successful<br>-1: operation failed |

|               |                                     |
|---------------|-------------------------------------|
| long          | <b>GetMSCount</b> ()                |
| Description:  | Returns the number of available MS. |
| Parameter:    | n/a                                 |
| Return value: | number of available MS              |

|               |  |
|---------------|--|
| boolean       | <b>SetMSParam</b> (long long_Index, BSTR str_Param, VARIANT var_Value)   |
| Description:  | Set properties of a MS.  |
| Parameter:    | long_Index: index (zero based) of the MS (long)<br>str_Param: parameter (string)<br>var_Value: parameter value (variant) |
| Return value: | true: operation was successful<br>false: operation failed  |
| Parameter,    | Position.X: double [m]   |

|                   |                      |                              |
|-------------------|----------------------|------------------------------|
| Variant data type | Position.Y:          | double [m]                   |
|                   | Position.Z:          | double [m] (absolute height) |
|                   | HeightOverGround :   | double [m] (relative to DEM) |
|                   | Dimension.X:         | double [m]                   |
|                   | Dimension.Y:         | double [m]                   |
|                   | ReferenceAnglePHI:   | double [deg]                 |
|                   | ReferenceAngleTHETA: | double [deg]                 |

|               |  |
|---------------|--|
| VARIANT       | <b>GetMSParam</b> (long long_Index, BSTR str_Param)  |
| Description:  | Returns the value for a MS parameter.  |
| Parameter:    | long_Index: index (zero based) of the MS (long)<br>str_Param: parameter (string), see SetMSParam<br>plus:<br>AntennaID: long: the antenna GUID |
| Return value: | parameter value: operation was successful<br>null: operation failed  |

|                                   |  |
|-----------------------------------|--|
| VARIANT                           | <b>GetMSResult</b> (long long_Index, BSTR str_Param, long long_BSID)   |
| Description:                      | Returns results for a MS.  |
| Parameter:                        | long_Index: index (zero based) of the MS (long)<br>str_Param: result parameter (string)<br>long_BSID: GUID of the base station (long), not used for all parameters   |
| Return value:                     | return value or matrix of values: operation was successful<br>null: operation failed   |
| Result parameter,<br>Return value | <i>MSPowerDB.BestServingCell</i> : double [dB]<br>sum of power values of signals related to the best serving cell;<br>null, if no signal was received;<br><br>BSID parameter is ignored<br><br><i>MSPowerDB</i> : double [dB]<br>sum of power values related to the BS with index BSID;<br>null, if no signal was received<br><br><i>PathLossDB.BestServingCell</i> : double [dB]<br>pathloss attenuation between MS and the best serving cell:<br>pathloss = BSPower – MSPower;<br>null, if no signal was received;<br><br>BSID is ignored<br><br><i>PathLossDB</i> : double [dB]<br>pathloss attenuation between MS and the BS with BSID:<br>pathloss = BSPower – MSPower;<br>null, if no signal was received<br><br><i>DelaySpread.BestServingCell</i> : double [ns]<br>delay spread of the signals related to the best serving cell; |

0, if no signal was received

BSID is ignored

*DelaySpread: double [ns]*

delay spread of the signals related to the best serving cell;  
null, if no signal was received

*SignalToInterferenceRatio: double [dB]*

Signal to interference ratio between the best serving cell and all  
interference signals  
null, if less than two signals are available

BSID is ignored

*ChannelImpulseResponseArray.AllEntries: variant (double)*  
matrix with the following content:

column(0) Magnitude.RealPart (P = 20log(Mag)!)  
column(1) Magnitude.ImaginaryPart  
column(2) Tau [ns]  
column(3) BSID  
column(4) MSPHI  
column(5) MSTHETA  
column(6) BSPHI  
column(7) BSTHETA

row(0..N-1) for N entries

null, if no signal was received:

BSID is ignored

*ChannelImpulseResponseArray.BestServingCell:*

see ChannelImpulseResponseArray.AllEntries, but only the signals  
related to the best serving cell are returned

null, if no signal was received

BSID is ignored

*ChannelImpulseResponseArray.*

see ChannelImpulseResponseArray.AllEntries, but only values  
related to the BSID are returned

null, if no signal was received

|                     |   |                    |                           |                    |                           |                    |                           |                     |                             |
|---------------------|---|--------------------|---------------------------|--------------------|---------------------------|--------------------|---------------------------|---------------------|-----------------------------|
| long                | <b>AddMS</b> (double double_Position_X, double double_Position_Y, double double_Position_Z, double double_Dimension_X, double double_Dimension_Y, long long_rxgroupid)  |                    |                           |                    |                           |                    |                           |                     |                             |
| Description:        | Adds a new MS.  |                    |                           |                    |                           |                    |                           |                     |                             |
| Parameter:          | <table> <tr> <td>double_Position_X:</td><td>x coordinate (double) [m]</td></tr> <tr> <td>double_Position_Y:</td><td>y coordinate (double) [m]</td></tr> <tr> <td>double_Position_Z:</td><td>z coordinate (double) [m]</td></tr> <tr> <td>double_Dimension_X:</td><td>Dimension in x (double) [m]</td></tr> </table> | double_Position_X: | x coordinate (double) [m] | double_Position_Y: | y coordinate (double) [m] | double_Position_Z: | z coordinate (double) [m] | double_Dimension_X: | Dimension in x (double) [m] |
| double_Position_X:  | x coordinate (double) [m]   |                    |                           |                    |                           |                    |                           |                     |                             |
| double_Position_Y:  | y coordinate (double) [m]   |                    |                           |                    |                           |                    |                           |                     |                             |
| double_Position_Z:  | z coordinate (double) [m]   |                    |                           |                    |                           |                    |                           |                     |                             |
| double_Dimension_X: | Dimension in x (double) [m]   |                    |                           |                    |                           |                    |                           |                     |                             |

|               |   |
|---------------|---|
|               | double_Dimension_Y: Dimension in y (double) [m] |
|               | long_rxgroupid: GUID of the MS Group (long)     |
| Return value: | index (zero based) for the created MS           |

|               |   |
|---------------|---|
| boolean       | <b>RemoveMS</b> (long long_Index)                         |
| Description:  | Removes a MS.   |
| Parameter:    | long_Index: Index (zero based) of the MS (long)           |
| Return value: | true: operation was successful<br>false: operation failed |

|               |   |
|---------------|---|
| boolean       | <b>RemoveMSAll</b> ()   |
| Description:  | Removes all MS's.   |
| Parameter:    | n/a   |
| Return value: | true: operation was successful<br>false: operation failed, for instance no MS available |

|               |  |
|---------------|--|
| boolean       | <b>SetMSAntenna</b> (long long_id)   |
| Description:  | Sets the antenna to use with all MS.   |
| Parameter:    | long_id: GUID of the antenna (long)  |
| Return value: | true: operation was successful<br>false: operation failed, for instance wrong GUID |

|               |  |
|---------------|--|
| long          | <b>GetMSGROUPCount</b> ()                  |
| Description:  | Returns the number of available MS groups. |
| Parameter:    | n/a  |
| Return value: | number of MS groups                        |

|               |   |
|---------------|---|
| long          | <b>AddMSGROUP</b> (BSTR str_Name)           |
| Description:  | Adds a new MS group.                        |
| Parameter:    | str_Name: name of the new MS group (string) |
| Return value: | GUID of the added MS group (long)           |

|               |                                      |
|---------------|--------------------------------------|
| boolean       | <b>RemoveMSGROUP</b> (long long_ID)  |
| Description:  | Removes an MS Group.                 |
| Parameter:    | long_ID: GUID of the MS group (long) |
| Return value: | true: operation was successful       |

|  |        |                  |
|--|--------|------------------|
|  | false: | operation failed |
|--|--------|------------------|

|               |  |  |
|---------------|--|--|
| long          | <b>GetMSGroupIDFromIndex(long long_index)</b>  |  |
| Description:  | Returns the GUID of the MS group by the index. |  |
| Parameter:    | long_index:                                    | index (zero based) of the MS group (long)    |
| Return value: | parameter value:                               | operation successful<br>-1: operation failed |

|               |  |  |
|---------------|--|--|
| long          | <b>GetMSGroupIDFromName(BSTR str_Name)</b> |  |
| Description:  | Returns the GUID of the MS group by name.  |  |
| Parameter:    | str_Name:                                  | name of the MS group (string)                |
| Return value: | parameter value:                           | operation successful<br>-1: operation failed |

|                                 |   |                             |
|---------------------------------|---|-----------------------------|
| boolean                         | <b>SetMSGroupParam(long long_ID, BSTR str_Param, VARIANT var_Value)</b> |                             |
| Description:                    | Changes settings for the MS group parameters.                           |                             |
| Parameter:                      | long_ID:  | GUID of the MS group (long) |
|                                 | str_Param:  | parameter (string)          |
|                                 | var_Value:  | parameter value (variant)   |
| Return value:                   | true:   | operation was successful    |
|                                 | false:  | operation failed            |
| Parameter,<br>Variant data type | Name:   | (string)                    |
|                                 | ActiveFlag:   | (boolean)                   |

|               |  |   |
|---------------|--|---|
| VARIANT       | <b>GetMSGroupParam(long long_ID, BSTR str_Param)</b> |   |
| Description:  | Returns the value for a MS group parameter.          |   |
| Parameter:    | long_ID:   | GUID of the MS group (long)             |
|               | str_Param:   | parameter (string), see SetMSGroupParam |
| Return value: | parameter value:                                     | operation was successful                |
|               | null:  | operation failed                        |

|               |   |  |
|---------------|---|--|
| long          | <b>GetAntennaCount()</b>                  |  |
| Description:  | Returns the number of available antennas. |  |
| Parameter:    | n/a                                       |  |
| Return value: | number of antennas                        |  |

|      |   |  |
|------|---|--|
| long | <b>GetAntennaIDFromIndex(long long_index)</b> |  |
|      | parameter value:                              | operation successful<br>-1: operation failed |

|               |   |
|---------------|---|
| Description:  | Returns the GUID of the antenna by the index.                 |
| Parameter:    | long_index: index (zero based) of the antenna (long)          |
| Return value: | parameter value: operation successful<br>-1: operation failed |

|               |   |
|---------------|---|
| long          | <b>GetAntennalIDFromName(BSTR str_Name)</b>                   |
| Description:  | Returns the GUID of the antenna by name.                      |
| Parameter:    | str_Name: name of the antenna (string)                        |
| Return value: | parameter value: operation successful<br>-1: operation failed |

|                                 |  |
|---------------------------------|--|
| boolean                         | <b>SetAntennaParam(long long_ID, BSTR str_Param, VARIANT var_Value)</b>                                      |
| Description:                    | Changes settings for the antenna parameters.   |
| Parameter:                      | long_ID: GUID of the antenna (long)<br>str_Param: parameter (string)<br>var_Value: parameter value (variant) |
| Return value:                   | true: operation was successful<br>false: operation failed  |
| Parameter,<br>Variant data type | Name: (string)<br>Gain: (double)<br>Polarization: (string)<br>values: VP, HP, LHCP, RHCP                     |

|               |   |
|---------------|---|
| VARIANT       | <b>GetAntennaParam(long long_ID, BSTR str_Param)</b>                                      |
| Description:  | Returns the value for an antenna parameter.   |
| Parameter:    | long_ID: GUID of the antenna (long)<br>str_Param: parameter (string), see SetAntennaParam |
| Return value: | parameter value: operation was successful<br>null: operation failed                       |

|               |   |
|---------------|---|
| long          | <b>GetAttractionAreaCount()</b>                   |
| Description:  | Returns the number of available attraction areas. |
| Parameter:    | n/a   |
| Return value: | number of the attraction areas                    |

|              |  |
|--------------|--|
| VARIANT      | <b>GetAttractionAreaParam(long long_Index, BSTR str_Param)</b> |
| Description: | return properties for attraction areas                         |
| Parameter:   | long_Index: index (zero based) of the attraction area (long)   |

|               |                  |  |
|---------------|------------------|--|
|               | str_Param:       | parameter (string)                                     |
| Return value: | parameter value: | operation was successful                               |
|               | null:            | operation failed, i.e. no transitions are available    |
| Parameter,    | Position.X:      | (double) [m]   |
| Return value  | Position.Y:      | (double) [m]   |
|               | Position.Z:      | (double) [m]   |
|               | Radius:          | (double) [m]   |
|               | Transitions:     | vector (long) with indices of related attraction areas |

|               |  |
|---------------|--|
| void          | <b>SetApplicationVisible(boolean bool_state)</b>                 |
| Description:  | set the RPS application either visible or invisible              |
| Parameter:    | bool_state:  |
|               | true: application is visible                                     |
|               | false: application is hidden (acts as service in the background) |
| Return value: | n/a  |

|               |   |
|---------------|---|
| void          | <b>Refresh()</b>  |
| Description:  | Refreshes the RPS window to reflect all recent changes through the COM interface. |
| Parameter:    | n/a   |
| Return value: | always true   |

### 7.3 Automation Example

An example is provided that demonstrates the usage of the COM automation interface. A macro written in Visual Basic for Applications (VBA) using Microsoft Excel is used to perform some simulations using all available built-in propagation models sequentially. Key indicators such as the simulation time or the number of detected rays are presented then in an Excel sheet.

Furthermore, this example can be used to perform simulations with different parameters and to compare them directly. The included file can be found here:

- subdirectory: %ApplicationPath%\examples\comexample
- filename: comexample1.xls

The sheet should look similar to the one shown in Fig. 7-2.

| COM Example                                  |  |       |           |            |                       |                        |
|--|--|-------|-----------|------------|-----------------------|------------------------|
| RPS File:                                    | c:\rps\Examples\comexample\comexample1.rps |       |           |            |                       |                        |
| Store Results in Table (1 or 2):             | 1  |       |           |            |                       |                        |
| <i>Press this Button to run the Macro...</i> |  |       |           |            |                       |                        |
| Summary 1                                    | RT2D                                       | RT3D  | COST231WI | COST231MWM | COST 231 WI<br>Plugin | COST 231 MWM<br>Plugin |
| BSCount                                      | 2  | 2     | 2         | 2          | 2                     | 2                      |
| MSCount                                      | 72   | 72    | 72        | 72         | 72                    | 72                     |
| RayCount                                     | 432  | 20889 | 72        | 72         | 0                     | 0                      |
| SimulationTime                               | 11   | 39    | 0         | 0          | 0                     | 0                      |
| Summary 2                                    | RT2D                                       | RT3D  | COST231WI | COST231MWM | COST 231 WI<br>Plugin | COST 231 MWM<br>Plugin |
| BSCount                                      | 2  | 2     | 2         | 2          | 2                     | 2                      |
| MSCount                                      | 72   | 72    | 72        | 72         | 72                    | 72                     |
| RayCount                                     | 333  | 337   | 72        | 38         | 72                    | 38                     |
| SimulationTime                               | 22   | 52    | 1         | 0          | 0                     | 0                      |

Fig. 7-2 COM example summary table

To use this example, the user has to enter a valid RPS file (e.g. comexample1.rps) and the corresponding path (for this example the path of the \comexample directory) into the upper part of the Excel sheet. The provided sample RPS file contains a simple scenario with one active transmitter and a row of receivers. Additionally, it can be specified in which table the simulation results will be inserted (1 or 2). You should choose 1 first.

The button Start Analysis runs the macro. This macro contains two important procedures:

- DoAnalysis() is called, when the button Start Analysis is clicked.
- AddMeasure(...) is called from the DoAnalysis() procedure for each simulation run.

DoAnalysis() establishes a connection to RPS first. This is done in CheckRPSObject() using the following snippet of code:

```
Dim obj As Object
Set obj = CreateObject("Rps.Document")
```

If the RPS.TLB file (located in the RPS application directory) is added as reference to the VBA project, the syntax of the RPS COM interface is available at design time. The TLB file contains the complete methods and syntax definitions of the RPS COM interface. To take advantage of the built-in VBA syntax highlighting, replace the above code line with the following one:

```
Dim obj As Rps.Document
```

RPS can be run in visible or hidden mode. The behavior can be changed using this method:

```
obj.SetApplicationVisible(True/False)
```

Running the simulation in visible mode gives the user the chance to cancel a simulation at any time.

After the connection is established, a file must be loaded into RPS. In the example this is done by the following call:

```
obj.Open(Range("B2").Value)
```

The path and filename are picked from the Excel sheet.

Next, the AddMeasure(...) procedure is called for each cycle in a series of simulation runs. Within this procedure all BS are declared as obsolete, to enforce a new simulation:

```
obj.SetBSPParam(BSID, "ObsoleteFlag", True)
```

Please note that the BS is addressed by its GUID (Global Unique ID), which is queried using

```
BSID = obj.GetBSIDFromIndex(i)
```

first. The simulation with a certain propagation model is performed, and some results are transferred into the Excel sheet. For further information, refer to the comments in the VBA code.

Now you could make a modification in the RPS file (e.g. turning an antenna by a certain angle, or changing the transmit power, etc.) to see the differences between two simulation runs. Open the Excel sheet again and select the results table 2 instead of 1. Restart the simulation by pressing the button. The results are automatically transferred into the sheet, and the differences are shown on the bottom.

As another example, a short code sample is provided that demonstrates the handling of variant arrays, for instance if a channel impulse response array is returned by the COM interface. The following lines of code retrieve the channel impulse response data for each available MS. Secondly, the delay  $\tau$  is transferred into an Excel sheet for each impulse response entry. Note that the variant arrays returned by RPS are zero-based arrays:

```
For j = 0 To obj.GetMSCount - 1
    v = obj.GetMSResult(j, "ChannelImpulseResponseArray.AllEntries", 0)
    Cells(j + 1, 1).Value = "MS " & j
    Cells(1, 2).Activate
    If IsArray(v) Then
        For i = 0 To UBound(v, 1)
            Cells(j + 1, i + 1).Select
            ActiveCell.FormulaR1C1 = v(i, 2)
        Next i
    End If
Next j
```



## 8 Examples

The distribution of RPS provides some examples and antenna diagrams. Additionally, the files of the Getting Started Tutorial are available as well.

### 8.1 Antenna Diagrams

With the RPS installation some calculated as well as practically measured radiation patterns of antennas are included in the subdirectory

`%ApplicationPath%\library\antennas`. They embrace standard antennas like dipole, lens, patch or horn antennas. For each antenna there are three files, comprising the configuration data in a `*.cfg` file, and the horizontal and the vertical radiation patterns in `*.dat` files.

### 8.2 Simple Indoor Environment

- subdirectory: `%ApplicationPath%\examples\SimpleIndoorEnvironment`
- filename: `SimpleIndoorEnvironment.rps`

This example shall help to explain the basic functions and modules of the ray tracing algorithms. The environment is comparable with one floor in an office building. A transmitter is placed within a room and the assigned antenna diagram is comparable with a vertical polarized patch antenna. One receiver is placed within the environment. If the detected rays are saved (see General Preferences dialog), the behavior of the different ray tracing algorithms and empirical models can be investigated in detail.

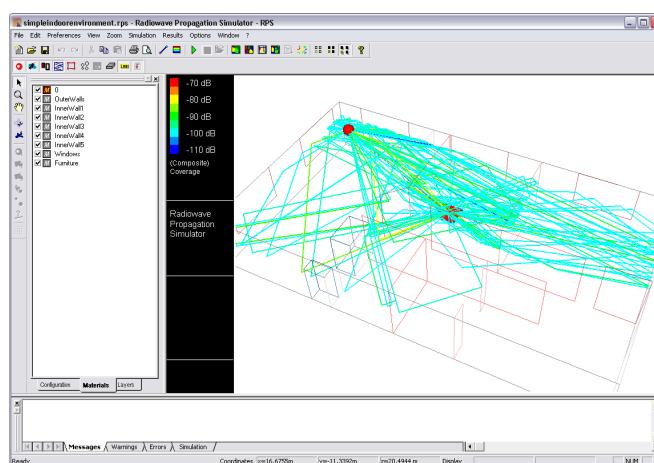


Fig. 8-1 Example Simple Indoor Environment (3D view)

This example is a good starting point to add new transmitters or to change the transmitter position. Furthermore, it is possible to add new receivers to calculate the radio channel along a straight line or within a sub area of the environment.

### 8.3 Pico Cellular Indoor Environment

- subdirectory: %ApplicationPath%\examples\PicoIndoorEnvironment
- filename: PicoIndoorEnvironment.rps

The second indoor example demonstrates a radio network planning example for an indoor wireless LAN network operating in the mm-wave range at 60 GHz. These type of networks will have very small cells. Since the attenuation of electromagnetic waves in walls is very high at these frequency ranges, each room must be covered by its own base station.

This example shall be used to get familiar with the selection and arrangement of suitable radiation patterns, matched to the geometrical environment.

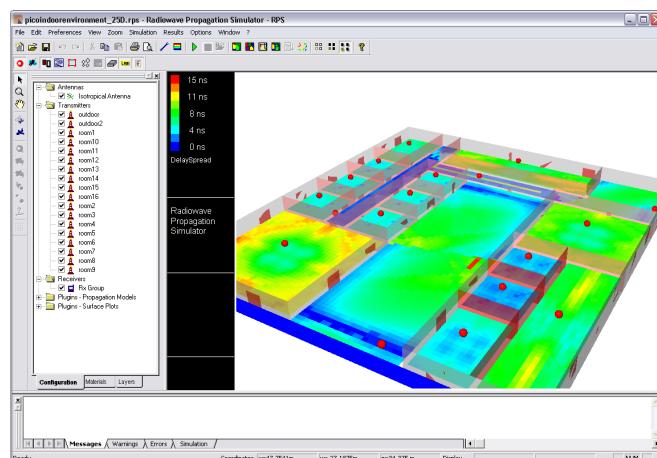


Fig. 8-2 Example pico cellular indoor environment

Furthermore, the following tasks shall help the user to learn more about the built-in functions of RPS:

- deactivate some transmitters and observe the coverage and SIR changes – no recalculation of the radio channel is required
- change the transmit power of some transmitters – the resulting coverage, delay spread, SIR is shown immediately
- change the receivers' polarization state to linear horizontal – only depolarized components are shown in coverage plots
- rotate the antennas of transmitters located in the middle hallway – the new prediction results are shown, no recalculation of the radio channel is required

### 8.4 Urban Outdoor Environment

- subdirectory: %ApplicationPath%\examples\UrbanOutdoorEnvironment
- filename: UrbanOutdoorEnvironment.rps

This example imitates an urban area. The environment comprises three transmitters. The radio channel is predicted in a rectangular area, where the distance between two adjacent receivers is 10 meters. The carrier frequency is 1GHz in this example.

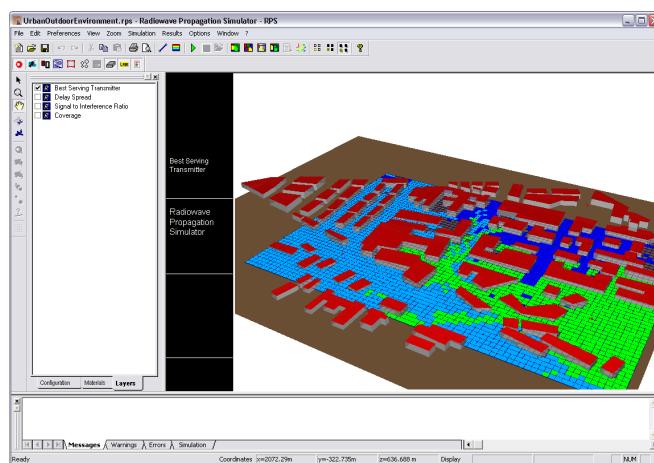


Fig. 8-3 Example Urban Outdoor Area

This example is dedicated to investigate the performance and the behavior of the ray tracing algorithms for different transmitter elevations. If the transmitters' height is lower or in order of the roofs of the buildings, the 2.5D ray tracing algorithms work very efficient. If the height of the transmitters is much higher than the mean building height, the 3D ray tracing algorithms shall be preferred. To minimize the required compute time utilizing the 3D algorithms, the  $\vartheta$ -values of the outgoing angles shall be set from  $0^\circ$  to  $90^\circ$ .

## 8.5 Dresden City Center Environment

- subdirectory: %ApplicationPath%\examples\Berlin
- filename: Dresden\_DEM.rps

This example demonstrates a complete and realistic scenario of the city center of Dresden. In addition to the examples depicted above, DEM surface height information is available. DEM information is necessary in scenarios, where the ground level cannot be modeled as a flat surface. In those environments, both transmitters and receivers must be placed in relation to the local height.

In this environment, 21 transmitters as site candidates were placed. All these transmitters have a realistic height over ground and a subset of them could provide satisfactory coverage for 2.5G or 3G cellular networks. Finally, a rectangular grid of receivers with a grid size of 25 meters was added relatively to the DEM surface level height.

The sample channel data given here was computed with the COST 231 WI model using a center frequency of 1 GHz.

The geographical and Cartesian coordinates are given as georeferenced coordinates. For an exact representation of the longitude/latitude coordinates the user should select "DHDN Germany 4" for the projection coordinate system and "DHDN" for the geographical coordinate system from the [Projection and Coordinate Settings Dialog](#), respectively.

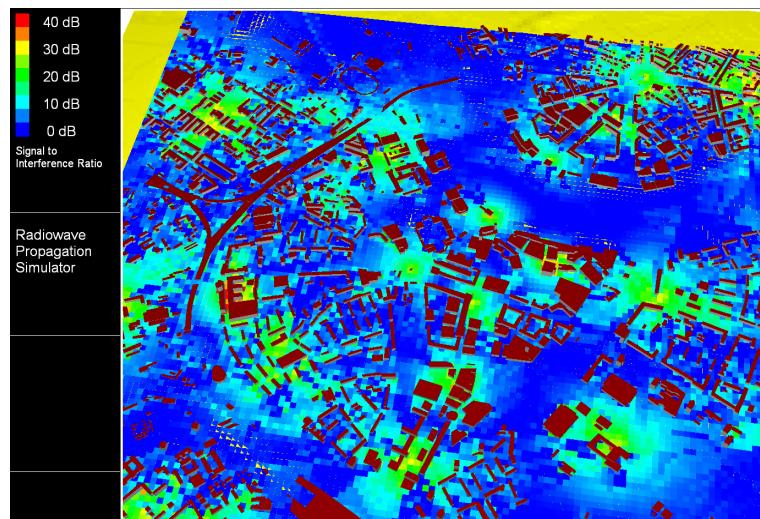


Fig. 8-4 Example Urban Outdoor Area

## 8.6 Suburban Outdoor Environment

- subdirectory: %ApplicationPath%\examples\SuburbanOutdoorEnvironment
- filename: SuburbanOutdoorEnvironment.rps

This example represents a suburban district. The region is characterized by small buildings. A transmitter is placed in the environment. The antenna diagram is similar to an ideal spherical half plane with a gain of 0 dB. The carrier frequency is 5.2 GHz. This scenario can be used to experience with a private hiperLAN network in such an outdoor scenario. The receivers are placed along a measurement route.

## 8.7 Dynamic Outdoor Example

- subdirectory: %ApplicationPath%\examples\DynamicOutdoorExample
- filename: DynamicOutdoorExample.rps

This example allows a good starting point for working with dynamic simulations. The scenario was chosen to be the urban example environment. Nine transmitters were placed into this environment, and moving corridors were defined along streets. To enable the dynamic simulations with pre-calculated radio channel data, receivers were placed along the moving corridors and the field strength is predicted for these positions.



Fig. 8-5 Example dynamic outdoor environment

### 8.8 Dynamic Indoor Example

- subdirectory: %ApplicationPath%\examples\DynamicIndoorExample
- filename: DynamicIndoorExample.rps

This example demonstrates, how an extensive definition of attraction areas and transition probabilities can be done. The environment is similar to the pico cellular indoor example. Over the entire area, a grid of receivers is defined. The relative distance of two adjacent receivers is 0.25 meters.

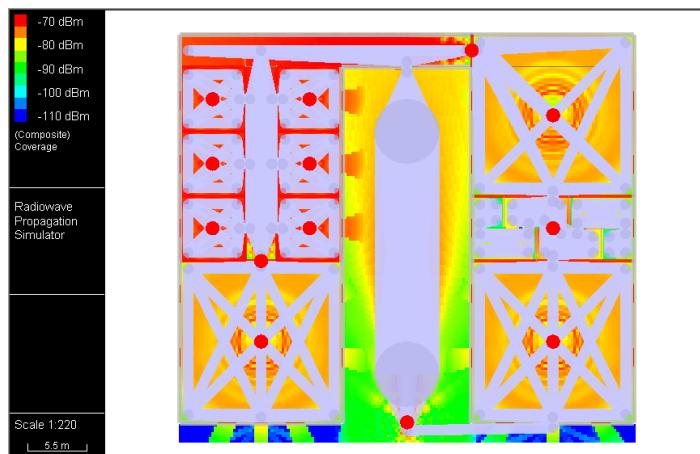


Fig. 8-6 Example dynamic indoor environment



## 9 Support

### 9.1 Electronic Support

Radioplan receives messages via email at **support@radioplan.com**. In most cases, messages will be acknowledged within 48 hours. Each acknowledgement will include an answer or an estimate of when an answer will be forthcoming. Be sure to visit our support center on the World Wide Web. Our web site may answer your question with its updated list of frequently asked questions, technical notes and tips, and helpful hints for using our products. Additionally, minor updates are provided online via the support center. It is accessible for registered users during the subscription period.

### 9.2 Telephone Support

If you have any unanswered questions or discover problems with any part of Radioplan's products or their documentation, please contact your technical support representative at

Phone: +49 (0)351 404 29 - 20

Fax: +49 (0)351 404 29 - 50

Telephone support is available from 9:00 a.m. to 5:00 p.m. CET Monday through Friday.

### 9.3 Support Center

Radioplan's support center offers additional information on our products. It is accessible for registered users until the subscription period has lapsed. Minor updates are provided online via the support center. Major updates are shipped to the customers under subscription immediately via mail.

The support center is reachable at <http://www.radioplan.com/support>.

A login and password is required, that is delivered with RPS to our clients.

### 9.4 Feedback, Comments, Suggestions

Your feedback, comments, and suggestions regarding the software products, their documentation, and our support is greatly appreciated. Feedback from our clients has ever contributed in the permanent improvement of our products and documentations.

Thank you for your contributions.

## 9.5 Contact Information

Internet: <http://www.radioplan.com>

e-mail: [support@radioplan.com](mailto:support@radioplan.com)

RPS Support Center: <http://www.radioplan.com/support>

Radioplan GmbH  
Altmarkt 10  
01067 Dresden  
Germany

Fon: +49 3 51 4 04 29 - 0  
Fax: +49 3 51 4 04 29 - 50

## 9.6 Third Party Components

RPS is developed using the following components of third party vendors:

- The MATLAB C/C++ library is licensed from The Mathworks Inc.
- The omniORB CORBA implementation is provided by Apasphere Ltd.
- The Stingray Studio Library is licensed from Rogue Wave Software, Inc.
- The VectorDraw Drawing Engine is licensed from VectorDraw Software Co.
- RPS can be delivered with hardware dongle protection by MARX Datentechnik GmbH.

## 10 Appendix A

This appendix contains some additional information about RPS. In particular it comprises references to different APIs, import/export data formats, window controls, etc.

### 10.1 Window Controls Reference

#### 10.1.1 Menus

The following overview gives a summary of the available functions that are accessible from the menus.

#### File Menu

|  |        |  |
|--|--------|--|
|  <b>New</b>                 | STRG+N | Creates a new project                            |
|  <b>Open...</b>             | STRG+O | Opens an existing project                        |
|  <b>Save</b>                | STRG+S | Saves the current project                        |
|  <b>Save As...</b>          |        | Saves the current project with a new name        |
| <hr/>  |        | Saves the current plot as image                  |
|  <b>Import</b>              | ▶      | Import functions (see <b>File -&gt; Import</b> ) |
|  <b>Export</b>              | ▶      | Export functions (see <b>File -&gt; Export</b> ) |
|  <b>Print...</b>            | STRG+P | Shows print dialog for current view              |
|  <b>Print Preview...</b>    |        | Shows print preview                              |
|  <b>Printer Setup...</b>    |        | Shows printer setup dialog                       |
| <hr/>  |        |  |
|  <b>1 getting started 5</b> |        |  |
|  <b>2 getting started 4</b> |        |  |
|  <b>3 getting started 3</b> |        |  |
|  <b>4 getting started 2</b> |        |  |
| <hr/>  |        |  |
|  <b>Exit</b>                |        | Quits the application                            |

#### File -> Import

|  |   |
|--|---|
| <b>Import Object List...</b>                               | Imports an OBL file                                 |
| <b>Import DXF or DWG Environment Data...</b>               | Imports a DXF or DWG file                           |
| <b>Import Antenna...</b>                                   | Imports an antenna                                  |
| <hr/>  |   |
| <b>Import Binary DEM Matrix...</b>                         | Imports a binary DEM matrix as raster file          |
| <b>Convert Binary DEM Matrix to Polygons...</b>            | Converts the raster DEM matrix into vector polygons |
| <b>Remove Binary DEM Matrix from Project...</b>            | Removes the raster DEM matrix from the project      |
| <hr/>  |   |
| <b>Import Measurements from Panasonic Area Analyzer...</b> |   |
| <b>Import Measurements from File...</b>                    |   |

## File -> Export

|  |   |
|--|---|
| Export Environment Data as Object List                   | Exports environment data as OBL file                                  |
| Export Environment Data as DXF or DWG                    | Exports environment data as DXF or DWG file                           |
| Export Received Power and Delay Spread                   | Exports received power and delay spread into tab-delimited ASCII file |
| Export Data As ASCII-File                                | Exports impulse response data into tab-delimited ASCII file           |
| Export DOA As ASCII-File                                 | Exports DoA data into tab-delimited ASCII file                        |
| Export Static Data as ASCII-Matrices in Multiple Files   | Exports static results data into tab-delimited ASCII files            |
| Export Static Data as Matrices readable with Matlab(TM)  | Exports static results data into MATLAB file                          |
| Export Dynamic Data as ASCII-Matrices in Multiple Files  | Exports dynamic results data into tab-delimited ASCII files           |
| Export Dynamic Data as Matrices readable with Matlab(TM) | Exports dynamic results data into tab-delimited MATLAB files          |

## Edit Menu

|                      |               |  |
|----------------------|---------------|--|
| Undo                 | ALT+RÜCK      | Undoes the last action (Environment Editor only)                       |
| Redo                 | STRG+Y        | Redoes the last action (Environment Editor only)                       |
| Cut                  | UMSCHALT+ENTF | Cuts out a selection (stored to clipboard)                             |
| Copy                 | STRG+C        | Copies a selection (stored to clipboard)                               |
| Paste                | STRG+V        | Pastes the clipboard contents at the cursor                            |
| Copy Plot as Graphic |               | Copies the current plot to the clipboard as an image (2D/3D view only) |

## Preferences Menu

|  |   |
|--|---|
| General Preferences  | Shows the general preferences dialog                                      |
| Graphical Preferences  | Shows the graphical preferences dialog                                    |
| Detect Diffraction Edges   | Detects all diffraction edges in the environment                          |
| Remove Diffraction Edges   | Removes all diffraction edges from the environment                        |
| <input checked="" type="checkbox"/> Auto Detect Diffraction Edges before each Simulation Run | Checkbox for automatic detection of diffraction edges on simulation start |
| Detect Missing Roofs   | Adds flat roofs to all buildings in the environment                       |
| Remove Receivers Below Roofs   | Removes receivers that are placed below non-vertical objects              |

## View Menu

|                                   |  |
|-----------------------------------|--|
| Projection...                     | Shows the projection dialog                      |
| 2D View                           | Checkbox for 2D view                             |
| 3D View                           | Checkbox for 3D view                             |
| Transmitters                      | Checkbox for viewing transmitters                |
| Receivers                         | Checkbox for viewing receivers                   |
| Objects                           | Checkbox for viewing environment objects         |
| Diffraction Edges                 | Checkbox for viewing diffraction edges           |
| Attraction Areas (Mobility Model) | Checkbox for viewing attraction areas            |
| Rays                              | Checkbox for viewing rays                        |
| Ray Options                       | Ray options (see <b>View -&gt; Ray Options</b> ) |
| Fill Obstacles                    | Checkbox for viewing solid environment polygons  |
| 3D Effects                        | 3D effects (enables fog in submenu)              |

### View -> Ray Options

|   |  |
|---|--|
| Rays for all Receivers  | Shows all saved rays   |
| <input checked="" type="checkbox"/> Complete Ray (All Segments) | Checkbox for viewing complete rays (all segments)              |
| <input type="checkbox"/> First Ray Segment                      | Checkbox for viewing first ray segments only (at transmitters) |
| <input type="checkbox"/> Last Ray Segment                       | Checkbox for viewing last ray segments only (at receivers)     |

### Zoom Menu

|                  |   |  |
|------------------|---|--|
| Zoom In          | q | Shows an enlarged part of the view         |
| Zoom Out         | r | Shows a downsized part of the view         |
| Fit to Window    | s | Fits the view to the window size           |
| Full Screen View | t | Shows the viewing area in full screen mode |

### Simulation Menu

|  |  |
|--|--|
| Run  | Starts a simulation  |
| Stop   | Stops a simulation   |
| Retrieve Intermediate Results                            | Retrieves intermediate results during a simulation                     |
| <input type="checkbox"/> Suggest Incremental Predictions | Checkbox for the incremental prediction mode                           |
| <input type="checkbox"/> Hybrid Mode for Next Simulation | Checkbox for the hybrid simulation mode                                |
| <input checked="" type="checkbox"/> 3D Ray Tracing       | Checkbox for the 3D ray tracing algorithm (local)                      |
| <input type="checkbox"/> 2.5D Ray Tracing                | Checkbox for the 2.5D ray tracing algorithm (local)                    |
| <input type="checkbox"/> 3D Client-Server                | Checkbox for the 3D ray tracing algorithm (Client-Server mode)         |
| <input type="checkbox"/> 2.5D Client-Server              | Checkbox for the 2.5D ray tracing algorithm (Client-Server mode)       |
| <input type="checkbox"/> COST 231 WI (Outdoor)           | Checkbox for the semi-empirical outdoor channel model                  |
| <input type="checkbox"/> COST 231 MWM (Indoor)           | Checkbox for the semi-empirical indoor channel model                   |
| <input type="checkbox"/> Plugin Model (Outdoor Mode)     | Checkbox for the currently selected propagation plug-in (outdoor mode) |
| <input type="checkbox"/> Plugin Model (Indoor Mode)      | Checkbox for the currently selected propagation plug-in (indoor mode)  |
| Run Dynamic Simulation                                   | Starts a dynamic simulation  |

### Results Menu

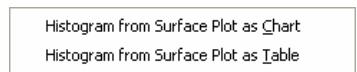
|                |  |
|----------------|--|
| Surface Plots  | Derived surface plots (see <b>Results -&gt; Surface Plots</b> )          |
| Chart Diagrams | Histograms from surface plots (see <b>Results -&gt; Chart Diagrams</b> ) |
| Statistics     | Shows a simulation statistics dialog                                     |
| Remove Results | Removes all result data  |

## Results -> Surface Plots



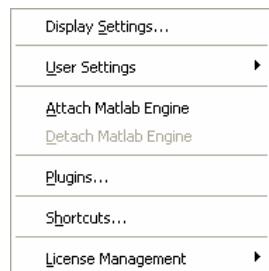
Creates a coverage surface plot  
 Creates a delay spread surface plot  
 Creates a signal-to-interference ratio surface plot  
 Creates a best server surface plot  
 Creates a user-defined surface plot

## Results -> Chart Diagrams



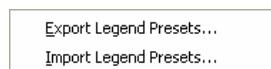
Shows a histogram from the current surface plot as chart  
 Shows a histogram from the current surface plot as table

## Tools Menu



Opens the display settings dialog  
 User-defined settings (see **Tools -> User Settings**)  
 Attaches the MATLAB engine for user-defined plug-ins  
 Detaches the MATLAB engine  
 Shows the plug-in dialog  
 Shows the shortcuts dialog  
 License management (see **Tools -> License Management**)

## Tools -> User Settings



Opens a file save dialog for legend presets export  
 Opens a file open dialog for legend presets import

## Tools -> License Management



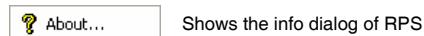
Shows the license registration dialog (only for license file distributions)  
 } Needed for license transfer (only for license file distributions)

## Window Menu



Checkbox for the standard toolbar  
 Checkbox for the status bar  
 Checkbox for the message window  
 Checkbox for the tree window

## Help Menu



Shows the info dialog of RPS

### 10.1.2 Toolbars

#### 10.1.2.1 Standard Toolbar

The standard toolbar contains icons for general file and print functions as well as simulation tasks and data presentation plots. Furthermore, it allows to switch between different edit/presentation modes (Environment Editor, 2D view, 3D view).

In addition to icons given in Fig. 10-1, the standard toolbar contains a list field where the propagation algorithm can be selected.

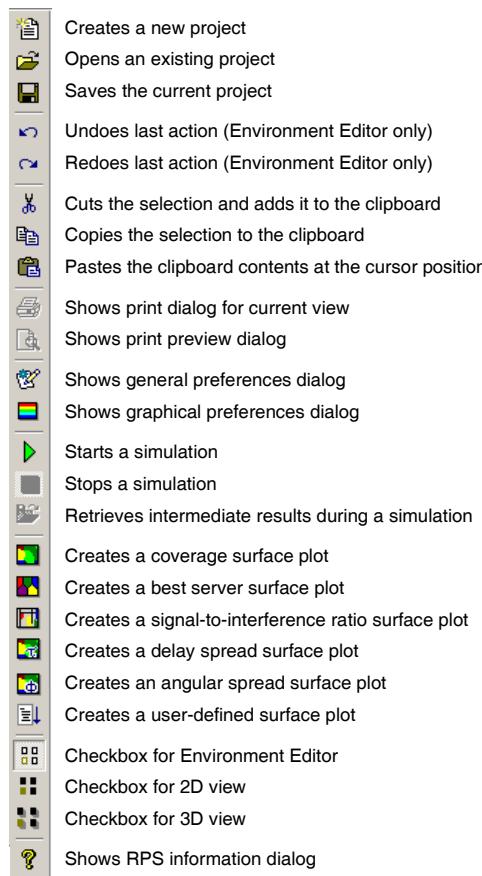


Fig. 10-1 Standard toolbar

#### 10.1.2.2 Component Toolbar

The components toolbar is used to selectively show or hide graphical objects of the environment, the network configuration, and the results.

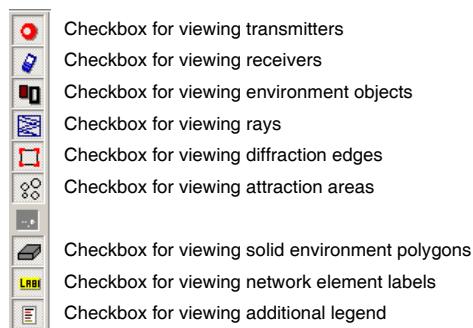


Fig. 10-2 Components toolbar

#### 10.1.2.3 Paint Toolbar

The paint toolbar contains icons for zooming, panning, rotation, and the interactive flying mode that are available in 3D view mode. Additionally, functions for interactive network setup editing, component selection, grid size modification, and result analysis are available using this toolbar.

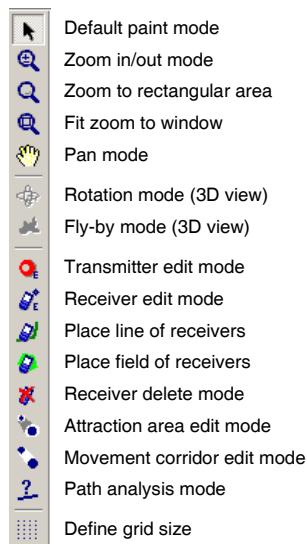


Fig. 10-3 Paint toolbar

## 10.2 Environment Editor Reference

The integrated Environment Editor is a comfortable and comprehensive editor for 3D building environments. Its usage is similar to AutoCAD. Basically it is controlled by commands that are usually followed by a selection of the affected objects (for modifying commands) and eventually the corresponding arguments. The commands can be entered in two different ways:

- as string commands typed in the command line or
- by using the icons of the Environment Editor toolbars.

### 10.2.1 String Command Reference

A complete reference of the available string commands for the Environment Editor is given in Table 10-1. These commands can be typed in the command line below the viewing area of the Environment Editor, completed by <Enter>. After that, the possible selection of affected objects and the input of further arguments is queried by corresponding requests in the status bar of the Environment Editor. These inputs can either be made graphically with the mouse cursor or by typing coordinates etc. in the command line.

Table 10-1 Command line strings for Environment Editor

| <i>Command</i>                        | <i>Description</i>  |
|---------------------------------------|---|
| <b>Zooming</b>                        |   |
| ZI                                    | Zoom In   |
| ZO                                    | Zoom Out  |
| ZE                                    | Zoom Extents  |
| ZP                                    | Zoom Previous   |
| ZW                                    | Zoom Window   |
| ZA                                    | Zoom All  |
| <i>Inline, i.e. during active job</i> |   |
| 'ZI                                   | Zoom In   |
| 'ZO                                   | Zoom Out  |
| 'ZE                                   | Zoom Extents  |
| 'ZP                                   | Zoom Previous   |
| 'ZW                                   | Zoom Window   |
| <b>Screen/Undo</b>                    |   |
| REGEN                                 | Rebuild the whole drawing   |
| REDRAW                                | Quick redraw  |
| PAN                                   | Shift current view  |
| U                                     | Undo  |
| REDO                                  | Redo  |
| <b>Editing</b>                        |   |
| COPY                                  | Copy function   |
| MOVE                                  | Move function   |
| ERASE                                 | Delete entity   |
| PURGE                                 | Remove unreferenced blocks and/or layers                          |
| SELECT                                | Select objects  |
| SCALE                                 | Scale an entity according to a base point                         |
| EXPLODE                               | Decompose a block into its entities                               |
| PROP                                  | Property for selection  |
| DIST                                  | Distance between two points                                       |
| ROTATE                                | Rotate entities in 3D space (Please use world coordinate system!) |
| <b>Drawing</b>                        |   |
| PLINE                                 | Polyline (2D: won't transfer to RPS views)                        |

| <i>Command</i>          | <i>Description</i>                                    |
|-------------------------|---|
| 3DWALL                  | Series of vertical walls of given height              |
| 3DFACE                  | 3D surface with 4 vertices                            |
| 3DPOLYGON               | 3D surface with arbitrary number of vertices          |
| 3DMESH                  | MxN matrix of 3Dfaces                                 |
| BOX                     | Cube (base point, length, width, height, rotation)    |
| CONE                    | Cone or cylinder (base point, 2 radii, # of segments) |
| SPHERE                  | Sphere (base point, radius, # of segments)            |
| PL2MESH                 | Convert Polyline to 3Dmesh (results in a panel/plate) |
| IMAGE                   | Insert Image (BMP, JPG, TIF, GIF)                     |
| <b>Block Operations</b> |   |
| MBLOCK                  | Group entities to named block                         |
| WBLOCK                  | Export existing block (or define a new one) to file   |
| INSERT                  | Insert block, which exists in current file            |
| BLOCKLIB                | Choose a block from library path                      |
| ARRAY                   | Manifold entities in (P)olar or (R)ectangular shape   |

### 10.2.2 Environment Editor Toolbars

The Environment Editor also provides some additional toolbars for direct access to all its commands and functions. They are shown in Fig. 10-4 to Fig. 10-6. The usage is similar to the command line strings, i.e. the user is queried for further arguments by according requests that are displayed in the status bar of the Environment Editor.

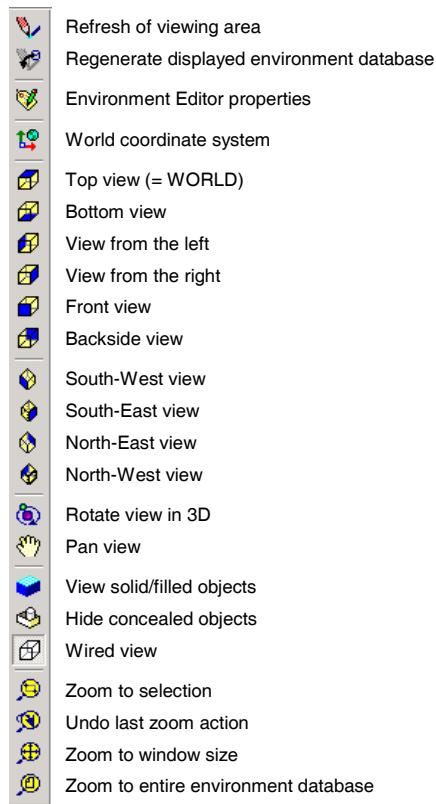


Fig. 10-4 View and zoom functions

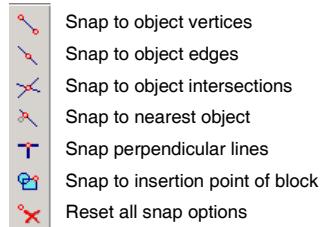


Fig. 10-5 Object snap functions

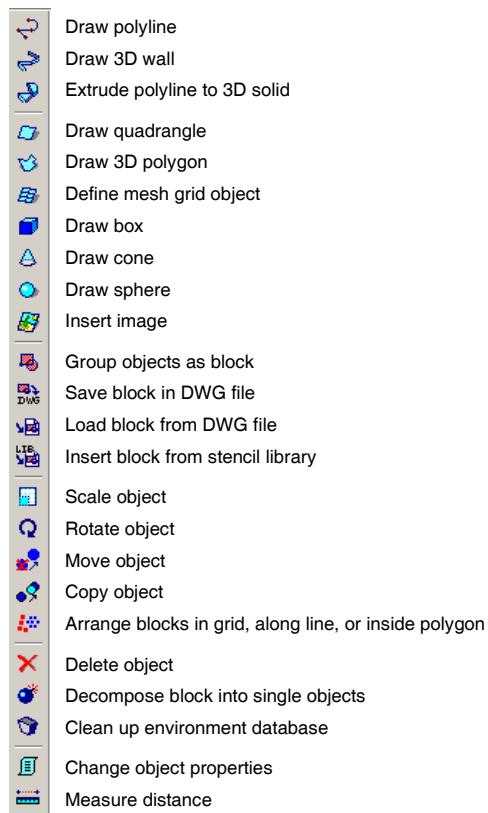


Fig. 10-6 Object draw and modify commands

### 10.3 Stencil Library Overview

A block library of often-used obstacles is delivered with RPS. These blocks are summarized in Table 10-2. The insertion point of the blocks is indicated by a red spot that is aligned at the bottom level of the blocks.

Each of the blocks has an own material layer stored in the corresponding DWG file. This layer is automatically created when the respective block is inserted. The stencil library is located in the directory %ApplicationPath%\RPS\library\Stencils\. The subdirectories are used as categories in the stencil library dialog, see Section 4.1.9.3.

Note that the material properties of the material layers cannot be stored in the DWG files. Thus, on importing a block, these properties have to be edited manually. It is recommended to enable the Filled flag for all types of solid obstacles. Hence, the permittivity values should also be set to reasonable values.

Table 10-2 Sample blocks in the stencil library (red spot: insertion point)

| Block Name         | Preview | Description                                      |
|--------------------|---------|--|
| <b>Advertising</b> |         |  |
| BILLBOARD5_00      |         | billboard 5m x 3m, thickness 0.2m, rack 1m high  |
| BILLBOARD10_00     |         | billboard 10m x 6m, thickness 0.4m, rack 2m high |
| BILLBOARD15_00     |         | billboard 15m x 9m, thickness 0.6m, rack 3m high |
| <b>Persons</b>     |         |  |
| PERSON_BASE        |         | person template, 1m tall                         |
| PERSON1_60         |         | human body, 1.60m tall                           |
| PERSON1_75         |         | human body, 1.75m tall                           |
| PERSON1_90         |         | human body, 1.90m tall                           |
| CAR4_50            |         | car 4.5m x 1.8m, 1.66m high                      |
| <b>Trees</b>       |         |  |
| TREE2_90           |         | tree, 2.90m high                                 |
| TREE4_50           |         | tree, 4.50m high                                 |
| TREE6_50           |         | tree, 6.50m high                                 |

#### 10.4 RPS File Format

The RPS file format contains the entire network configuration, all simulation parameters, simulation results, ray paths etc. It is an ASCII based file to make it portable for both Windows and Unix platforms.

The RPS format consists of lines beginning with a specifier. This specifier describes the type of database entry defined by the line. Each line is divided into columns separated by tabulators. The first line identifies the RPS file format, which is internally used for

compatibility to former versions of RPS. The files are organized into sections. Each section contains a descriptive header.

Along with the headers (comment lines) it should be possible to understand the entries in the RPS files. For data extraction we recommend to use our export files and the MATLAB interface. Please note, that the RPS file format could be changed significantly due to the introduction of new features in upcoming versions of RPS.

## 10.5 OBL File Format

The OBL (Object List) file format of RPS is an ASCII format for vector data. It supports the import of real 3D polygons. A polygon can be modeled by at least 3 three-dimensional vertices (XYZ-Points). Each polygon can be assigned to a material layer represented by its name. Moreover, a label can be added, for instance 'xxx building'.

The OBL file format is a simple ASCII tab-delimited file. An example of the file structure is demonstrated in Fig. 10-7.

```

6 FILEVERSION
C Polygons: LayerName Label VertexCount X0 Y0 Z0 X1 Y1 Z1 ...
F CUSTOMLAYER0 Building1 4 1390 575 0 1390 575 16 1405 480
    16 1405 480 0
F CUSTOMLAYER0 Building1 4 1405 480 0 1405 480 16 1415 480
    16 1415 480 0
F CUSTOMLAYER0 Building1 4 1415 480 0 1415 480 16 1420 460
    16 1420 460 0
F CUSTOMLAYER0 Building1 4 1420 460 0 1420 460 16 1445 465
    16 1445 465 0
C END

```

Fig. 10-7 Sample OBL file

An "F" as the first character of a line indicates a polygon specification. A "C" represents a comment line. The file should contain a comment as last line, for instance "C END".

If polygons with more than four vertices are imported, they are automatically decomposed into triangles.

The OBL format can be used as an exchange format to transform building databases that are not available in DXF or DWG format into a format readable by RPS. Its simple structure favors an automated format conversion.

## 10.6 Antenna File Format

In general, the radiation pattern of an antenna  $C(\theta, \varphi)$  is three-dimensional. In most cases only a vertical and a horizontal cut of this 3D antenna diagram is available and can be used to approximate a three-dimensional antenna beam pattern.

The antenna data is stored in three files: a header file (extension .cfg) and two data files (extension .dat) for the horizontal and vertical antenna diagrams. The header file contains the ID and a description for the antenna and its gain. Moreover it has references to the associated data files of the horizontal and vertical diagrams. Each of the data files consists of a collection of lines, each giving the angle, the attenuation in dB and a (optional) phase. Comment lines start with a '#'. Two different data formats are possible:

Format 1: <angle> [tab] <attenuation> [tab] <optional phase>

Format 2: <angle> = <attenuation> , <phase>

Format 1 (RPS Format) is a common tab-delimited file format. The 2<sup>nd</sup> format type is compatible with the antenna format used by Radioplan's dynamic UMTS simulation platform "WiNeS". The angle in Format 2 is separated from the attenuation value by a "=" character, and an optional phase information can be added separated by a comma. The type of format used for the antenna patterns can be specified by the parameter:

Format = WINES (default value), or

Format = RPS,

respectively.

Note that the attenuations must be normalized to zero dB. The gain is given in dBi, that means in relation to an isotropic source.

The interpolation formula for the 3-dimensional antenna pattern has been improved compared to earlier releases of RPS. The current version is given by:

$$A[dB] = Gain[dB] - \left( C_{\varphi}(\varphi) + (C_g(\vartheta) - C_{\varphi}(0^\circ)) \cdot D + (C_g(180^\circ - \vartheta) - C_{\varphi}(180^\circ)) \cdot (1 - D) \right)$$

with

$$D = \frac{|180^\circ - \varphi|}{180^\circ}$$

$C_{\varphi}(\varphi)$  : relative magnitude in dB to C(0°) from 0° to 360° (horizontal diagram)

$C_g(\vartheta)$  : relative magnitude in dB to C(0°) from -90° to 90° (vertical diagram)

Basically, this formula adds the logarithmic attenuations given by the horizontal and vertical antenna diagram. Depending on  $\varphi$ , the vertical attenuation values for the front and back half plane are combined, where the front half plane is normalized with the horizontal diagram at  $\varphi = 0^\circ$ , and the back half plane is normalized with the horizontal diagram at  $\varphi = 180^\circ$ . This normalization avoids errors, if the vertical and horizontal diagrams do not fit exactly. For some special, mainly omni-directional antenna diagrams, this normalization cannot be applied. Therefore, a parameter

CombinationFlag = [0/1]

can be defined in the header file. If this parameter is set to 1, the vertical half planes are not normalized and the interpolation formula is then given by:

$$A[dB] = Gain[dB] - \left( C_{\varphi}(\varphi) + C_g(\vartheta) \cdot D + C_g(180^\circ - \vartheta) \cdot (1 - D) \right)$$

## 10.7 Supported Raster Data File Formats

A large number of raster data formats are supported for import. They can either be loaded as image layers or as network configuration matrix for DEM, morphology (clutter), traffic, or pathloss. The most common of these file formats are briefly introduced in the following Subsections with some remarks on special features and their georeferencing ability.

### 10.7.1 TIFF / GeoTIFF Format

Most forms of TIFF and GeoTIFF files are supported where the latter ones are georeferenced. All common GeoTIFF projections are supported. Georeferencing from GeoTIFF is supported in the form of one tie point and pixel size, a transformation matrix, or a list of GCPs. In order to use some uncommon projected and geographic coordinate systems and translate them into OGC WKT, it is necessary to have the EPSG \*.csv files available. They must be found at the location pointed to by the GEOTIFF\_CSV environment variable.

If no georeferencing information is available in the TIFF file itself, an ESRI world file (see Section 10.7.1.1) with the extension \*.tfw, \*.tiffw or \*.wld, as well as a MapInfo \*.tab file (only control points used, coordinate system ignored) could be used.

#### 10.7.1.1 ESRI World File Format

The ESRI World File format is an ASCII format to associate an image file with a geographical reference. It contains information about the absolute location of the image as well as the scaling and alignment of the image pixels. The usual extension for such a header file is \*.wld or \*.tfw. Usually, the accompanied image file is given in the TIFF format. However, this is not an exclusive condition.

A \*.wld file consists of six lines with a numeric value in each line. The first two lines specify the pixel scaling vector for the x-direction, the third and fourth lines give the pixel scaling vector in y-direction, and the last two lines locate the orientation point (usually the upper left corner) of the image. In either of these line pairs, the first line contains the x-coordinate and the latter line contains the y-coordinate of the vector or the position, respectively.

Since RPS uses a Cartesian coordinate system internally, the values in the second and third lines must be zero. Thus it is ensured that the image is aligned to the x- and y-axes of the underlying coordinate system. If the orientation point is the upper left corner of the image, the value in the fourth line (pixel scaling vector in y-direction) must be negative because the y-vector points downward then. The values are assumed being given in meters.

### 10.7.2 BIL File Format

The BIL file format represents one or more matrices in binary format. A \*.hdr file is used to specify the exact data format of the binary matrix (\*.bil file). RPS supports both 8 and 16 bit precision formats with "I" or "M" byte order formats. Currently the number of bands defined within the BIL matrix is restricted to one.

### 10.7.3 Arc/Info Raster Formats

#### 10.7.3.1 Arc/Info ASCII Interchange Format

This format is the ASCII interchange format for Arc/Info Grid, and takes the form of an ASCII file, plus sometimes an associated \*.prj file containing information on an affine georeferencing transform and some projections.

Grids are treated as signed 16bit integer unless the first scan line contains decimal values in which case the image is treated as 32bit floating point.

### 10.7.3.2 Arc/Info Binary Grid Format

This format is the internal binary format for Arc/Info Grid incl. information on an affine georeferencing transform and some projections, and takes the form of a coverage level directory in an Arc/Info database. To open the coverage select the coverage directory, or an \*.adf file (such as hdr.adf) from within it.

### 10.7.4 EOSAT Fast Format

EOSAT datasets normally consist of one or more raster data files (e.g. BAND1.DAT, BAND2.DAT, ...) and a header file which must be called HEADER.DAT. To open the dataset select the HEADER.DAT file.

Only the raster data can be accessed. Georeferencing and other meta data is ignored.

### 10.7.5 Erdas Imagine Format

The Erdas Imagine \*.img format is supported with some limitations. Pyramid levels and metadata are ignored, but palettes and georeferencing is captured. It supports the Erdas band types u8, s8, u16, s16, u32, s32, f32, f64, c64 and c128.

### 10.7.6 GIF File Format

Normal and interlaced GIF files are supported. GIF files always appear as having one color-mapped eight bit band. Generally, GIF files have no integrated support for georeferencing. However, if an ESRI world file exists with the \*.wld extension (see Section 10.7.1.1), it will be read and used to establish the geotransform for the image.

### 10.7.7 Grid eXchange File Format

This is a raster exchange format propagated by Geosoft, and made a standard in the gravity/magnetic field. The format also includes support for georeferencing information and projections.

### 10.7.8 Hierarchical Data Format (Release 4)

There are two HDF formats, HDF4 (4.x and previous releases) and HDF5. These formats are completely different and *not* compatible. Here, only HDF4 file import is supported. NASA's Earth Observing System (EOS) maintains its own HDF modification called HDF-EOS which is also supported for import. This modification is suited for use with remote sensing data and fully compatible with underlying HDF.

Hierarchical Data Format is a container for several different datasets. For data storing, Scientific Datasets (SDS) are most often used. SDS is a multidimensional array filled by data. One HDF file may contain several different SDS arrays. They may differ in size, number of dimensions, and may represent data for different regions. The import of remote sensing and geospatial datasets in form of raster images is supported.

For georeferencing the HDF files, the ASTER Level 1B and ASTER DEM formats are supported.

### 10.7.9 Japanese DEM Format

Japanese DEM files can be imported, normally having the extension \*.mem. These files are a product of the Japanese Geographic Survey Institute.

These files are represented as having one 32bit floating band with elevation data. The georeferencing of the files is recognized as well as the coordinate system (always latitude/longitude on the Tokyo datum).

#### **10.7.10 JPEG File Format**

The widely used JPEG JFIF format is supported. JPEG files are represented as one band (grey scale) or three band (RGB) datasets with Byte valued bands.

There is no direct support for georeferencing information or metadata for JPEG files. But if an ESRI world file exists with the \*.wld or \*.jgw suffices (see Section 10.7.1.1), it will be read and used to establish the geotransform for the image. Overviews can be built for JPEG files as an external \*.ovr file.

Note that also tiled TIFF images with JPEG compressed tiles can be imported.

#### **10.7.11 Atlantis MFF Raster File Format**

Atlantis Scientific's MFF datasets consist of a header file (typically with the extension \*.hdr) and a set of data files with extensions like \*.x00, \*.b00 and so on. To open a dataset select the \*.hdr file.

Reading latitude/longitude GCPs (TOP\_LEFT\_CORNER, ...) is supported but there is no support for reading affine georeferencing or projection information. Unrecognized keywords from the \*.hdr file are ignored.

Diverse data types are supported, including 8, 16, 32 and 64 bit data precisions in integer, real and complex data types. In addition, tile organized files (as produced by the Atlantis SAR Processor - APP) can be imported.

MFF files are not normally portable between systems with different byte orders. However, the keyword BYTE\_ORDER can be used which can take a value of LSB (little-endian), and MSB (big-endian). This may be manually added to the \*.hdr file if required.

#### **10.7.12 PCI Labeled Raw Format**

The PCI \*.aux labeled raw raster file format of all PCI data types (8U, 16U, 16S, and 32R) is supported. To open a PCI labeled file, select the raw data file itself. The \*.aux file (which must have a common base name) will be checked for automatically.

Georeferencing, projections, and other metadata is ignored.

#### **10.7.13 Portable Network Graphics**

Grey scale, pseudo-colored, palettes, RGB and RGBA PNG files (with extension \*.png) are supported as well as precisions of eight and sixteen bits per sample.

ESRI world files with the extensions of \*.wld, \*.tfw or \*.tifw (see Section 10.7.1.1) will be read for georeferencing the PNG file. Single transparency values in grey scale files will be recognized as a nodata value.

#### **10.7.14 USGS DOQ Format**

The USGS (US Geological Survey) DOQ (Digital Ortho Quad) format is supported, including reading of an affine georeferencing transform, and capture of the projection string. Both the old, unlabeled DOQ format and the new, labeled DOQ format can be handled.

#### **10.7.15 USGS SDTS DEM Format**

Support for reading USGS SDTS (Spatial Data Transfer Standard) formatted DEMs is included. USGS DEMs always have a data type of signed sixteen bit integer. Projection and georeferencing information can also be extracted.

SDTS datasets consist of a number of files. Each DEM should have one file with a name like XXXCATD.DDF. This should be selected to open the dataset.

#### **10.7.16 X11Pixmap Format**

XPM (X11Pixmap Format) image files are color-mapped one band images primarily used for simple graphics purposes in X11 applications.

The XPM interface does not support georeferencing (not available from XPM files) nor does it support XPM files with more than one character per pixel. New XPM files must be color-mapped or grey scale, and color tables will be reduced to about 70 colors automatically.



## 11 Appendix B

### 11.1 Dielectric Properties of Selected Materials

In the following table dielectric properties of selected materials including there references are shown. Please note that the permittivity is frequency dependent, especially the imaginary part  $\epsilon_r''$ .

Table 11.1 Dielectric properties of selected materials

| Material                              | f in GHz | d in cm | $\epsilon_r'$ | $\epsilon_r''$ | Source   |
|---------------------------------------|----------|---------|---------------|----------------|----------|
| Reinforced concrete, outer wall       | 1.8      | 40      | 5             | 0.4            | [Läh94]  |
| Reinforced concrete, inner wall       | 1.8      | 15/10   | 5             | 0.4            | [Läh94]  |
| Reinforced concrete, different floors | 1.29     |         | 3.9-5.4       |                | [Hor96]  |
| Reinforced concrete, pillar           | 1.8      | 30      | 5             | 0.4            | [Läh94]  |
| Concrete ( $\rho=0.7\text{kg/dm}^3$ ) | 1        | 20      | 2.0           | 0.5            | [Cic94a] |
| Concrete                              | 1        | 24      | 7             | 0.8            | [Cic94b] |
| Concrete                              | 1.8      | 40      | 3.5           | 0.9            | [Bac96]  |
| Concrete, floor and ceiling           | 1.8      | 50      | 5             | 0.4            | [Läh94]  |
| Brick, inner wall                     | 1.8      | 10      | 4             | 0.4            | [Läh94]  |
| Metal, inner wall                     | 1.8      | 3       | 1             | $>10^6$        | [Läh94]  |
| Wood                                  | 1        | 5       | 2.5           | 0.03           | [Cic94b] |
| Plaster, inner wall                   | 1.8      | 10      | 2.5           | 0.3            | [Läh94]  |
| Plaster, inner wall                   | 1.8      | 1.3     | 2.5           | 0.1            | [Läh94]  |
| Glass                                 | 1        | 1       | 6.0           | 0.05           | [Cic94b] |
| Glass window, 3 plates                | 1.8      |         | 19            | 0.1            | [Bac96]  |
| Plastics                              | 1        | 10      | 4.0           | 0.4            | [Cic94a] |
| Plastics                              | 1        | 13      | 4.1           | 0.4            | [Cic94b] |

More dielectric constants are available online in the RPS Support Center at <http://www.radioplan.com/support> (restricted access for RPS customers).

## 11.2 Notational Conventions

|   |   |
|---|---|
| $\vec{p}$   | Polarization of the electromagnetic field strength with respect to the propagation direction $\vec{r}$                |
| $\vec{r}$   | Propagation direction of the electromagnetic wave (represented by a ray)  |
| $f$   | Frequency in GHz, if not other defined  |
| $d$   | Distance between transmitter and receiver in m, if not other defined  |
| $\underline{n}$                                     | Refractive index ( $\underline{n} = \sqrt{\underline{\epsilon}_r}$ )  |
| $\alpha$  | Incident angles   |
| $\underline{\epsilon}_r = \epsilon' - j \epsilon' $ | complex permittivity including real and imaginary part  |
| $\underline{h}(x, y, z, \tau)$                      | site specific channel impulse response  |
| $P_r(x, y, z)$                                      | Received power at the position r(x,y,z)   |
| $P_t(x, y, z)$                                      | Transmitted power at position s(x,y,z)  |
| $S_r(x, y, z)$                                      | Delay spread at position (x,y,z)  |
| $C_t(\vartheta, \varphi), C_r(\vartheta, \varphi)$  | Radiation patterns of transmitting and receiving antennas   |
| $G_t, G_r$  | Gain of a transmitter or receiving antenna compared to a isotropic source (EIRP, Equivalent Isotropic Radiated Power) |
| $\underline{\Gamma}$                                | Reflection coefficient  |
| $\underline{D}$                                     | Diffraction coefficient   |
| $p(aa), p(d)$                                       | transition probabilities of attraction areas  |

## 11.3 Abbreviations and Acronyms

|       |  |
|-------|--|
| CDF   | Cumulative Distribution Function                                       |
| COM   | Components Object Model  |
| CORBA | Common Object Request Broker Architecture                              |
| COST  | European Cooperation in the Field of Scientific and Technical Research |
| DEM   | Digital Elevation Model  |

|            |                                     |
|------------|-------------------------------------|
| DLL        | Dynamic Link Library                |
| ECEF       | Earth Centered, Earth Fixed         |
| EIRP       | Equivalent Isotropic Radiated Power |
| EPSG       | European Petroleum Survey Group     |
| GIS        | Geographical Information System     |
| GO         | Geometrical Optics                  |
| GUI        | Graphical User Interface            |
| HP         | Horizontal Polarization             |
| LHCP       | Left Hand Circular Polarization     |
| ORB        | Object Request Broker               |
| PDF        | Probability Density Function        |
| RHCP       | Right Hand Circular Polarization    |
| RPS        | Radiowave Propagation Simulator     |
| UTD        | Uniform Theory of Diffraction       |
| VP         | Vertical Polarization               |
| WCS        | World Coordinate System             |
| WGS        | World Geographic System             |
| WKT Format | Well Known Text Format              |

## 11.4 References

- [Läh94] Lähteenmäki, J.: Testing and verification of indoor propagation models. COST-231 TD (94) 111, Darmstadt, Sep. 6-8, 1994
- [Hor96] Horikoshi, J.; Tanaka, K.; Morinaga, T.: 1.2 GHz band wave propagation measurements in concrete building for indoor radio communications. IEEE Trans. Veh. Technol., vol. VT-35, no. 4, Nov. 1996, pp. 146-151
- [Cic94a] Cichon, D. J.: Strahlenoptische Modellierung der Wellenausbreitung in urbanen Mikro- und Picofunkzellen. Forschungsberichte aus dem Institut für Höchstfrequenztechnik und Elektronik der Universität Karlsruhe, Band 8, 1994
- [Cic94b] Cichon, D. J.; Becker, T.; Wiesbeck, W.: A ray launching approach for indoor and outdoor applications. COST 231 TD(94) 32, Lisbon, Portugal, Jan. 18-21, 1994

- [Bac96] De Backer, B.; Börjeson, H.; Olyslager, F.; De Zutter, D.: The study of wave-propagation through a windowed wall at 1.8 GHz. 1996 IEEE 46th Vehicular Technology Conference, vol. 1, pp. 165-169
- [Hue97] Hübner, J: Radio propagation prediction in picocells at 60 GHz- comparison of simulation and measurement results. MMMCOM Workshop Dresden, 12.-13. May 1997, pp. 75-79
- [Voi99a] Voigt, J.; Hübner, J.; Förster, E.; Fettweis, G. P.: "Design Pattern for a Single Frequency TDMA-System in a Typical Office Environment at 60 GHz", Proceedings of IEEE Vehicular Technology Conference (VTC'99), pp. 1570-1574, Houston, TX, May 1999.
- [Voi99b] Voigt, J.; Hübner, J.; Förster, E.; Fettweis, G. P.: "Investigation of Wireless Communications Systems using Dynamic Simulation Models including Simultaneous Field Strength Prediction", Proceedings of European Wireless'99 and 4. ITG-Convention on "Mobile Communications", Munich, Germany, October 1999.

## 12 Index

### A

|                             |                |
|-----------------------------|----------------|
| Alpha Blending .....        | 60             |
| Angular Spread.....         | 181            |
| Antenna                     |                |
| Antenna Diagram.....        | 46, 123        |
| Assign to Receivers .....   | 122            |
| Assign to Transmitters..... | 46             |
| Azimuth .....               | 200            |
| Definition .....            | 123, 264       |
| Downtilt.....               | 56, 120, 198   |
| Import .....                | 216            |
| Polarization .....          | 81             |
| Template .....              | 120            |
| Automation Interface .....  | <i>See COM</i> |

### B

|                        |                        |
|------------------------|------------------------|
| Background Image ..... | 91                     |
| Bandwidth .....        | 151                    |
| Base Station .....     | <i>See Transmitter</i> |
| Best Serving Cell..... | 179                    |

### C

|                             |                                     |
|-----------------------------|-------------------------------------|
| Center Frequency.....       | <i>See Frequency</i>                |
| Chart Plots .....           | 209                                 |
| Client-Server Mode .        | <i>See Parallelized Ray Tracing</i> |
| COM                         |                                     |
| Automation Example .....    | 241                                 |
| Automation Interface.....   | 229                                 |
| Specification.....          | 230                                 |
| Configuration               |                                     |
| Environment.....            | <i>See Environment Editor</i>       |
| Network Setup .....         | 71, 120                             |
| Receiver .....              | 40                                  |
| Simulation .....            | 147                                 |
| Transmitter .....           | 39, 124                             |
| Contact Information.....    | 252                                 |
| Contour Plots .....         | 182                                 |
| Coordinate System .....     | 53                                  |
| in Environment Editor ..... | 89                                  |
| Transformation.....         | 53                                  |
| Coverage .....              | 178                                 |

### D

|   |        |
|---|--------|
| Delay Spread.....                       | 180    |
| DEM Layer.....                          | 58, 96 |
| Dielectric Properties of Materials..... | 271    |
| Diffraction                             |        |
| Definition .....                        | 80     |
| Diffraction Edges                       |        |
| Autodetection.....                      | 119    |
| Definition .....                        | 70     |
| DXF/DWG Files                           |        |
| Export.....                             | 219    |
| Import .....                            | 217    |
| Dynamic Simulation .....                | 169    |
| Mobility Model.....                     | 169    |

### E

|                                      |            |
|--------------------------------------|------------|
| Empirical Simulation Models .....    | 143        |
| Environment Editor                   |            |
| Add a new Building.....              | 38, 99–103 |
| Add Flat Roofs.....                  | 119        |
| Add new Obstacles .....              | 108–16     |
| Background Image .....               | 91         |
| Blocks .....                         | 104        |
| Command Overview .....               | 259        |
| Coordinate Systems .....             | 89         |
| Distance Measurement.....            | 118        |
| Environment Objects .....            | 69         |
| Export of Environment.....           | 104        |
| Import of Environment .....          | 104        |
| Material Database .....              | 98         |
| Material Layers .....                | 57, 93     |
| Object Properties.....               | 116        |
| Object Snap .....                    | 92         |
| Overview .....                       | 87         |
| relative position with '@' .....     | 103        |
| Stencil Library .....                | 105, 262   |
| Toolbars .....                       | 260        |
| Undo Buffer.....                     | 53         |
| Usage .....                          | 88         |
| Views .....                          | 89         |
| Wobble Function .....                | 110        |
| Examples                             |            |
| Antenna Diagrams .....               | 245        |
| Dresden City Center Environment..... | 247        |
| Dynamic Indoor Example.....          | 249        |

|  |                            |
|--|----------------------------|
| Dynamic Outdoor Example.....                   | 248                        |
| Pico Cellular Indoor Environment.....          | 246                        |
| Simple Indoor Environment .....                | 245                        |
| Suburban Outdoor Environment.....              | 248                        |
| Urban Outdoor Environment.....                 | 246                        |
| <b>Export</b>                                  |                            |
| ASCII File Export.....                         | 220                        |
| Chart Plots .....                              | 211                        |
| Copying Table Data .....                       | 214                        |
| Directions of Arrival .....                    | 220                        |
| DXF/DWG Files.....                             | 219                        |
| Dynamic Channel Data.....                      | 173, 226                   |
| OBL Files .....                                | 219                        |
| Path Analysis.....                             | 221                        |
| Received Power and Delay Spread .....          | 219                        |
| Simulation Results.....                        | 219                        |
| Static Channel Data .....                      | 223                        |
| Surface Plot as TIFF Image.....                | 185                        |
| <b>F</b>                                       |                            |
| <b>File Formats</b>                            |                            |
| Antenna File Formats.....                      | 264                        |
| OBL File Format.....                           | 264                        |
| Raster Data ~ .....                            | 265                        |
| RPS File Format.....                           | 263                        |
| <b>Fly-by Mode</b> .....                       | 75                         |
| <b>Frequency</b>                               |                            |
| ~ @ Transmitter .....                          | 128                        |
| ~ Template .....                               | 151                        |
| <b>G</b>                                       |                            |
| <b>Geometrical Optics</b> .....                | 78                         |
| <b>Getting Started</b> .....                   | 37                         |
| <b>Graphical User Interface</b>                |                            |
| 2D and 3D Views .....                          | 73                         |
| Clipboard Functions .....                      | 53                         |
| Fly-by Mode.....                               | 75                         |
| Full Screen View .....                         | 53                         |
| Key Navigation in 3D Views.....                | 77                         |
| Menus .....                                    | 253                        |
| Message Window.....                            | 52                         |
| Moving Graphics .....                          | <i>See Pan Mode</i>        |
| Overview.....                                  | 49                         |
| Rotation in 3D Views.....                      | 75                         |
| Show and Hide Graphical Components.....        | 52                         |
| Status Bar.....                                | 52                         |
| Summary .....                                  | 17                         |
| Surface Plots .....                            | 177                        |
| Toolbars.....                                  | 51, 257                    |
| Tree Window .....                              | 51                         |
| Viewing Area .....                             | 52                         |
| Zoom Functions .....                           | 74                         |
| <b>Grid Size</b> .....                         | 39, 124, 134               |
| <b>H</b>                                       |                            |
| <b>Hardware Dongle</b> .....                   | 22, 31                     |
| <b>Histogram</b> .....                         | 185                        |
| <b>Hybrid Mode</b> .....                       | 147                        |
| <b>I</b>                                       |                            |
| <b>Import</b>                                  |                            |
| Antenna .....                                  | 216                        |
| DXF/DWG Files .....                            | 217                        |
| OBL Files.....                                 | 218                        |
| Raster File ~ .....                            | 265                        |
| <b>Installation</b>                            |                            |
| Directories on the Installation CD-ROM .....   | 21                         |
| Installation Process .....                     | 22                         |
| OmniNames CORBA Name Server .....              | 30, 157                    |
| Plug-in Directories.....                       | 83                         |
| Setup .....                                    | 26                         |
| Upgrade.....                                   | 20                         |
| <b>Interfaces</b>                              |                            |
| Summary .....                                  | 18                         |
| <b>K</b>                                       |                            |
| <b>Key Navigation in 3D Views</b> .....        | 77                         |
| <b>L</b>                                       |                            |
| <b>Layers</b>                                  |                            |
| Coinciding Result Layers .....                 | 182                        |
| Color Settings .....                           | 66                         |
| DEM Layer .....                                | 103                        |
| Layer Concept.....                             | 57                         |
| Layer Settings Dialog .....                    | 44                         |
| Legend Presets .....                           | 69                         |
| Material Layers.....                           | 57, 93                     |
| Result Layers .....                            | 59, 178                    |
| Settings .....                                 | 63                         |
| <b>License</b>                                 |                            |
| ~ File .....                                   | 34                         |
| ~ Transfer .....                               | 35                         |
| Configuration .....                            | 29                         |
| Floating ~ .....                               | 32                         |
| Hardware Dongle.....                           | <i>See Hardware Dongle</i> |
| Registration .....                             | 35                         |
| <b>Load Balancing</b> .....                    | 162                        |
| <b>Local Processor Units</b> .....             | 148                        |
| <b>LOS Preview Plot</b> .....                  | 131, 198                   |
| <b>M</b>                                       |                            |
| <b>Material Database</b> .....                 | 271                        |
| <b>MATLAB</b>                                  |                            |
| Attaching and Detaching the MATLAB Engine..... | 84                         |
| Data Export .....                              | 224, 227                   |
| MATLAB Scripts .....                           | 85                         |
| <b>Menu Bar</b>                                |                            |
| Customize .....                                | 50                         |
| <b>Menus</b> .....                             | 253                        |
| <b>Mobile Station</b> .....                    | <i>See Receiver</i>        |
| <b>Mobility Model</b> .....                    | 169                        |
| <b>Moving Graphics</b> .....                   | <i>See Pan Mode</i>        |
| <b>N</b>                                       |                            |
| <b>Navigation in 3D Views</b> .....            | 75                         |
| <b>Noise Floor</b> .....                       | 152                        |

**O****Objects**

|                                |          |
|--------------------------------|----------|
| 3D Object .....                | 102      |
| 3DFace .....                   | 99       |
| Add a new Building .....       | 38       |
| Box .....                      | 104      |
| Cone .....                     | 104      |
| Definition .....               | 69       |
| Material Layer Selection ..... | 99       |
| Modify Objects .....           | 108      |
| Polyface .....                 | 104      |
| Polyline .....                 | 101      |
| Properties .....               | 116      |
| Solid Object .....             | 102      |
| Sphere .....                   | 104      |
| Over-Rooftop Propagation ..... | 119, 152 |

**P**

|   |               |
|---|---------------|
| Pan Mode .....  | 75            |
| Parallelized Ray Tracing .....                            | 156           |
| Load Balancing .....                                      | 162           |
| Server Pool .....   | 160           |
| Server Setup .....  | 159           |
| Path Analysis .....                                       | 195           |
| Penetration .....   | 80            |
| Plug-ins  |               |
| Custom Parameters .....                                   | 85            |
| Custom Post-Processing .....                              | 204           |
| Custom Propagation Models .....                           | 163           |
| Custom Surface Plots .....                                | 182, 186      |
| Dialog .....  | 84            |
| Directories .....   | 83            |
| MATLAB Scripts .....                                      | 85            |
| MATLAB User Library Path .....                            | 86            |
| Overview .....  | 82            |
| Programming .....   | 164, 187, 205 |
| Script Editor .....                                       | 85            |
| Point Analysis .....                                      | 192           |
| Polarization .....  | 81            |
| Prediction Algorithms .. <i>See Simulation Algorithms</i> |               |
| Preferences   |               |
| General Preferences                                       |               |
| Client-Server .....                                       | 159           |
| Dynamic Simulation .....                                  | 171           |
| Simulation .....  | 42, 147       |
| Printing  |               |
| ~ from Environment Editor .....                           | 119           |
| Chart Plots .....   | 211           |
| Scripts .....   | 86            |
| Surface Plots .....                                       | 186           |

**R**

|  |          |
|--|----------|
| Ray Launching Approach .....                     | 81       |
| Ray Splitting .....                              | 153      |
| Ray Tracing .. <i>See Ray Launching Approach</i> |          |
| Ray Tracing Algorithms .....                     | 141, 154 |
| Received Power .....                             | 178      |
| Receiver .....                                   | 133      |
| Add a Receiver .....                             | 40       |
| Definition .....                                 | 73, 135  |

|                                      |          |
|--------------------------------------|----------|
| Rx Groups .....                      | 133      |
| Reflection .....                     | 78       |
| Result Layers                        |          |
| Surface Plots .....                  | 44       |
| Results                              |          |
| 2D and 3D Views .....                | 43       |
| Analysis of Simulation Results ..... | 176      |
| Angular Spread .....                 | 181      |
| Best Serving Cell .....              | 179      |
| Channel Impulse Response .....       | 193      |
| Chart Plots .....                    | 209      |
| Coinciding Result Layers .....       | 61, 182  |
| Contour Plots .....                  | 182      |
| Coverage .....                       | 178      |
| Coverage along a Path .....          | 196      |
| Custom Surface Plots .....           | 182, 186 |
| Delay Spread .....                   | 180      |
| Delay Spread along a Path .....      | 197      |
| Direction of Arrival .....           | 194, 203 |
| Getting Started .....                | 43       |
| Histogram .....                      | 185      |
| Path Analysis .....                  | 45, 195  |
| Point Analysis .....                 | 192      |
| Ray Path Analysis .....              | 206      |
| Result Layers .....                  | 178      |
| Retrieve Results .....               | 156      |
| Signal-to-Interference Ratio .....   | 47, 180  |
| Statistical Table Analysis .....     | 214      |
| Table .....                          | 213      |
| Tx Analysis .....                    | 197      |
| Rotation in 3D Views .....           | 75       |
| RPS Server .....                     | 158      |
| Rx Group .....                       | 133      |

**S**

|   |              |
|---|--------------|
| Satellite Mode .....                            | 142          |
| Save Detected Rays .....                        | 153          |
| Server .....                                    | 158          |
| Server Pool .....                               | 160          |
| Shortcuts                                       |              |
| Configuration Dialog .....                      | 51           |
| Signal-to-Interference Ratio .....              | 180          |
| Simulation                                      |              |
| 2.5D Ray Tracing .....                          | 42, 142, 152 |
| 3D Ray Tracing .....                            | 141          |
| Algorithms Overview .....                       | 141          |
| Angular Restrictions .....                      | 151          |
| Bandwidth .....                                 | 151          |
| Best Fitting Ray Tracing Algorithm .....        | 143          |
| Custom Propagation Models .....                 | 163          |
| Dynamic Simulation .....                        | 169          |
| Empirical Models for Indoor Environments .....  | 144          |
| Empirical Models for Outdoor Environments ..... | 143          |
| Hybrid Mode .....                               | 147          |
| Incremental Prediction Mode .....               | 47, 146      |
| Local Processor Units .....                     | 148          |
| Noise Floor .....                               | 152          |
| Optimal Performance Recommendations .....       | 145–47       |
| Over-Rooftop Propagation .....                  | 152          |
| Parallelized Ray Tracing .....                  | 156          |
| Ray Launching Approach .....                    | 81           |
| Ray Splitting .....                             | 153          |

|   |              |
|---|--------------|
| Ray Tracing Algorithms.....                       | 154          |
| Restriction of Propagation Phenomena .....        | 152          |
| Retrieve Results .....                            | 156          |
| Satellite Mode.....                               | 142          |
| Save Detected Rays .....                          | 153          |
| Setup .....                                       | 42, 147, 157 |
| Single Reflection at Non-vertical Obstacles ..... | 152          |
| Starting RPS Servers.....                         | 158          |
| Statistics.....                                   | 175          |
| Step Size .....                                   | 151          |
| Summary of Algorithms.....                        | 17           |
| Single Reflection at Non-vertical Obstacles ....  | 152          |
| Statistics .....                                  | 175          |
| Stencil Library .....                             | 105, 262     |
| Support   |              |
| Electronic Support.....                           | 251          |
| Support Center.....                               | 251          |
| Telephone Support.....                            | 251          |
| Surface Plots.....                                | 59, 177      |
| LOS Preview Plot .....                            | 131          |
| System Requirements.....                          | 20           |
| System Templates.....                             | 151          |
| <b>T</b>  |              |
| Table.....  | 213          |
| Copy/Fill.....                                    | 130          |
| Editing .....                                     | 96, 130      |
| Filter Functions .....                            | 130          |
| Third Party Software.....                         | 252          |
| Toolbars  |              |
| Component Toolbar .....                           | 73, 257      |
| Paint Toolbar .....                               | 258          |
| Standard Toolbar .....                            | 257          |
| Transmitter .....                                 | 124          |
| Add a Transmitter .....                           | 39           |
| Definition.....                                   | 72, 126      |
| States .....                                      | 132, 147     |
| Tx Analysis                                       |              |
| Antenna Azimuth Angle Analysis .....              | 200          |
| Antenna Tilt Analysis .....                       | 198          |
| Direction of Arrival.....                         | 203          |
| Transmit Power Analysis.....                      | 202          |
| <b>U</b>  |              |
| Undo Buffer.....                                  | 53           |
| <b>W</b>  |              |
| Workflow Model .....                              | 16           |
| Simulation.....                                   | 144          |
| Workspace .....                                   | 61           |
| Color Palette .....                               | 66           |
| Display Settings .....                            | 63           |
| Layer Settings .....                              | 63           |
| Legend Presets.....                               | 69           |
| <b>Z</b>  |              |
| Zoom Functions.....                               | 74           |