Definiens

Developer 7

User Guide

Definiens AG

www.definiens.com

Imprint and Version

Document Version 7.0.1.872

Published by

Definiens AG
Trappentreustr. 1
D-80339 München
Germany

Phone +49-89-231180-0
Fax +49-89-231180-90

E-mail **info@definiens.com**
Web **http://www.definiens.com**

Dear User,

Thank you for using **Definiens** software. We appreciate being of service to you with image intelligence solutions.

At Definiens we constantly strive to improve our products. We therefore appreciate all comments and suggestions for improvements concerning our software, training, and documentation.

Feel free to contact us via web form on the Definiens support website **http://www.definiens.com/support/index.htm**.

Thank you.

Legal Notes

Table of Contents

1 Overview

Definiens Developer is a powerful Integrated Development Environment (IDE) for rapid image analysis solution development.

Developers have unlimited access to the full functionality of the Definiens Cognition Network Language along with development and workflow tools to aid in the rapid development of new image analysis solutions. The developed solutions, called ruleware can be saved for automated image analysis both in stand-alone and distributed environments using **Definiens eCognition Server**.

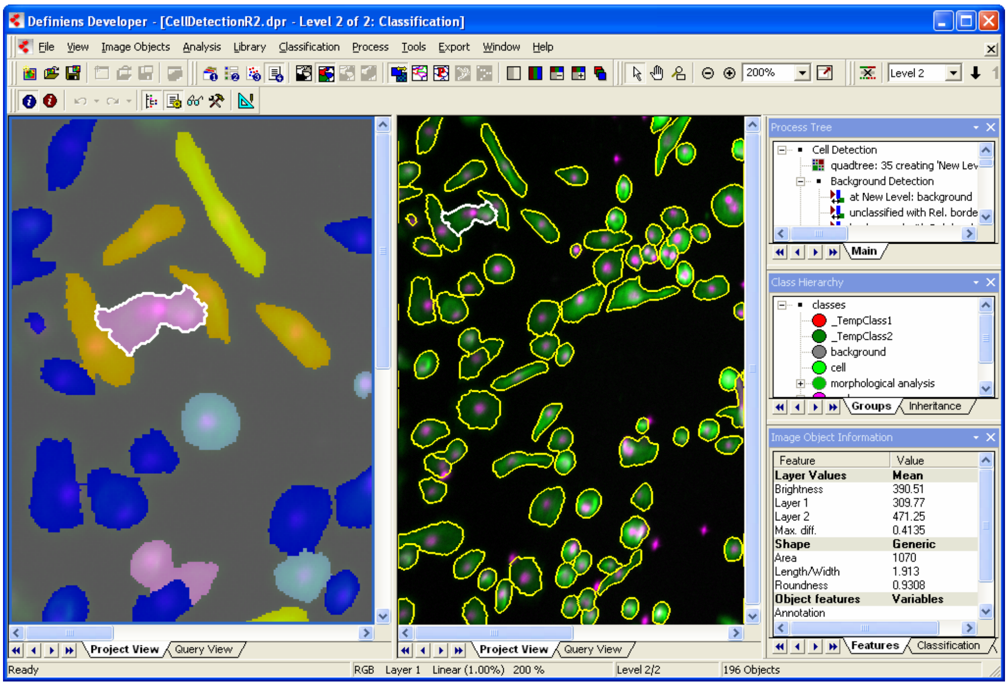


Figure 1: Main window (Image data courtesy of Cenix BioScience GmbH).

Definiens Developer highlights:

- Graphical development environment allows rapid development of re-usable components, rules and modules. The development of image analysis solution is carried out interactively. The impact on the results of each development step can be immediately visualized and traced in the displayed image.
- The Definiens Cognition Network Language delivers powerful image intelligence logic capabilities for querying and analyzing segmentation, context, shape and any other variable.
- Control panel for running parallel analysis on the server installation.

- Industry-specific functionality can be added using the appropriate **Definiens Applications**.
- Resource sharing facility to enable expertise exchange between groups and departments lowering the total cost of ownership.
- Hierarchical object relationship support allows extraction of information at an unprecedented level of accuracy and complexity.
- Plug-in technology components to support new proprietary and non-proprietary image formats.
- Comprehensive algorithm and feature toolbox with open API to support plug in of custom features and algorithms.
- Process profiler to support development of computationally efficient applications.

Definiens Developer also contains all of the functionality provided by **Definiens Architect**, **Definiens Analyst**, and **Definiens Viewer**.

For an overview of **Definiens** product components see the separate **Installation Guide**.

→ Installation Guide

1.1 What's New in Version 7

Improvements for the Rule Set Development Workflow

- Three new types of variables for classes, features and image object levels facilitate development of parameters of actions as used in **Definiens Architect**.

→ About Variables on page 257

- Undo process editing operations allows you to go back to a previous state in process editing.

→ Undo Process Editing Operations on page 139

- Find and Replace within rule sets for classes, features, image object levels, variables, image layers, and thematic layers.

→ Find and Replace Rule Set Items on page 147

- Organize and save process snippets to a library for reuse in other rule sets.

→ Save and Reuse Process Snippets on page 148

- Generate a rule set documentation from comments to processes.

→ Document a Process Sequence on page 149

- The handling of customized algorithms has been improved concerning the dependencies of the rule set items. Rule set items and their dependencies are visualized in a newly structured list. Scope dependencies of rule set items are applied automatically according to applicable consistency rules.

→ Create a Customized Algorithm on page 278
→ About the Scope of Rule Set Items of a Customized Algorithm on page 276

- A new **contrast split** algorithm allows segmentation of an image or an image object into dark and bright regions.

→ Reference Book

- Some settings that control the rules of behavior of a rule set have been moved from the **Options** dialog box to the new **set rule set options** algorithm. This includes vectorization thresholds and distance calculations.

→ Reference Book
→ Options on page 452

Improved Viewing Options

- Split the project view to use it with different viewing modes when investigating projects.

→ Use Different Viewing Modes on page 116

- A new transparency slider allows you to adjust the transparency of the classification overlay.

→ Adjust Classification Transparency on page 110

- A new simultaneous magnification of the image facilitates detail investigation.

→ Magnify the Image in a Separate Window on page 106

- Display a new scale bar in the project view.

→ View a Scale Bar on page 107

- Display thumbnail view of images.

→ Change the View in the Workspace on page 80

- Use the new heat map with integrated thumbnails t to compare workspace statistics or tile statistics.

Available only for Definiens eCognition **Life** Server

→ View Thumbnails in the Heat Map Window on page 380

Workspace Automation

- Merge results back to the main scene.

→ Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI) on page 299

- Extract image objects as scene subsets. Other image objects within the bounding box are hidden by a no-data-mask.

→ Copy of a Scene Subset on page 303

- New algorithm for collecting statistics of tiles or subset scenes.

→ Collect Statistical Results of Subscenes on page 305

- New algorithm for deleting scenes.

→ Reference Book

- Submit a randomly selected portion of tiles for processing.

Preprocessing Algorithms

- A set of preprocessing algorithms allow apply filters to image layers at the pixel level. Such preprocessed layers are typically used within the segmentation process or for the classification process and can be used to improved the quality of the information extraction.

→ Reference Book

Data Import and Export

- The usability of the customized importer has been improved and enhanced metadata support enables generation of all metadata as features and supports multiple metadata items, simplifying development of solutions such as change detection.

→ Customize the Import of Scenes into a Workspace on page 63

- Improved shape file export including better string handling and class name and class color features.

Data Processing

- A time-out function allows you to define automatic cancellation of image analysis after a period of time.

→ Start Automated Image Analysis on page 363

Software Development Kit

- The software development kit (SDK) of Definiens includes three application programming interfaces (API). Each API comes with a user guide document (.pdf) and a reference help file (.chm) and samples. This SDK is provided with Definiens Developer, there is no additional licensing requirement for site wide usage.

→ SDK User Guides and References on page 471

Definiens Data Management Compatible

- Compatible with the **Definiens Data Management** option. Allowing cost effective and secure management of analysis results data on an enterprise scale.

Available only for Definiens **Data Management**

User Documents

- New recommendations about how to start learning about the **Definiens Developer**.

→ Get a Quick Start on page 14

- A guided tour **Mapping Impervious Surface** presenting a geographic example of image analysis.It includes sample data to follow in practice. It focuses on the basic steps involved in developing a rule set using **Definiens Developer** and submitting a set of images for batch processing.

→ Tutorials and Guided Tours on page 470

- New introduction into object-oriented image analysis.

→ Introduction into Object-Oriented Image Analysis on page 124

- A new chapter focuses on the extended workflow of rule set development including the strategic approach. This chapter enables you to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ About Development Strategies on page 324

- Reworked chapter about segmentation basics.

→ Create Image Objects by Segmentation on page 157

- Extended section about focusing processes by using the domain concept.

→ Focus Processes by Using the Domain Concept on page 197

1.2 Get a Quick Start

This User Guide addresses you as a user of **Definiens Developer**, whose goal is to develop rule sets and ruleware that process automated image analysis. For a successful quick start in **Definiens Developer** you need some introductory activities to become familiar with the various tools and capabilities.

Some previous knowledge is helpful for easy learning.

→ Recommended Skills on page 122

Depending on the type of user you are, we recommend different quick starts for working-in with **Definiens Developer**. Select your user type and follow the recommended quick start path:

- **Update User**: You are experienced in using a recent version of **Definiens Developer**. If so, you may consult the **What's New** section referring you directly to

→ What's New in Version 7 on page 12

instructions about new functions.

- **Upgrade User**: You are experienced in using other Definiens products like **Architect** or **Analyst**.

→ Upgrade User's Quick Start on page 15

- **Advanced User**: You are an image analysis expert or software developer or have comparable advanced software knowledge.

→ Advanced User's Quick Start on page 15

- **Novice User**: You are not yet experienced with **Definiens** software or other software development tools.

→ Novice User's Quick Start on page 16

Definiens offers hands-on training designed specifically to hone the skills of users.

→ Definiens Training on page 124

1.2.1 Upgrade User's Quick Start

As an upgrade user of other Definiens products like **Architect** or **Analyst**, you can focus your quick start on the development environment and how to develop rule sets.

1. Upgrade users may start by reading about some unknown **Basic Concepts** related to development functions.

→ Basic Concepts on page 19

2. To get a brief introduction to the development workflow of **Definiens** image analysis we recommend the **Tutorial Basic Knowledge 1**. Using sample data, you perform a very simple process sequence.

→ Tutorials and Guided Tours on page 470

   Go to the data folder **TutorialBasicKnowledge1**, which is installed in the **Definiens** installation directory. The default path is **C:\Program Files\Definiens Developer** *Version number***Examples \TutorialBasicKnowledge1**.

3. If you are using **<podname> Developer** for geographic image analysis, we recommend that you work through the guided tour **Mapping Impervious Surface** presenting a geographic example of image analysis. It includes sample data to follow in practice. This guided tour focuses on the basic steps involved in developing a rule set using **Definiens Developer** and submitting a set of images for batch processing.

   Guided tours are installed separately. Consult the **Installation Guide** for details.

→ Installation Guide on page 469

4. After reviewing starting instructions for creating basic rule sets, continue by learning how to benefit from all tools of the development environment. **The Develop Efficient Rule Sets** chapter provides detailed handing instruction to build efficient and reusable rule sets and other ruleware.

→ Start Developing Rule Sets on page 121
→ Develop Efficient Rule Sets on page 192

   Simultaneously, you should consult the **About Strategies** chapter focusing on composing rule sets based on the **Definiens** Cognition Network Language (CNL). Learn how to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ About Development Strategies on page 324

1.2.2 Advanced User's Quick Start

Based on your experiences as an image analysis expert or software developer or equipped with comparable advanced software knowledge, you can focus on the specifics of the **Definiens** development environment and how to debelop **Definiens** rule sets.

1. Advanced users may start by reading the **Basic Concepts** chapter introducing elementary terms.

→ Basic Concepts on page 19

2. The **Workflow** chapter presents an overview of the basic operational sequence. It refers you to different relevant chapters of the basic operational sequence.

→ Workflow on page 32

3. To get a brief introduction to the development workflow of **Definiens** image analysis we recommend the **Tutorial Basic Knowledge 1**. Using sample data, you perform a very simple process sequence.

   Go to the data folder **TutorialBasicKnowledge1**, which is installed in the **Definiens** installation directory. The default path is **C:\Program Files\Definiens Developer** *Version number***\Examples \TutorialBasicKnowledge1**.

→ Tutorials and Guided Tours on page 470

4. If you are using **<podname> Developer** for geographic image analysis, we recommend that you work through the guided tour **Mapping Impervious Surface** presenting a geographic example of image analysis. It includes sample data to follow in practice. This guided tour focuses on the basic steps involved in developing a rule set using **Definiens Developer** and submitting a set of images for batch processing.

   Guided tours are installed separately. Consult the **Installation Guide** for details.

→ Installation Guide on page 469

5. After reviewing starting instructions for creating basic rule sets, continue by learning how to benefit from all tools of the development environment. **The Develop Efficient Rule Sets** chapter provides detailed handing instruction to build efficient and reusable rule sets and other ruleware.

→ Start Developing Rule Sets on page 121

→ Develop Efficient Rule Sets on page 192

   Simultaneously, you should consult the **About Strategies** chapter focusing on composing rule sets based on the **Definiens** Cognition Network Language (CNL). Learn how to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ About Development Strategies on page 324

1.2.3 Novice User's Quick Start

For users who are not yet experienced with **Definiens** software or other software development tools, we provide multiple learning documents.

However, some previous knowledge is helpful for easy learning.

→ Recommended Skills on page 122

1. Start learning about **Definiens Developer** by using the **Tutorial Basic Knowledge 1**. Using sample data, you perform a very simple process sequence. Thus, you get a basic idea about the common elements of the **Definiens Developer** user interface and the basic development workflow of **Definiens** image analysis.

   Go to the data folder **TutorialBasicKnowledge1**, which is installed in the **Definiens Developer** installation directory. The default path is **C:\Program Files\Definiens Developer** *Version number***\Examples \TutorialBasicKnowledge1**.

→ Tutorials and Guided Tours on page 470

2. If you are using **<podname> Developer** for geographic image analysis, we recommend that you work through the guided tour **Mapping Impervious Surface** presenting a geographic example of image analysis. It includes sample data to follow in practice. This guided tour focuses on the basic steps involved in developing a rule set using **Definiens Developer** and submitting a set of images for batch processing.

Guided tours are installed separately. Consult the **Installation Guide** for details.

→ Installation Guide on page 469

3. In the User Guide, the current document, the **Workflow** chapter presents an overview of the basic operational sequence.

→ Workflow on page 32

4. Work through the instructional chapters describing how to use the main functions to fulfill main tasks of image analysis with **Definiens Developer**. These instructions are sequenced according a basic workflow.

→ Start Definiens Software on page 37
→ Load and Manage Data on page 43
→ View Data in Projects on page 93
→ Start Developing Rule Sets on page 121
→ Process Data on page 361
→ Examine Results on page 379
→ Export Data on page 401

5. After reviewing starting instructions for creating basic rule sets, continue by learning how to benefit from all tools of the development environment. **The Develop Efficient Rule Sets** chapter provides detailed handing instruction to build efficient and reusable rule sets and other ruleware.

   Simultaneously, you should consult the **About Strategies** chapter focusing on composing rule sets based on the **Definiens** Cognition Network Language (CNL). Learn how to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ Start Developing Rule Sets on page 121
→ Develop Efficient Rule Sets on page 192
→ About Development Strategies on page 324

1.3 About the Software Development Kit (SDK)

Definiens Developer includes a software development kit (SDK) enabling you to modify **Definiens** software without knowledge of the whole development code base.

The SDK consists of three components each representing an application programming interfaces (API):

→ SDK User Guides and References on page 471

- **Data IO API** for developing drivers, or plug-ins for data connectivity and integration.
- **Engine API** for expanding the capabilities of the **Definiens** Analysis Engine Software by plug-ins providing additional algorithms or features.
- **Automation API** for automating **Definiens** software and orchestrating the different processes. Local automation allows you to embed the eCognition AnalysisEngine into other software. Remote Processing enables you to provide functionality that can be, for instance, used in a script, in order to automate **Definiens eCognition Server**.

If the SDK is not yet installed on your machine, rerun the **Definiens Developer** installation. During the installation sequence on the **Choose Components** dialog box, you have to select the **SDK** check box only.

→ Installation Guide
→ Installation Guide on page 469

2 Basic Concepts

Contents in This Chapter

This chapter introduces elementary terms of image analysis with **Definiens Developer**.

2.1 Image Understanding

When human beings make use of their eyes, they are performing a complex mental procedure. This procedure is called image understanding. In this section, you can investigate an approach to visual cognition and learn about how **Definiens** software reproduces this ability.

When you survey a region with your eyes, you bring certain areas in your surroundings into focus. You probably register that a certain area has a particular size, form, and color. Thus, in your vision, it becomes an object. For example, you see a red round area and classify it as a red, round object.

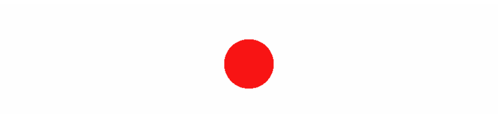


Figure 2: A red, round object.

Figure 3: A red, round object.

Underneath you can see another object, which you already know to be a human being. Immediately you combine both figures and relate them to each other.

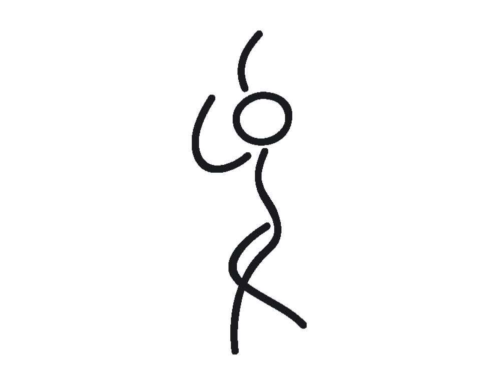


Figure 4: A human being.

Both objects may be located in front of a third large blue object.

Figure 5: A blue object.

The three objects are connected by defined relationships (for example: **underneath**, or **in front of**). The features of the objects and their relationships allow you to recognize a child playing a ball in the sky. This cognition process compares your view of the objects and their relationships with the existing knowledge of your memory.

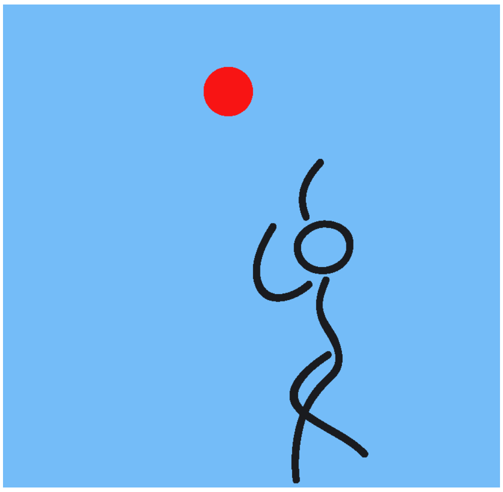


Figure 6: What do you recognize?

Similar to human vision, the **Definiens** concept of image understanding is based on a correct segmentation of the visual image content of interest against other visual image content. Segmentation is performed by splitting the image into zoned partial areas of differing characteristics. The segments are called **image objects**.

These image objects can be classified according to particular criteria. The resulting classes are related and arranged in semantic groups, which represent knowledge about the image. This structure of knowledge representation allows automated image analysis.

To set up such object-oriented representations of knowledge, **Definiens Developer** offers an iterative workflow. Put simply, it consists of the basic iterating procedures of **segmentation** (or reshaping) and **classification**. Other procedures help in finding image objects and in organizing image analysis.

2.2 Image and Image Layer

An image is a set of raster image data. An image consists of at least one image layer based on pixels. Each image layer represents a type of information, based on an array of pixels. The most common layers are the Red, Green and Blue (RGB) image layers, but there are other image layer types, such as image layers with biomarkers used in life sciences or with NIR data used in remote sensing.

An image is stored as **.tif**, **.bmp**, **.frm**, or other raster file format.

Within **Definiens Developer** images are represented by scenes.

2.3 Scene

Definiens Developer loads image data as scenes, which means that each scene usually represents one image.

A scene consists of one or more image layers or channels of one image file. When working with a combination of different data, a scene usually contains several image files with multiple image layers and optionally thematic layers.

A scene can include additional information related to the image content, such as metadata, geocoding, or geo information.

Depending on the image reader or camera, a scene combines multiple views of the same piece of reality, each of them in a separate layer. To put it simply, a scene can include several images of the same thing, each of them providing different information.

Each scene representing one set of data used in **Definiens** software is managed in a project. Thus, scenes are the combined input image data for both projects and **Definiens** workspaces.

2.4 Project

In **Definiens** software, a project manages a scene.

A project is a wrapper for all information related to a particular scene which is the input image data. It stores references to at least one image layer and related result information from image analysis expressed by classified image objects. Furthermore, a project contains metadata like layer aliases and unit information. Optionally, a project can enclose thematic layers.

During creation of a project, a scene is referenced into the project. Image analysis extracts result information from a scene and adds it to the project. This information is expressed in classified image objects. When viewing analyzed projects you can investigate both the input scene and the classification of image objects representing a result.

A project is saved as a **.dpr** project file.

2.5 Workspace

A workspace is a container for projects, saved as **.dpj** file.

A workspace file contains image data references, projects, exported result values and references to the used ruleware. Furthermore, it comprises the import and export templates, result states, and metadata.

In the **Workspace** window, you administer the workspace files. Here you manage all relevant data of your image analysis tasks.

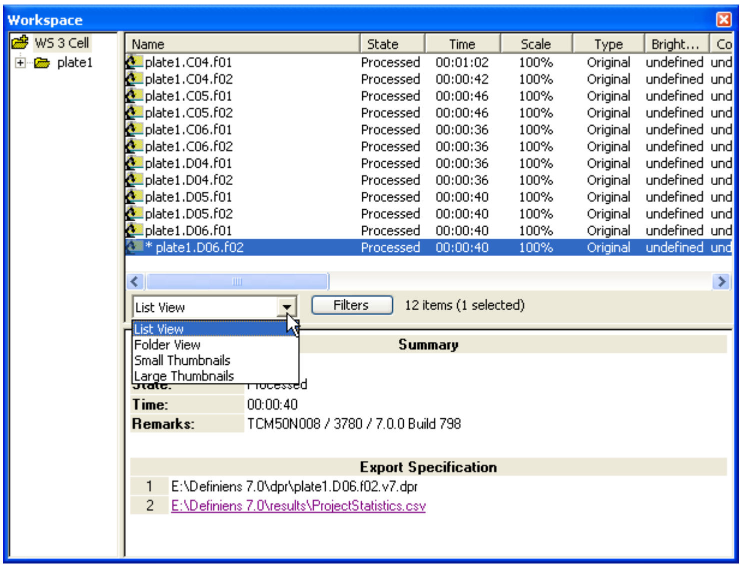


Figure 7: Workspace window with Summary and Export Specification and drop-down view menu.

2.6 Image Object

During image analysis, a scene—representing an image—is split into image objects. An image object is group of connected pixels in a scene. Each image object represents a definite region in an image.

Image objects can provide information about this definite image region. Every image object is linked to its neighbors. Together the image objects perform a network, which enables access to the context of each image object.

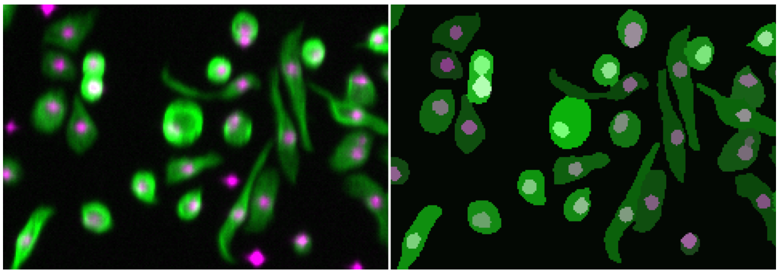


Figure 8: Image Object detection (right) in a cell image (right). (Image data courtesy of Cenix BioScience GmbH.)

2.7 Image Object Level

A scene, representing an image, is segmented into image objects during the process of image analysis. Image objects are organized into image object levels. An image object level serves as an internal working area for the image analysis.

→ Scene on page 22

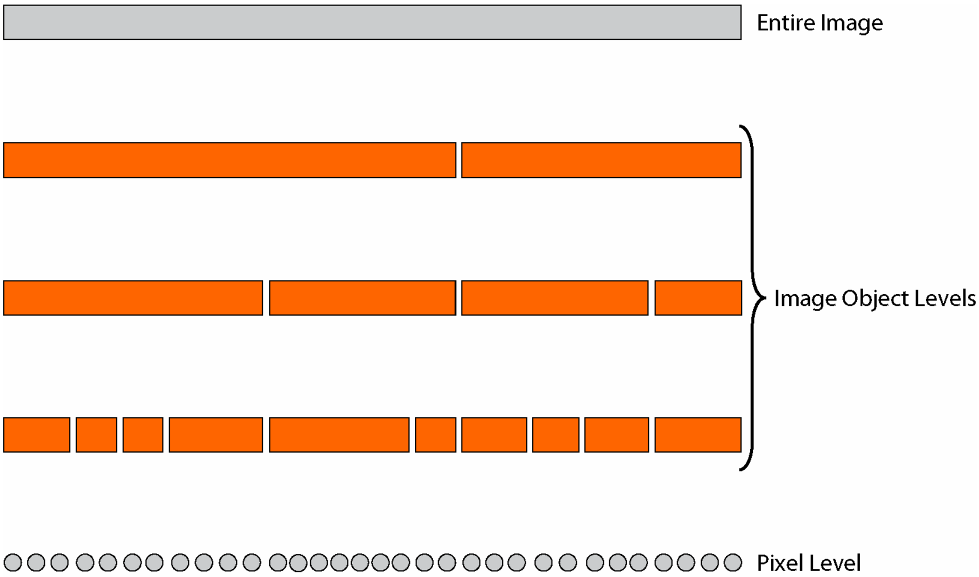


Figure 9: Each image object level consist of a layer of image objects.

During image analysis multiple image object levels can be created and layered above the basic pixel level. Two or more image object levels build an image object hierarchy.

→ Image Object Hierarchy on page 26

> **Note**
>
> Keep in mind the important difference between image object levels and image layers. Image layers represent the data already existing in the image when it is first imported. In contrast, image object levels store image objects, which represent the data. Thus, they serve as internal working areas.

An image object level can be created by segmentation from the underlying pixel level or from an existing image object level. In addition, you can create an image object level by duplicating an existing image object level.

Image objects are stored in image object levels. Image object related operations like classification, reshaping and information extraction are done within image object levels. Thus, image object levels serve as internal working areas of the image analysis.

Every image object is linked to its neighbors. Together the image objects perform a cognition network, which enables access to the context of each image object.

2.8 Feature

In **Definiens** software, a feature is an attribute that represents certain information concerning objects of interest, for example measurements, attached data or values.

There are two major types of features:

- Image **Object features** are related to image objects. Object features describe spectral, form, hierarchical, or other properties of an image object, for example its **Area**.
- **Global features** are not related to an individual image object, for example the **Number of classified image objects** of a certain class.

Image Object Features

Since regions in the image provide much more information than single pixels, there are many different image object features for measuring color, shape, and texture of the associated regions. Even more information may be extracted by taking the network structure and the classification of the image objects into account. Important examples of this type of features are the **Rel. border to** neighboring objects of a given class and **Number of subobjects** of a given class.

Global Features

Global features describe the current network situation in general. Examples are the **Mean** value of a given image layer or the **Number of levels** in the image object hierarchy. Global features may also represent metadata as an additional part of the input data. For example the type of tissue in a toxic screen might be expressed via metadata and thus incorporated into the analysis.

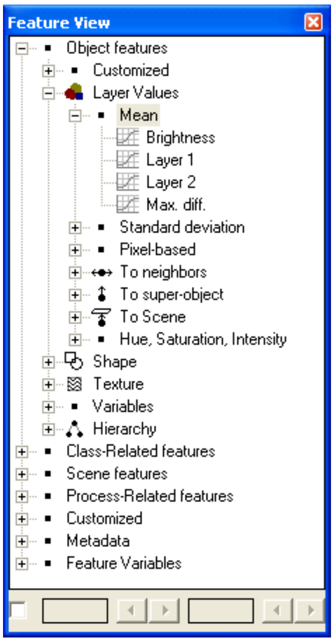


Figure 10: Features in the Feature View window.

2.9 Class and Classification

A class is a category of image objects. It can both be used to simply label image objects or to describe its semantic meaning. Classification is a procedure that associates image objects with an appropriate class labeled by a name and a color.

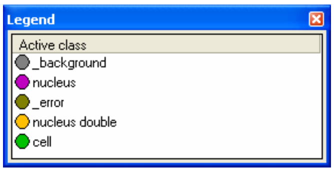


Figure 11: Legend window listing classes.

Information contained in image objects is used as a filter for classification. Based on this, image objects can be analyzed according defined criteria and assigned to classes that best meet these criteria.

The classes can be grouped in a hierarchical manner, allowing the passing down of their defining class descriptions to child classes using the inheritance hierarchy. Classes form a structured network, called class hierarchy.

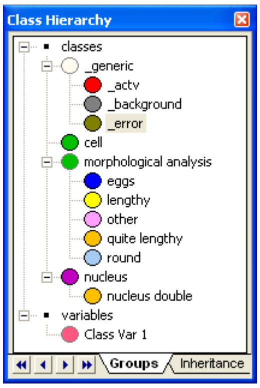


Figure 12: Sample Class Hierarchy window.

Through the process of classification, each image object is assigned to a certain—or no—class and thus connected with the class hierarchy. The result of the classification is a network of classified image objects with concrete attributes, concrete relations to each other and concrete relations to the classes in the class hierarchy.

2.10 Image Object Hierarchy

During image analysis, multiple image object levels can be created and layered above the basic pixel level. Two or more image object levels build the image object hierarchy.

Put simply: The image object hierarchy serves as a storage rack for all image objects levels which represent the different shelves storing the image objects. Thus the image object hierarchy provides the working environment for the extraction of image information.

The entirety of image objects is organized into a hierarchical network of image objects. Such a network is called image object hierarchy. It consists of one or more image object levels, from fine resolution on the lowest image object level to coarse resolution on the highest image object level.

Image objects within an image object level are linked horizontally. Similarly, image objects are linked vertically in the image object hierarchy. The image objects are networked in a manner that each image object knows its context, that are its neighbors, its superobject on a higher image object level and its subobjects on a lower image object level.

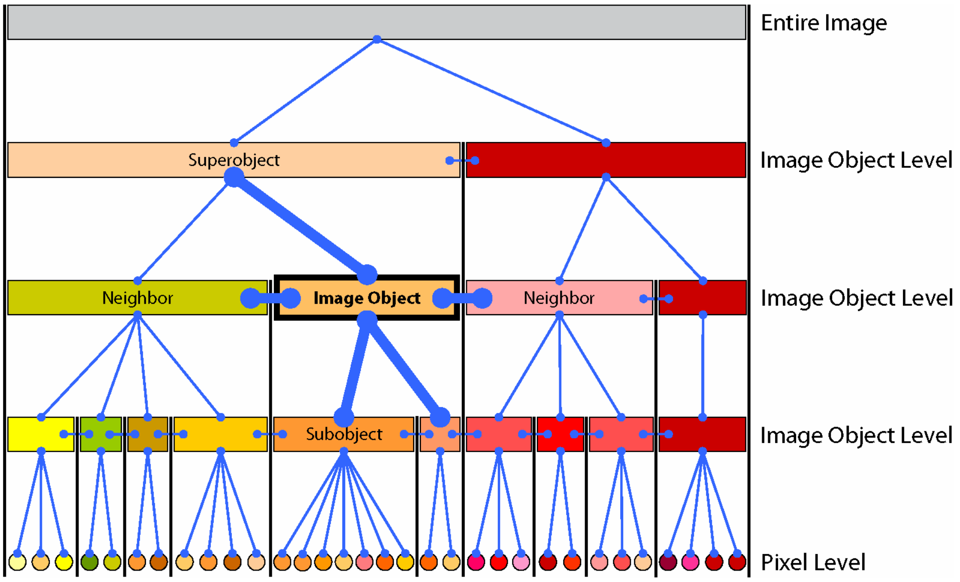


Figure 13: Within the image object hierarchy, each image object is linked to its neighbors, its superobject, and its subobjects.

To assure definite relations between image object levels, no image object may have more than one superobject but it can have multiple subobjects. The border of a superobject is consistent with the border of its subobjects.

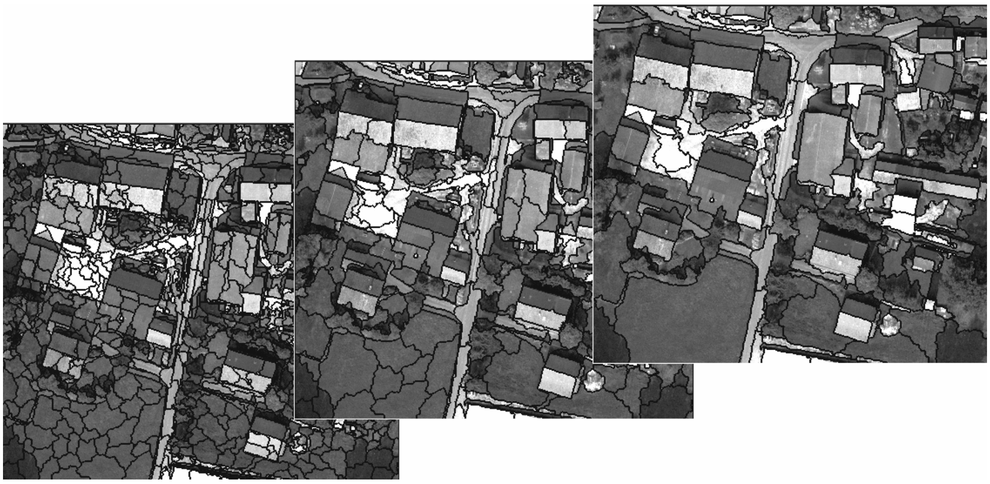


Figure 14: Three image object levels of a sample image.

2.11 Definiens Application

A **Definiens** application can extend each of **Definiens Enterprise Image Intelligence** platform products providing industry- or user-specific ruleware and functions.

Definiens applications enlarge the capabilities of both the **Definiens Enterprise Image Intelligence (EII)** Clients (**Definiens Developer, Architect, Analyst, Viewer)** and the processing environment **Definiens eCognition Server**.

Definiens applications enable users of clients like **Definiens Architect** to create ready-to-use solutions for their specific image analysis problem. Started with one of the **Definiens Image Intelligence Clients**, an application completes the client functionalities by using particular ruleware, controls, and workflows needed for industry- or user-specific tasks.

Example: If you start a **Definiens** client like **Developer**, **Architect**, **Analyst**, or **Viewer** together with, for example, ABC application, you can use specific ABC functionalities for your particular ABC tasks. If you start a client with another XYZ application, you benefit from other specific XYZ functionalities needed for your particular XYZ tasks.

Definiens offers a range of applications. They are optional and licensed through application-specific licenses.

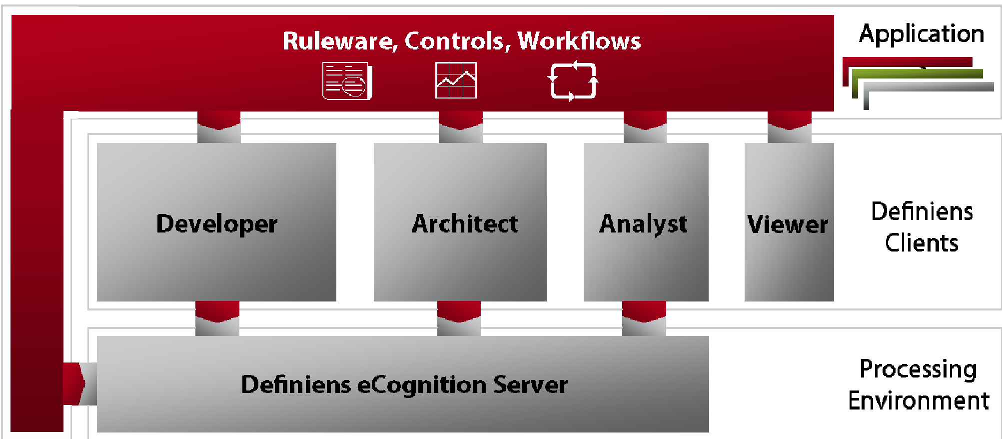


Figure 15: Definiens applications extend the Enterprise Image Intelligence clients and processing environment.

2.12 Solution

A solution is designed by **Definiens Architect** users as a ready-to-use image analysis solving a specific image analysis problem.

A solution provides an image analysis rule set configured for a specific type of image data. A solution is assembled from predefined building blocks called actions.

2.13 Action

An action represents a predefined building block of an image analysis solution. Configured actions can perform different tasks like object detection, classification or export of results to file. Actions are sequenced and together they represent a ready-to-use solution accomplishing the image analysis task.

A configured action consists of a set of processes with defined parameters. Standard action definitions, which are just unconfigured actions, are provided in action libraries.

Special task actions can be designed according specific needs using **Definiens Developer**.

2.14 Ruleware and Rule Sets

Definiens Developerprovides a development environment for creating ruleware based on the **Definiens** Cognition Network Language (CNL). **Ruleware** is a piece of software applicable to defined image analysis tasks, such as a rule set, an action, or a solution.

Rule sets represent the code of ruleware assembling a set of functions. Based on rule sets, you can create **actions**, which are packaged modules of rule sets with a user interface for configuration of parameters. You can assemble actions in action libraries and provide them to non-developer users for easy creation of solutions for defined image analysis tasks. Non-developer users configure and assemble actions and save them as a **solution**, ready to use for processing on image data.

2.15 Process

Definiens Developer provides an artificial language for developing advanced image analysis algorithms. These algorithms use the principles of object oriented image analysis and local adaptive processing. This is achieved by processes.

A single process is the elementary unit of a rule set providing a solution to a specific image analysis task. Processes are the main working tools for developing rule sets.

> **Note**
>
> In **Definiens Developer**, the term **Process** is used for both a single process and a process sequence.

The main functional parts of a single process are the algorithm and the image object domain. A single process allows the application of a specific algorithm to a specific region of interest in the image. All conditions for classification as well as region of interest selection may incorporate semantic information.

Processes may have an arbitrary number of child processes. The resulting process hierarchy defines the structure and flow control of the image analysis. Arranging processes containing different types of algorithms allows the user to build a sequential image analysis routine.

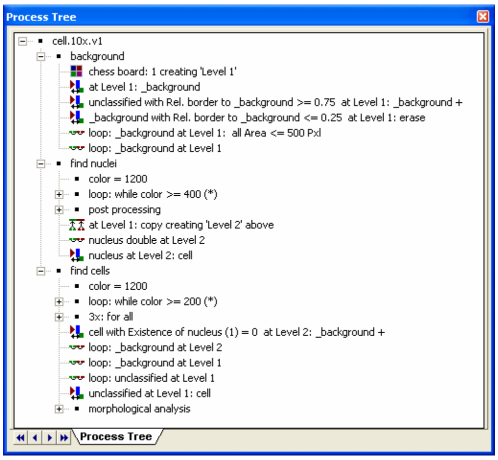


Figure 16: Process Tree window.

2.15.1 Algorithm

The algorithm defines the operation the process will perform. This can be generating image objects, merging or splitting image objects, classifying objects, and so on. The two main functions of algorithms are generating or modifying image objects and classifying image objects. In addition to these, a set of other algorithms help to define all necessary operations to set up an image analysis routine. The following functional categories of algorithms exist:

- Process related operation
- Segmentation algorithms
- Basic Classification algorithms
- Advanced Classification algorithms
- Variables operation algorithms
- Reshaping algorithms
- Level operation algorithms
- Interactive operations algorithms
- Sample operation algorithms
- Image layer operation algorithms
- Thematic layer operation algorithms
- Export algorithms
- Workspace automation algorithms

2.15.2 Image Object Domain

The image object domain describes the region of interest where the algorithm of the process will be executed in the image object hierarchy.

The image object domain is defined by a structural description of the corresponding subset. Examples for image object domains are the entire image, an image object level or all image objects of a given class.

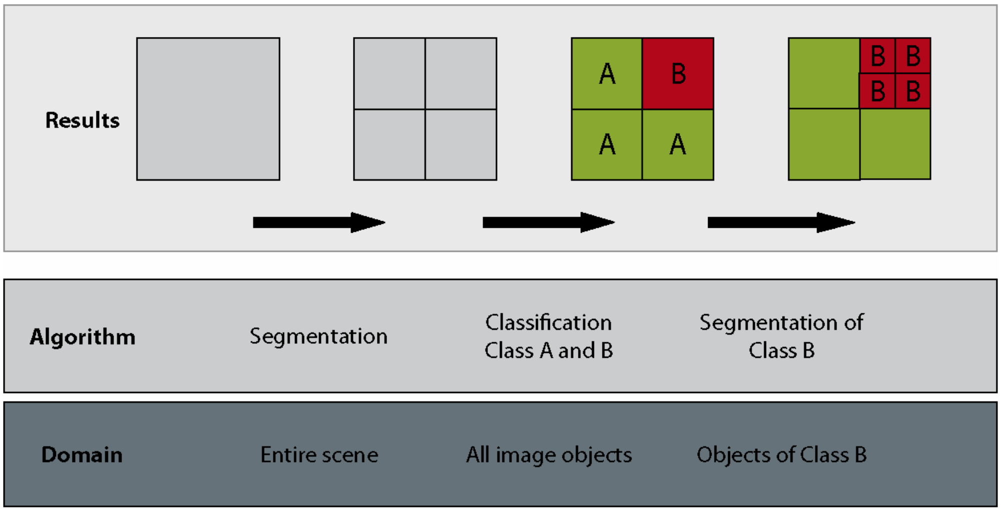


Figure 17: Workflow of a process sequence.

By applying the usual set operators to the basic image object domains, many different image object domains can be generated. The process will then loop over the set of image objects in the image object domain and apply the algorithm to every single image object.

Therefore, image objects domains may be defined relative to the current image object of the parent process, for example the subobjects or the neighboring image objects of the parent process object (PPO).

3 Workflow

Contents in This Chapter

This chapter presents a overview of the workflow and sequences the basic operational steps.

3.1 Start Definiens Developer

Start a **Definiens** client like **Developer**, **Architect**, **Analyst**, or **Viewer** together with an application to use specific functionalities for your specific tasks.

→ Start Definiens Software on page 37

Industry- or user-specific applications enable users of clients like **Definiens Architect** to create ready-to-use solutions for their specific image analysis problem. Starting with other clients, an application completes the client functionalities by providing particular tools and controls needed for industry- or user-specific tasks.

Definiens offers a range of applications.

3.2 Load and Manage Data

Images can be analyzed in a single project or in a workspace which can contains several projects.

→ Load and Manage Data on page 43

A single project is used to develop your image analysis solutions. A project refers to image data (called scenes). Image analysis extracts information from a scene and adds it to the project. Multiple projects can be put together and managed in a workspace.

A workspace is a file that bundles projects so that you can perform both manual and automated image analysis. In the **Workspace** window, you manage your image analysis tasks and administrate all relevant data. Thus, you control the workflow of automated image analysis.

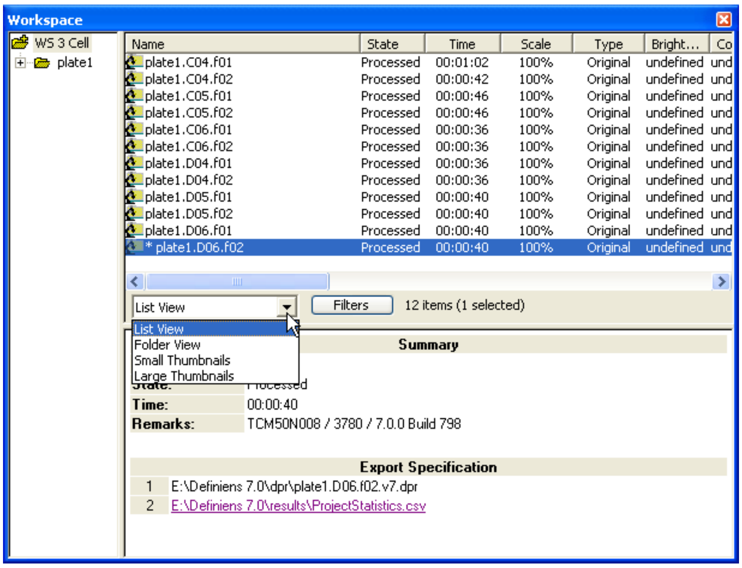


Figure 18: Workspace window with Summary and Export Specification and drop-down view menu.

To run an automated image analysis you first have to import scenes to add image data to the workspace. In a workspace, each scene is administrated as a project.

3.3 View Data in Projects

→ View Data in Projects on page 93

You have several options for viewing and investigating your **Definiens** projects visually. Some of the visualization tools are dependent on the extent of analysis that has been performed.

→ View a Project on page 84
→ Use Workflow Views on page 40

1. You can explore individual image layers right from the start, after loading scenes, and throughout the image analysis, including the investigation of results. You can define the color composition for the display of image layers and set equalizing options.

→ Change Display of Image Layers on page 95

2. You can use different zoom and pan functions at any stage of image analysis.

→ Zoom Within Projects on page 101

3. After a basic segmentation that creates the first image objects, you can view their borders, called outlines.

→ View Outlines of Image Objects on page 108

4. Next, you classify image objects using various methods. You will commonly review the classification results extensively.

→ Change Display of Image Objects on page 109

   A specific visualization of features (image object attributes) enables you to preview the relevance of comprised information.

→ Feature View on page 111

5. For refinement of your image analysis, you can navigate visually through the different (if there is more than one) levels of the image object hierarchy. This enables you to to maintain information about the different image object levels and to perform more classifications.

→ Navigate Within Projects on page 114

6. Additionally, you can use several views of a project at the same time.

→ Use Different Viewing Modes on page 116

If you work with polygons you can display polygons and skeletons.

→ View Polygons on page 112
→ View Skeletons on page 114

Figure 19: Project View: views of different image layers of a new project.

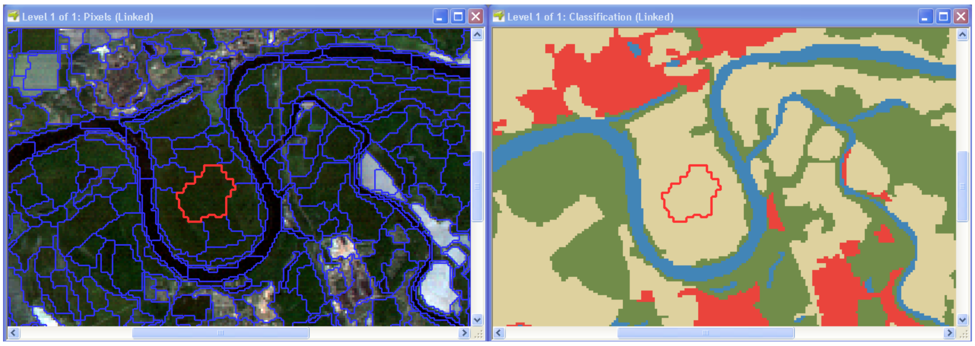


Figure 20: Project View: different views of a classified project.
Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.

3.4 Develop Rule Sets

The process of detecting objects in an image consists of a sequence of segmentation and classification procedures. The sequence is defined in the rule set, which represents the program code. It consists of a list of processes, which represent commands, organized in a tree-like structure.

→ About Image Analysis with Image Objects on page 124

The typical workflow of the rule set development consists of three iterative activities:

- Use one the different ways of retrieving information about image objects.

→ Get Information on Image Objects on page 126

- Edit one or more lines in the rule set.

→ Use Processes to Build Rule Sets on page 130

- Run a branch of the rule set process tree for testing.

→ Execute Processes for Testing on page 151

At the end you should have a rule set, that—when executed entirely on a image data—detects the image objects of interest as defined by the image analysis task.

If you are already experienced in rule set development, you may continue with the extended workflow of rule set development focusing the strategic approach.

→ How to Approach Developing a Rule Set on page 324

Instructions About Developing Rule Sets

This User Guide provides three chapters about developing rule sets:

- **Start Developing Rule Sets** introduces the technology and the development workflow and explains the basic tools. This chapter enables you to create basic rule sets for simple image analysis tasks.

→ Start Developing Rule Sets on page 121

- **Develop Efficient Rule Sets** instructs you in using the available tools to build efficient and reusable rule sets and other ruleware. This chapter enables you to benefit from the tools of the development environment.

→ Develop Efficient Rule Sets on page 192

- **About Development Strategies** focuses on composing rule sets based on the **Definiens** Cognition Network Language (CNL). This chapter enables you to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ About Development Strategies on page 324

3.5 Process Data

Processing applies ruleware, which may be a rule set or a solution, to the image data. Thus, you receive image analysis results, and the results in turn enable you to evaluate the ruleware.

→ Process Data on page 361

There are different methods for processing data:

- Analyze a single project by executing manually a rule set in the **Process Tree** window.
- Analyze a single project by executing manually a solution in the **Analysis Builder** window.
- Analyze single or multiple projects in a **Workspace** in an automated environment.
- When you analyze single projects with large image size, the image analysis maybe speed up by creating subsets or tiles of a larger project or image.

3.6 Examine Results

After processing image data you can examine the results, thus allowing you to evaluate the used rule set.

→ Examine Results on page 379

Various methods enable you to explore and evaluate the image analysis results. In addition, you may want to use manual editing tools to influence the results by hand.

View and investigate the results of an image analysis.

Results of an image analysis can be both numerical and visual. Numerical results are statistics and result values. Visual results are represented by the classification of image objects.

You can view statistics and compare numerical result values of projects. Depending on the analysis rule set, certain export functionalities enable the calculation of result values in relation to both single projects and large data sets of a workspaces.

Additionally, you can open a project to view and investigate it. A number of viewing options enable various visualizations of analyzed projects. Various display options enable in-depth exploration of both the image object hierarchy and image objects.

3.6.1 Manual Editing

Manual editing of image objects allows you to manually modify the result of an image analysis. It can be applied to highlight or reclassify certain objects or to quickly improve the analysis result without re-editing the underlying rule set.

→ Manual Editing on page 391

Image objects can be manually merged with neighboring objects, assigned to a selected class, or split into smaller objects.

Additionally, thematic objects can be manually created, merged, cut, deleted, or saved as a shapefile.

3.7 Export Data

You have several options for exporting data from the system in order to use it in other applications. Analysis result data can be exported as graphic information in vector or raster format. Additionally, statistical information can be created.

→ Export Data on page 401

There are different basic methods to export analysis results:

- Data export triggered by a rule set that specifies where and how data is exported.
- Data export triggered by an action that specifies where and how data is exported.
- Data export initiated by **Export** menu commands. This is used to report data on a currently open project.

4 Start Definiens Developer

Contents in This Chapter

This chapter provides an introduction for starting **Definiens Developer**. In addition, it gives you an overview of the user interface.

4.1 Start Definiens Software

Start a **Definiens** client like **Developer**, **Architect**, **Analyst**, or **Viewer** with selectable portals offering predefined user interface arrangements.

A portal provides a selection of tools and user interface elements typically used for image analysis within an industry or science domain. However, most tools and user interface elements that are hidden by default are still available.

The following portals are available; the offered choice of portals depends on your **Definiens eCognition Server** edition:

Portal	Description	Available only for Definiens eCognition
Ell Life	Standard portal for the life sciences domain.	• Life Server
Cell	Recommended for the cell based image analysis. Standard portal for **Definiens Cellenger** application.	• Life Server
Tissue	Recommended for the tissue based image analysis. Standard portal for **Definiens Tissue Proliferation Assay** application.	• Life Server
Ell Earth	Standard portal for the remote sensing and geographic sciences domain.	• Earth Server

1. Go the **Windows Start** menu and choose **Start > All Programs >Definiens Developer > Definiens Developer**.

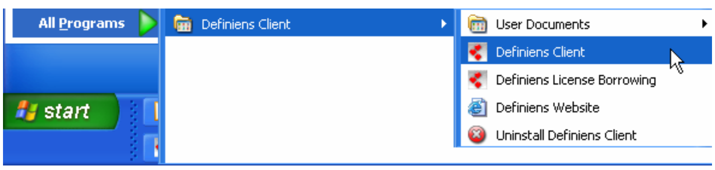


Figure 21: Start a Definiens client from the Windows Start Menu.

A selection dialog box opens.

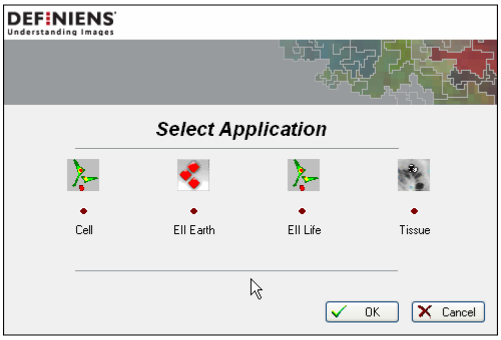


Figure 22: Selection dialog box offering different portals.

2. Select a portal.
3. Confirm with **OK**. The main window opens.

Tip

Start Multiple Clients

You can start and work on multiple ***Definiens Enterprise Image Intelligence Clients*** *simultaneously. This is helpful if you want to open more than one project concurrently. Note that you cannot interact directly between the running applications as they are running independently.*

4.2 Main Window

The main window contains the following elements:

- Menu bar, toolbars and status bar
- Project view
- Windows such as the **Image Object Information** window

You can add additional windows to your layout and hide them again, if you do not need them. Normally, additional windows find their place in the default layout. You can drag them to the desired position and size.

→ Handle Docking Windows on page 41

You can choose between different workflow views which are preset layouts of the user interface. A workflow view displays all required windows for each of the major workflow steps.

→ Use Workflow Views on page 40

4.2.1 Menu Bar

Each option in the menu bar contains a drop-down menu.

→ Menus on page 411

4.2.2 Toolbars

Toolbars contain buttons or list boxes to help you view or analyze your images.

→ Toolbars and Command Buttons on page 423

→ Customize on page 439

To check and modify the displayed toolbars, select **View** > **Toolbars** in the main menu bar. Alternatively, you can go to the **Toolbars** tab of the **Customize** dialog box.

Toolbars maybe docked at the edges of the main window. Also, they maybe undocked as floating toolbars. Holding Ctrl while dragging deactivates the magnetic snapping, allowing you to place the toolbars anywhere on your screen.

Right-clicking anywhere inside the menu bar and toolbar area allows you to access a context menu which provides a selection of the **View** menu options.

4.2.3 Status Bar

The status bar at the bottom of the **Definiens Developer** main window contains several fields which provide general information about your data.

Example: Cursor position in the project view, number of layers, number of objects, current layer.

Figure 23: Status bar.

4.2.4 Project View

The project view displays the image to be analyzed. In the project view, the image data and the analysis result are visualized. The project view is inside the main window. You can choose from a variety of display options which include viewing classifications, objects, features, layers, and so on.

→ View Data in Projects on page 93

4.2.5 Windows

Windows like **Image Object Information** facilitate the image analysis. They provide functions or controls for viewing and working with images. They can be opened using the appropriate menu items or by clicking the respective icons in the toolbars. Changes or selections in the windows are automatically shown in the project view.

Windows can be moved to almost any position on your screen.

→ Handle Docking Windows on page 41

Example: Some of the most important windows are **View Settings** and **Image Object Information**.

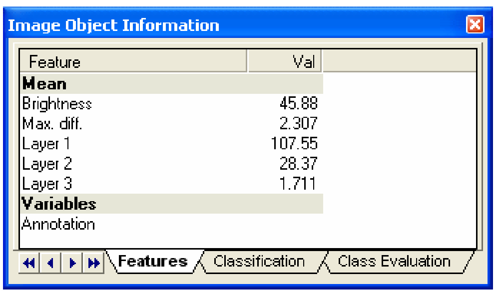


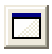
Figure 24: The Image Object Information window.

4.3 Workspace Window

In the **Workspace** window, you manage data with the help of projects.

→ Manage Projects in a Workspace on page 56

In the left pane of the window, there is a tree view with folders structuring your data. The right pane contains an entry for each layer of a project as well as additional information about the selected project.


Workspace

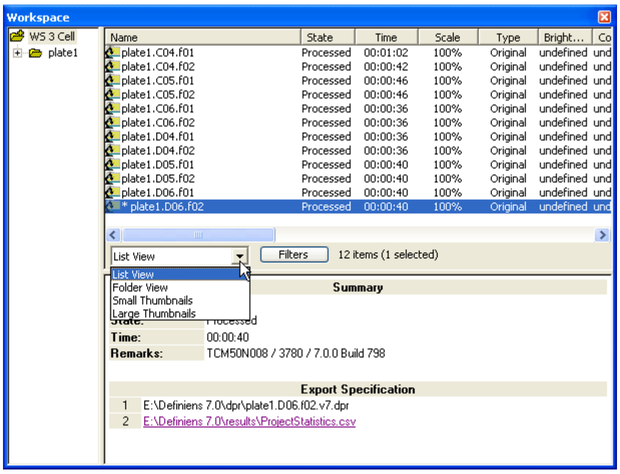


Figure 25: Workspace with drop-down view menu.

4.4 Use Workflow Views

Workflow views are preset layouts of the user interface displaying all required windows for each of the major workflow steps.

To activate, choose a workflow view on the **View** menu:

- Use the **Manage Data View** when loading and managing the images.
- Use the **Analysis Builder View** to assemble and configure the actions.
- Use the **View Results View** to analyze the results of your data processing.
- Use the **Develop Rule Set View** for developing rule sets.


Manage Data View


Analysis Builder View


View Results View


Develop Rule Set View

If you modify a current workflow view, for example, by moving or resizing a window, you can save this setting. Just choose **View > Save Current View** on the main menu bar. When you return to the specific workflow view, the modified settings of the user interface layout are used.

If you want to return to the default settings of all workflow views and other user interface settings select **View > Restore Default** on the main menu bar.

→ Customize the User Interface on page 42

4.5 Handle Docking Windows

Windows such as **Image Object Information** can be moved anywhere on your main window.

By default, these windows are magnetic and will snap to the edges of the main window. You can change the default by deselecting **Enable Docking** on the **Window** menu.

→ User Interface Elements on page 410

You can change the position of a window as follows:

- Drag the window to the desired position by clicking on the window title bar. The outlines of the window are visible while you drag, giving you an indication of where the window will roughly snap to.
- Change the size of the window by shifting the movable side bars between window.

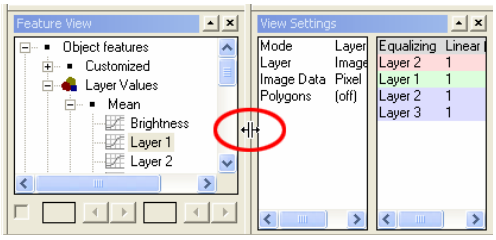


Figure 26: Shift the borders between windows to change their size.

- Windows can be aligned horizontally or vertically. For a horizontal alignment, drag the window down (or, for a horizontal alignment at the top, up). For a vertical alignment, drag the window to either sides of the main window.

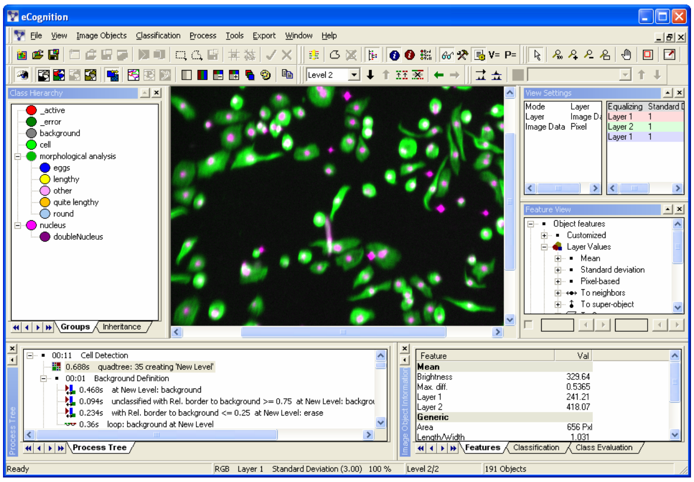


Figure 27: Main window with docked windows on the left, right and bottom (Image courtesy of Cenix BioScience GmbH).

Windows can be undocked so they float on your screen. This is especially useful if you are working with two monitors.

You can undock a window as follows:

- Drag the window to the center of the screen. The window is undocked when the outlines change to a small rectangle while you drag.
- Press Ctrl while dragging. This allows you to move the undocked window anywhere on your screen. If you release the Ctrl key while still dragging, the window will snap to the edge of the main window again.

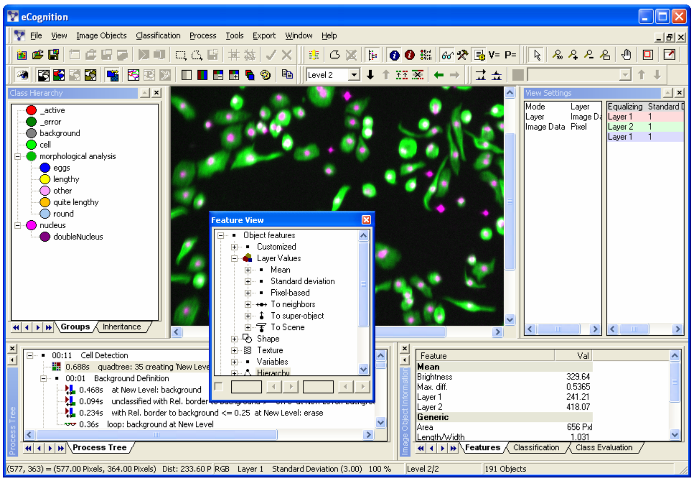


Figure 28: Main window with undocked Feature View window dragged inside the project view while pressing the [Ctrl] key.

4.6 Customize the User Interface

You can customize the user interface according to your preferences and modify each preset layout separately.

→ Use Workflow Views on page 40

After closing and reopening **Definiens Developer**, the customized user interface is loaded.

To customize and reset toolbars, keyboard shortcuts and menus, choose **View > Customize** on the main menu bar to open the **Customize** dialog box.

→ Customize on page 439

To go back to the default settings, select **View > Restore Default** on the main menu bar.

To change the appearance of the application, go to **View >Application Appearance** and select one of the following:

- **Office 2000**
- **Office XP**
- **Office 2003**
- **Windows XP**
- **Visual Studio 2005**

5 Load and Manage Data

Contents in This Chapter

This chapter gives you an overview on how to load and manage data. This includes opening workspaces and loading projects.

5.1 Work with Projects

A single project is used to develop your image analysis solution.

A project refers to image data (called scenes) and related analysis results. Image analysis extracts information from a scene and adds it to the project.

Multiple projects are bundled and administrated in a workspace.

→ Manage Projects in a Workspace on page 56

5.1.1 Open a Separate Project

A separate project is a project that is not part of a workspace. You can open a separate project and investigate it in the project view.

Precondition: You need a **.dpr** file containing project. To view results, you need a **.dpr** file containing an analyzed project.

1. To view a project that is not part of a workspace, you must open it from a separate **.dpr** file. Choose **File > Open Project** on the main menu bar. The **Open Project** dialog box opens.

Open Project

2. Select a project stored as **.dpr** file.
3. Click **open** and the project will display in the project view.

If you try to open a second project in one **Definiens Developer** instance, the other one closes automatically. To open and view more than one project simultaneously, you have to start multiple **Definiens Developer** instances of Definiens Developer; each for opening another project.

→ Close a Project on page 56
→ Start Definiens Software on page 37

Note

If an existing project or its image data have been moved, the references to the image data may have got lost. If the **Locate** *image file* dialog box opens, the image data need to be relocated.

5.1.2 Create a New Project

When creating a project, you import image layers and optionally thematic layers into a new project. You can rearrange the image layers, select a subset of the image or modify the project default settings. In addition, you can add metadata.

An image file contains one or more image layers. For example, an RGB image file contains three image layers, which are displayed through the Red, Green and Blue channels (layers).

→ RGB on page 486

1. To open the **Create Project** dialog box, do one of the following:

   - Choose **File > New Project** on the main menu bar.
   - Go to the left pane of the **Workspace** window, right-click a folder and choose **Add Project** on the context menu.

   The **Import Image Layers** dialog box opens.

Create New Project

→ Create a New Project Within a Workspace on page 87

2. Select image data.

   To open some specific file formats or structures, you have to proceed as follows:

   2.1. First select the correct driver in the **Files of type** drop-down list box.

→ Supported Image Formats on page 47

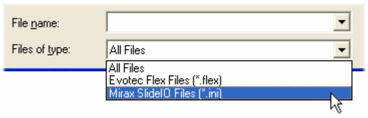


Figure 29: Select a Driver.

   2.2. Then select from the main file in the files area.

   If you select a repository file (archive file), another **Import Image Layers** dialog box opens in which you can select from the contained files.

→ Import Image Layers on page 448

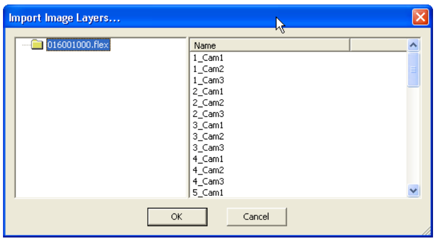


Figure 30: Import Image Layers dialog box listing items of a repository file.

   Press **Open** to display the **Create Project** dialog box.

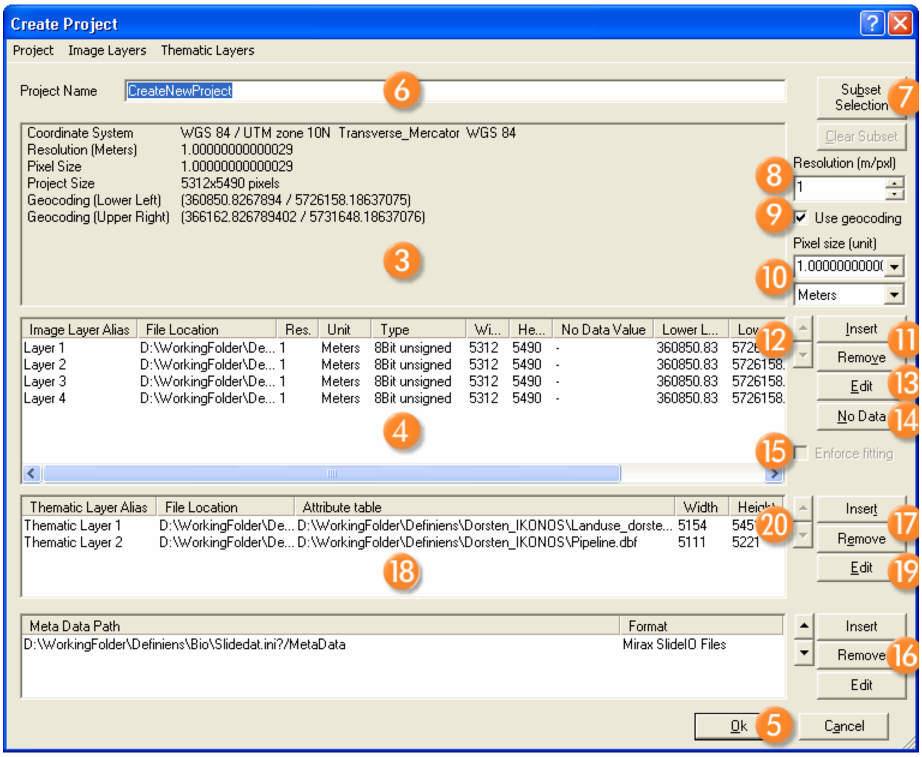


Figure 31: Create Project dialog box.

3. The project header area 3 displays project properties concerning pixel size, coordinate system, geocoding and other.
4. All preloaded image layers are displayed 4 along with their properties.
5. Click **OK** 5 to create the new project. If a workspace is open, the new project is displayed in the right pane of the **Workspace** window. Otherwise, the image is displayed in the project view.

Options

6 Edit the **Project Name**.

7 Define or clear a subset if the complete image to be analyzed is relatively large and you want to work on a smaller area to save processing time. For details, see section **Subset Selection**.

→ Subset Selection on page 461

8 If you want to rescale the scene during import, edit the scale factor in the text box corresponding to the scaling method used: **Resolution (m/pxl)**, **Magnification (x)**, **Percent (%)**, or **Pixel (pxl/pxl)**.

→ About Scales on page 49

9 To use the geocoding information from an image file to be imported, select the **Use geocoding** checkbox.

10 For feature calculations, value display, and export, you can edit the **Pixels size** and select another **Unit**. If you keep the default **Auto**, the unit conversion is applied according to the unit of the coordinate system of the image data as follows:

- In case of included geocoding information, the pixel size is equal to the resolution.
- In other cases, pixel size is 1.

In special cases you may want to ignore the unit information from the included geocoding information. To do so, deactivate the **Initialize unit conversion from**

→ Options on page 452

input file item of the **Options** dialog box.

4 List of image layers. To select an image layer, click it. To select multiple image layers, press Ctrl and click on the image layers.

11 To insert an additional image layer, do one of the following:

- Click the **Insert** 11 button.
- Choose **Image Layers > Insert** on the menu bar of the dialog box.
- Right-click inside the image layer list 4 and choose **Insert** on the context menu.

The **Import Image Layer** dialog box opens. Select the desired data.

To remove one or more image layers, select the desired layers and click **Remove**.

12 Select an image layer and use the up and down arrows to change the order of the layers.

4 You can edit layer aliases. Double-click or right-click an image layer and choose **Edit**. The **Layer Properties** dialog will open.

→ About Layer Aliases on page 54

→ Layer Properties on page 449

13 Alternatively you can click the **Edit** 13 button.

14 To set the value of those pixels that are not to be analyzed, click **No Data**. The **Assign No Data Values** dialog box opens.

→ Assign No-Data-Values on page 433

15 If you import image layers of different sizes, the largest image layer dimensions determine the size of the scene. When importing without using geocoding 9, the smaller image layers are handled as follows: Smaller image layers keep their size if the **Enforce Fitting** check box is cleared. If you want to stretch the smaller image layers to the scene size, select the **Enforce Fitting** check box.

16 Load **Metadata** files. Click the **Select** button to edit the path. An **Import Metadata** dialog box opens, where you can select a file.
When importing metadata you must convert available external metadata to an internal metadata definition. Select an item in the list and click the **Edit** button to open the **Metadata Conversion** dialog box.
To remove a metadata file, select and click the **Remove** button.

→ Convert Metadata to Provide it to the Feature Tree on page 52

Thematic Layer Options

17 To insert a thematic layer, do one of the following:

→ Working with Thematic Layers on page 283

- Click the **Insert** 17 button.
- Choose **Thematic Layers > Insert** on the menu bar of the dialog box.
- Right-click inside the thematic layer list 18 and choose **Insert** from the context menu.

The **Import Thematic Layer** dialog box, which is the equivalent of the **Import Image Layers** dialog box, opens.

18 List of thematic layers. To select a thematic layer, click it. To select multiple image layers, press Ctrl.

18 You can edit thematic layer aliases. Double-click or right-click a thematic layer and choose **Edit**. The **Layer Properties** dialog will open.

→ Layer Properties on page 449

19 Alternatively you can click the **Edit** 19 button.

17 To remove one or more image layers, select the desired layers and click **Remove**.

20 If you use more than one thematic layer, select a layer and use the up and down arrows to change their order.

> **Note**
>
> There is no option to assign **No Data Values** for thematic layers.

5.1.2.1 Supported Image Formats

Various file drivers enable you to import the following image data formats:

Import File Format (Driver)	File Exten-sion	Com-pression Algorithm	Known Restric-tions	Geo-Refer-encing	Win-dows	Linux	Read/ Write	Available only for Definiens eCognition **Life** Server	Available only for Definiens eCognition **Earth** Server
8 bit transparent PNG	**.png**	-	-	-	Yes	Yes	RW	X	X
Aperio	**.svs**	-	-	-	Yes	**No**	R	X	
Aperio Composite Images	**.aci**	-	-	-	Yes	Yes	R	X	
Applied Imaging	**.xml**	-	-	-	Yes	**No**	R	X	
Bacus WebSlide	**.ini**	-	-	-	Yes	**No**	R	X	
Cellomics DIB	**.dib**	-	-	-	Yes	Yes	R	X	
CEOS (Spot for instance)		-	-	**No**	Yes	Yes	R		X
ENVI .hdr Labelled Raster	**.hdr**	-	-	Yes	Yes	Yes	RW		X
Envisat Image Product	**.n1**	-	-	**No**	Yes	Yes	R		X
Erdas Imagine	**.img**	-	-	**No**	Yes	Yes	RW		X
Erdas LAN	**.lan**	-	2 GB	Yes	Yes	Yes	R		X
Erdas LAN	**.gis**	-	2 GB	Yes	Yes	Yes	R		X
ESRI Arc/Info ASCII Grid	**.asc**	-	2 GB	Yes	Yes	Yes	R		X
ESRI Arc/Info Binary Grid	**.adf**	-	-	Yes	Yes	Yes	RW		X
ESRI Shape File	**.dbf**	-	-	Yes	Yes	Yes	RW		X
ESRI Shape File	**.shp**	-	-	Yes	Yes	Yes	RW	X	X
Evotec Opera II FLEX	**.flex**	**-**		-	Yes	**No**	R	X	
First Generation USGS DOQ	**.doq**	DOQ1	-	Yes	Yes	Yes	R		X
GE Raven FRM	**.frm**		-		Yes	Yes	R	X	
Graphics Interchange Format	**.gif**	GIF	2 GB	**No**	Yes	Yes	RW	X	X
GRASS Rasters	-	GRASS	-	Yes	Yes	Yes	R		X
Hierarchical Data Format Release 4	-	HDF4	-	Yes	Yes	Yes	RW		X
Hierarchical Data Format Release 5	-	HDF5	-	Yes	Yes	Yes	RW		X
Idrisi Raster	**.rst**	-	-	Yes	Yes	Yes	RW		X
JPEG JFIF	**.jpg**	-	4 GB	**No**	Yes	Yes	RW	X	X
JPEG 2000	**.jp2** **.j2k**	JPEG 2000	2 GB					X	X
JPEG 2000	**.jp2** **.j2k**	JP2KAK	-	**No**	Yes	Yes	RW	X	X

Microsoft Windows Device Independent Bitmap	**.bmp**	-	4 GB	**No**	Yes	Yes	RW	X	X
Military Elevation Data	**.dt0 .dt1**	DTED	-	Yes	Yes	Yes	RW		X
Mirax Slide IO	**.ini**	-	-	-	Yes	**No**	R	X	
NDF	**-**	NLAPS Data Format	-	**No**	Yes	Yes	RW		X
New Labelled USGS DOQ	**.doq**	DOQ2	-	Yes	Yes	Yes	R		X
NITF	**.nitf**	NITF 2.1	4 GB	Yes	Yes	Yes	RW		X
PCI Labelled AUX	**.aux**	pAUX	-	**No**	Yes	Yes	R		X
PCI Geomatics Database File	-	PCIDISK	-	Yes	Yes	Yes	RW		X
PCI Library v7.0	-	-	-	Yes[1]	Yes	Yes	RW		X
PCI Library v8.2	-	-	-	Yes[1]	Yes	Yes	RW		X
PCI Library v9.0	-	-	-	Yes[1]	Yes	Yes	RW		X
PCI PIX file format	**.pix**	-	-	**No**	Yes	**No**	R		X
PCO SensiCam	**.b16**	-	-	**No**	Yes	**No**	R	X	
PCRaster	**.map**	PC Raster	-	Yes	Yes	Yes	RW		X
Portable Network Graphics	**.png**	-	-	**No**	Yes	Yes	RW	X	X
Tagged Image File	**.tif**	TIF	4 GB	Yes	Yes	Yes	RW	X	X
Tagged Image File (GeoTIFF)	**.tif**	GTif	4 GB	Yes	Yes	Yes	RW	X	X
USGS ASCII DEM	**.dem**	-	-	Yes	Yes	Yes	R		X
X11 Pixmap	**.xmp**	XPM	-	**No**	Yes	Yes	R	X	X

[1] Georeferencing is supported, though there may be some limitations in support of datums and ellipsoids. If ground control point (GCP) segments are present, the first will be used, and the rest ignored.

Other formats are available on request. Contact the Definiens sales team at **info@definiens.com**.

Note

For best results, we recommend use of uncompressed image formats because the information content is of higher quality than in formats using a compression that loses data. For example, an uncompressed **.tif** file will produce better analysis results than a compressed **.jpg** file.

Note

All import formats supported for image layers also apply for thematic layers. Additionally, **.shp** vector files are supported.

For polygon shapefiles and for **.img** files, the attribute table is automatically selected. For all other formats the respective attribute table has to be specifically indicated. From the **Load attribute table** dialog, the following file formats can be selected:

- **.txt** ASCII text files
- **.dbf** Dbase files
- **.csv** Comma separated values files

5.1.2.2 Maximum Image Size

The image size is limited theoretically, to 2^{62} pixels. However, testing has been performed only on images up to 20 gigapixel (141 421 x 141 421 = 2^{10} pixels) only.

Furthermore, you have to consider limitations in processing large projects while executing an image analysis.

→ About Processing Limitations on page 370
→ How to Handle Large Data Sets on page 346

5.1.2.3 About Scales

The scale of a scene or of image data describes the ratio of the depicted distance to the distance in reality. When working with scenes in projects and workspaces, **Definiens Developer** provides different types of scale modes used for displaying and calculating scale factors.

Scale Mode	Description	Example
Unit (m/pxl)	Resolution expressed in meters per pixel.	20 m/pxl
Magnification	Magnification factor used similar as in microscopy.	20x
Percent	Relation of the scale to the original scene scale.	20%
Pixels	Relation of pixels to the original scene pixels.	1:20 pxl/pxl

You can set the type of scale mode in the **Options** dialog box under the **Display scale with** item.

→ Options on page 452

You can check the scale mode in the following displays:

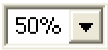
- **Scale** column in the right pane of the **Workspace** window

→ Manage Projects in a Workspace on page 56

- Scale text box in the **Create Project** and **Modify Project** dialog boxes located below the **Subset** buttons

→ Create a New Project on page 44
→ Modify a Project on page 54

- For **Magnification** only, the **Zoom** combo box changes.



Zoom

→ Zoom to a Magnification Zoom Value on page 104

The scale mode is used for calculating scaling operations when creating scene copies or copies of scene subsets.

→ Work at Different Scales on page 373
→ Rescaled Scene Copy on page 303
→ Rescaled Copy of a Scene Subset on page 304

Note

The scaling results may differ depending on the scale mode. Example: If you enter 40, you work at the following scales, which are calculated differently:

Options dialog box setting	Scale of the scene copy or subset to be created
Units (m/pixel)	40m per pixel
Magnification	40x
Percent	40% of the resolution of the source scene
Pixels	1 pxl per 40 pxl of the source scene

5.1.2.4 Import Image Layers of Different Scale

You can insert image layers and thematic layers with different resolutions (scale) into **Definiens Developer**.

They need not have the same number of columns and rows. To combine image layers of different resolution (scale), the images with the lower resolution—having a larger pixel size—are resampled to the size of the smallest pixel size. If the layers have exactly the same extent and geographical position, then geocoding is not necessary for the resampling of images.

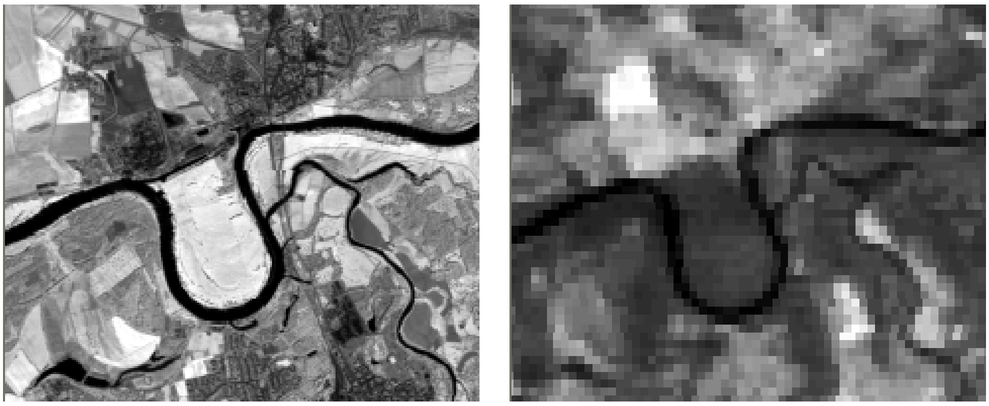


Figure 32: Left: Higher resolution - small pixel size. Right: Lower resolution - image is resampled to be imported.

5.1.2.5 About Geocoding

Geocoding is the assignment of positioning marks in images by coordinates. The position marks serve as geographic identifiers. but geocoding is helpful for life sciences image analysis too. Typical examples include working with subsets, at multiple magnifications, or with thematic layers for transferring image analysis results.

Geocoding information is automatically detected by **Definiens Developer** or can be entered manually. Images without geocodes create automatically a virtual coordinate system with 0/0 upper left and the unit of 1 pixel. For such images, geocoding represents the pixel coordinates instead of geographic coordinates.

Definiens Developer cannot reproject image or thematic layers. Therefore all image layers must belong to the same coordinate system in order to be read properly. If the coordinate system is supported, geographic coordinates from inserted files are detected automatically. If the information is not included in the image file but is nevertheless available, you can open a dialog for each layer to insert the geocoding information

→ Layer Properties on page 449

manually by double-clicking the appropriate layer or right-clicking and choosing **Edit**. This opens the **Layer Properties** dialog box.

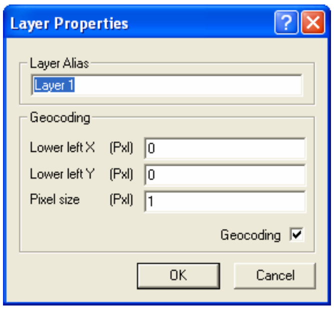


Figure 33: The Layer Properties dialog box allows you to edit the geocoding information.

5.1.2.6 Multisource Data Fusion

If the loaded image files are georeferenced to one single coordinate system, image and thematic layers with a different geographical coverage, size or resolution can be inserted.

This means that image and thematic data of various origins can be utilized simultaneously. The different information channels can be brought into a reasonable relationship to each other.

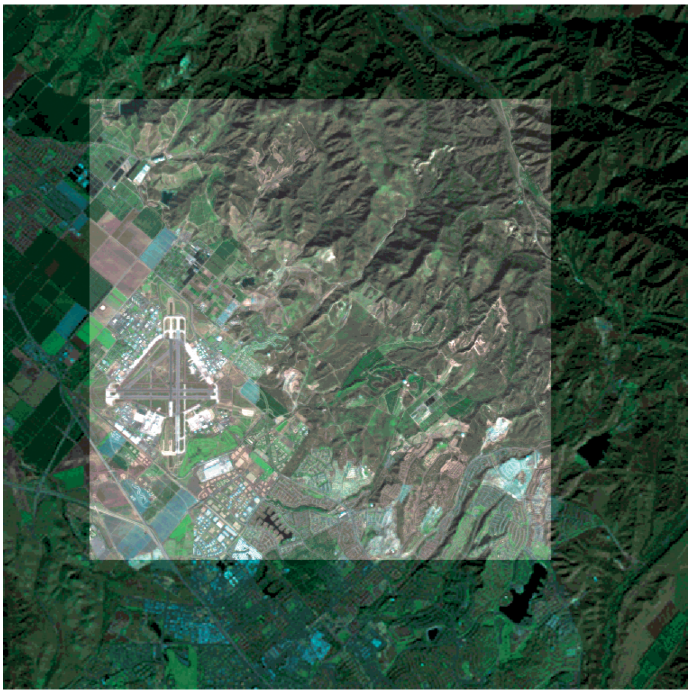


Figure 34: Layers with different geographical coverage.

5.1.2.7 Convert Metadata to Provide it to the Feature Tree

When importing data, you can provide a selection of available metadata. To do so, you have to convert external metadata to an internal metadata definition. This provides a selection of the available metadata to the feature tree and enables its usage in rule set development.

When developing rule sets, metadata definitions will be included in rule sets, enabling the serialization of metadata usage.

1. When importing metadata you must convert available external metadata to an internal metadata definition:
   - Within the **Create Project** dialog box, go to the **Metadata** group. After loading and selecting a metadata file, click the **Edit** button to open the **Metadata Conversion** dialog box.
   - Within the **Customized Import** dialog box, go to the **Metadata** tab and select **Conversion settings**. Click the ellipsis button to open the **Metadata Conversion** dialog box.

→ Create a New Project on page 44

→ Metadata Import Options on page 72

(ellipsis button)

2. Click the **Scan Metadata** button to display available external metadata items in the left column and the related internal metadata item name in the right column.

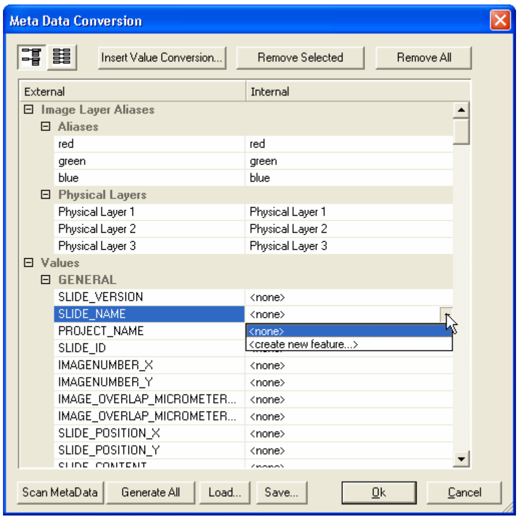


Figure 35: Metadata Conversion dialog box displaying sample metadata items.

3. The **Image Layer Aliases** group lists metadata items concerning image layers aliases.
4. The **Values** group below lists metadata items which can be selected for display in the feature tree. Select a metadata item and click the drop-down arrow button placed inside the right field. Select an existing feature from a drop-down list to be used as the name of the internal metadata item.

5. Alternatively, you can select **<create new feature>** from a drop-down list. The **Parameter Configuration** dialog box opens.

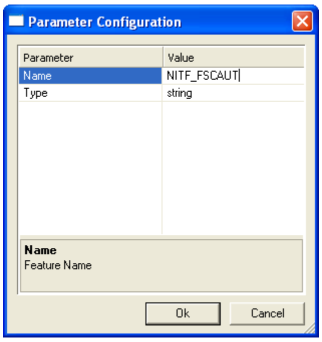


Figure 36: Metadata Item dialog box.

6. Define **Name** and **Type** of the internal metadata item. Click **OK** to create the feature.

7. To create features for all listed metadata items, click the **Generate All** button. All features are created as **string** item with the same name as the metadata item. To modify a feature, select one meta data item and click the drop-down arrow button. Select **<create new feature>** from the drop-down list to open the **Parameter Configuration** dialog box for editing.

8. After reconfirming and closing the internal metadata is displayed in the feature tree, for example in the **Feature View** window. Here you may also edit the features, if you need to.

→ Compare Feature Values by Using the Feature View on page 179

Options

9. List the values of metadata items by clicking the **Values** button. To go back to the list of internal names, click the **Names** button.


Values


Names

10. You can convert selected values of a metadata item. This enables you to use an internal value which is different from the external values.

    To do so, select a metadata item and click the **Insert Value Conversion** button. The **Enter External Value** dialog box opens asking for the external value that you want to be converted. It will be added in a **Value Conversion** sub group below the metadata item. In the **Internal** column on the right, you can edit the internal value to be used instead of the external value.

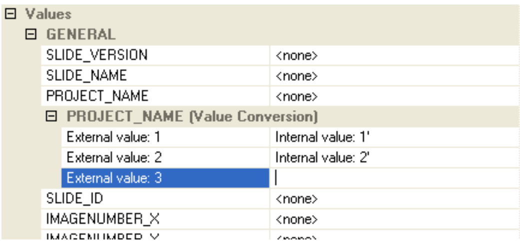


Figure 37: Value conversion of the metadata item PROJECT_NAME in the Metadata Conversion dialog box.

11. If you want to save the conversion settings of the **Metadata Conversion** dialog box to an **.xml** file, click the **Save** button.

12. It can be reloaded by using the **Load** button to facilitate the conversion of multiple metadata.

13. To remove metadata items use the **Remove Selected,** or **Remove All** button or press Del on the keyboard.

5.1.3 About Layer Aliases

A layer alias is a name for an image layer or a thematic layer that you can choose to replace standard layer aliases. You can use a descriptive name for easier identification of an image layer.

The basic function of layer aliases is to make the workflow more transparent and independent from the initial input data. Thus, when assigning and using layer aliases, all steps of your analysis become more transferable, because all layer-sensitive operations and features can optionally refer to the layer aliases or to the layers themselves. Hence, processes, classes and customized features become more independent and flexible.

Layer aliases are assigned **automatically** by the following actions:

- During project creation, standard layer aliases like **Layer 1**, **Layer 2** are always assigned.
- During project creation, if there is metadata concerning image layer aliases.
- Using customized import, you can define layer aliases to be assigned automatically while import of image layers and thematic layers.

→ Image Layers Import Options on page 67
→ Thematic Layers Import Options on page 70

- After project creation, when loading a rule set providing layer aliases definitions.

→ Assign Image Layer Alias on page 433

Layer aliases are assigned **manually** by the following actions:

- When creating a new project in the **Create Project** dialog box.

→ Create a New Project on page 44

- When modifying a project in the **Modify Project** dialog box by selecting **File > Modify Open Project** on the main menu bar.

→ Layer Properties on page 449

- After project creation by choosing **Process > Edit Aliases > Image Layers Aliases** or **> Thematic Layers Aliases** on the main menu bar.

→ Manage Layer Aliases on page 450

- After project creation by metadata conversion.

→ Convert Metadata to Provide it to the Feature Tree on page 52

5.1.4 Modify a Project

Modify a selected project by exchanging or renaming image layers or through other operations.

Depending on the project state, different modification options are available:

- Projects that have not yet been analyzed can be modified by exchanging image layers, editing image layer aliases, and editing the project name. You can also select a subset of the scene, assign the value of those pixels which should not be analyzed (**No Data** values), and edit the display unit and, if available, geocoordinates.

→ Subset Selection on page 461

- Projects that contain image objects as a result of image analysis can only be modified by exchanging image layers, editing image layer aliases, and editing the project name. If you want to use all available modification options, you can go to the **Project History** and roll back the project to the **Created** state. Alternatively, you can add a new project using the same image data.

→ Inspect the History of a Project on page 85

Precondition: Projects that contain objects as a result of image analysis have to be opened.

→ View a Project on page 84

1. To modify a project, do one of the following:

- Go to the **Workspace** window. Right-click a project and choose **Modify**.
- Open a project and choose **File > Modify Open Project** on the main menu bar.

2. The **Modify Project** dialog box opens.

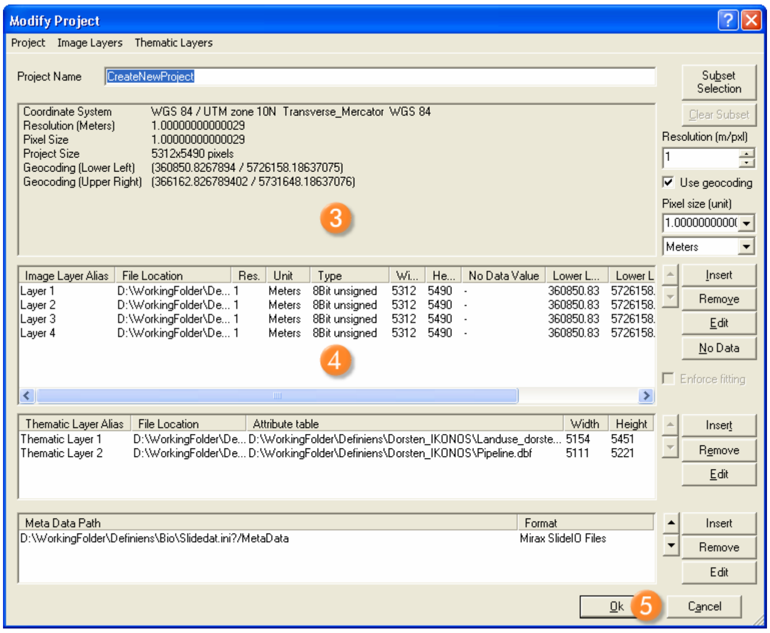


Figure 38: Modify Project dialog box.

3. The project header area 3 displays project properties concerning pixel size, coordinate system, geocoding and other.
4. All image layers are displayed 4 along with their properties.
5. Click **OK** 5 to modify the project. If a workspace is open, the new project is displayed in the right pane of the **Workspace** window. Otherwise, the image is displayed in the project view.

Options

For further options, see the **Create a New Project** section. Note that the only difference from creating a project is that the layer order cannot be changed in an open project.

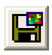
→ Create a New Project on page 44

5.1.5 Save a Project

Save the currently open project to a project file (extension **.dpr**).

To save a project, do one of the following:

- Choose **File > Save Project** on the main menu bar.
- Choose **File > Save Project As** on the main menu bar. The **Save Project** dialog box opens. Select a folder and enter a name for the project file (**.dpr**). Click the **Save** button to store the file.


Save Project

View settings are saved in a separate file with the extension **.dps**. If you want to move an **Definiens** project file and keep the view settings, move the settings file as well.

> **Note**
>
> If you created a project without an open workspace, the project cannot be saved as a part of the workspace. Just save this project as a separate **.dpr** file and import it to the workspace of your choice.

5.1.6 Close a Project

Close the currently open project.

To close a currently open project, do one of the following:

- Click the **Close Project** button.


Close Project

- Choose **File > Close Project** on the main menu bar.
- Create a new project or workspace.

→ Create a New Project on page 44
→ Create a New Workspace on page 57

- Open an existing project or workspace.

→ Open an Existing Workspace on page 84
→ Open a Separate Project on page 43

5.2 Manage Projects in a Workspace

Use the **Workspace** window, you view and manage all the data in your workspace. Here you open a workspace file that bundles multiple projects.

→ Work with Projects on page 43

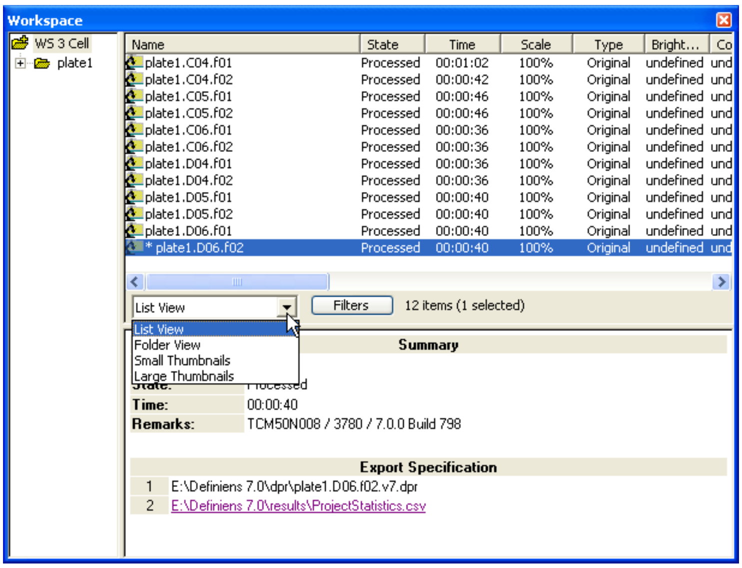


Figure 39: Workspace window with Summary and Export Specification and drop-down view menu.

The **Workspace** window is split in two panes:

- The left pane contains the **Workspace** tree view. It represents the hierarchical structure of the folders that contain the projects. Click a plus sign (+) to expand a folder. Click a minus sign (–) to collapse a folder. Select any folder by clicking.
- On the right pane, you see the content of the selected folder. You can choose among the display options **List View**, **Folder View**, and two **Thumbnail Views**.

→ Change the View in the Workspace on page 80

For **List View** and **Folder View** the right pane of the **Workspace** window contains information about the selected project: its state, scale, the time of the last processing and available remarks.

The **Scale** column displays the scale of the scene.

→ About Scales on page 49

Depending on the processed analysis, there are additional columns providing exported result values.

To change the column width, use the context menu to **Expand all columns** or **Collapse empty columns**.

Find Out More

Scene, Projects, Result Values, and Workspaces

*While a **scene** is just image data, image analysis extracts information from a scene. This information is represented by classified image objects.*

A scene is the combined input image data from a project or a workspace. A scene consists of an image (with at least one image layer) and optionally additional information related to the image content like thematic layers, geo coding, or geo information. Image analysis extracts result information from a scene and adds it to a project. This information is expressed by classified image objects. When viewing analyzed projects you can investigate both the input scene and the classification of image objects representing a visual result.

*Image objects are used to extract **result values,** representing numerical result of image analysis. These values are stored in **.csv** files.*

*A **workspace** bundles references to image data, project, exported result values, and the used rule set. Furthermore it includes the import and export templates, result statuses, and metadata. A workspace is saved as **.dpj** file.*

5.2.1 Open the Workspace Window

To open the **Workspace** window, do one of the following:

- Choose **View** > **Workspace** from the main menu.
- Click the **View Workspace** button


View Workspace

5.2.2 Create a New Workspace

Create a new workspace and edit the default name and file location for it.

1. To create a new workspace, do one of the following:

- Click the **Create New Workspace** button on the toolbar.
- Choose **File** > **New Workspace** from the main menu bar.

Create New Workspace

The **Create New Workspace** dialog box opens.

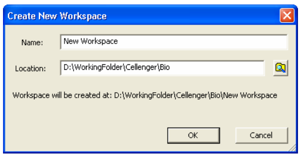


Figure 40: Create New Workspace dialog box.

2. Enter a **Name** for the new workspace. The default name is **New Workspace**.
3. Browse for a **Location** to save the new workspace.
4. Confirm with **OK** to create the workspace. It is displayed as the root folder in the left pane (the tree view) of the **Workspace** window.

Browse for Folders

The **.dpj** workspace file is created in a new folder with the name of the new workspace. By default this folder is the output root folder under which all **.dpr** project files, result images, statistics and others are stored.

If you want to define another output root folder, we recommend that you do this before you load scenes to the workspace. However, if you do, you can modify the path of the output root folder under which all output files are stored.

→ Move a Workspace on page 83

1. To modify the path of the output root folder choose **File** > **Workspace Properties** on the main menu bar. The **Workspace Properties** dialog box opens.

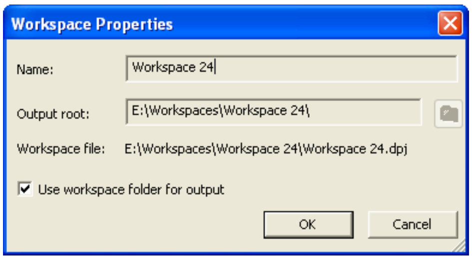


Figure 41: Workspace Properties dialog box.

2. Clear the **Use workspace folder** check box and change the path of the output root folder by editing it. Alternatively, you can click the ellipsis button and browse to an output root folder.

(ellipsis button)

5.2.3 Import Scenes to a Workspace

Before you can start working on data, you must import scenes to add image data to the workspace. Each scene is administrated as a project.

To add image data to a workspace, you must import scenes. You can select different predefined import templates according to the image acquisition facility producing your image data.

If you want to import just a single scene to a workspace, use the **Add Project** command.

→ Create a New Project Within a Workspace on page 87

Precondition: Create a workspace before importing scenes. Alternatively you can open an existing workspace and select a folder.

→ Create a New Workspace on page 57

1. To import scenes to a workspace, do one of the following:
   - Click the **Import Scenes** button on the **File** toolbar.
   - Choose **File** > **Predefined Import** on the main menu bar.

   The **Import Scenes** dialog box opens.

Import Scenes

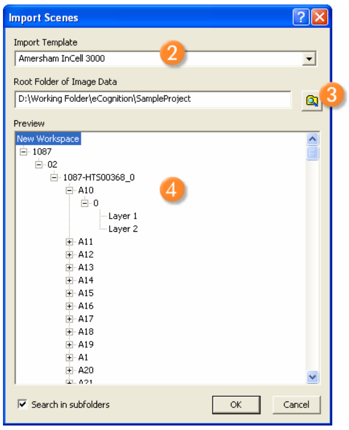


Figure 42: Import Scenes dialog box.

2. Select an **Import Template** 2 according to the file structure of your image data. This structure is determined by the image acquisition facility producing your image data.

→ Supported Import Templates on page 59

3. Browse 3 to open the **Browse for Folder** dialog box. Select a root folder that contains image data. Click **OK**.

Browse for Folders

4. The subordinate file structure of the selected image data root folder is displayed in the **Preview** 4 field. Click the plus sign buttons to expand a folders. Click the minus sign button to collapse a folder.

(expand)

(collapse)

5. Click **OK** to import scenes. The tree view on the left pane of the **Workspace** window displays the file structure of the new projects, each of which administrate one scene.

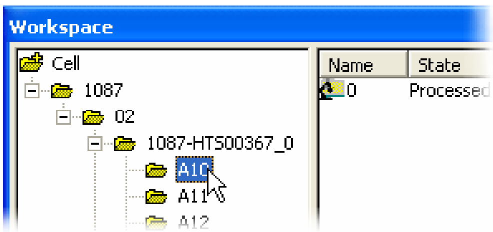


Figure 43: Folder structure in the Workspace window.

5.2.3.1 Supported Import Templates

You can use various import templates to import scenes. Each import template is provided by a connector. Connectors are available according to the used server edition of the **Definiens eCognition Server**.

- Generic import templates are available for simple file structures of import data. When using generic import templates, make sure that the file format you want to import is supported.

→ Supported Image Formats on page 47

- Import templates provided by connectors are used for loading the image data according to the file structure that is determined by the image reader or camera producing your image data. The standardized import templates are based on connectors, enabling the images from specific imaging devices to be used by **Definiens Developer.**

- In addition to the following table, customized import templates can be created for more specialized file structures of import data.

→ Customize the Import of Scenes into a Workspace on page 63

Generic Import Templates

Import Template (Connector)	Description	File Formats	File Based	Win-dows	Linux	Available only for Definiens eCognition **Life** Server	Available only for Definiens eCognition **Earth** Server
Generic - One File per Scene[1]	A scene may consist of multiple image layers. All image layers are saved to one file.	All	Yes	Yes	Yes	X	X
Generic - One Scene per Folder[1]	All files that are found in a folder will be loaded to one scene.	All	Yes	Yes	Yes	X	X
Generic - Image with common thematic[1]	Multiple geocoded image files with a common thematic layer.	All	Yes	Yes	Yes		X
Geo Coded - One File per Scene[1]	A scene may consist of multiple geocoded image layers. All image layers are saved to one file.	All	Yes	Yes	Yes		X
Geo Coded - One Scene per Folder[1]	All geocoded files that are found in a folder will be loaded to one scene.	All	Yes	Yes	Yes		X

[1] Generic Import Templates may support additional instruments or image readers not listed here. For more information about unlisted import templates contact **Definiens** support via internet on the website **http://www.definiens.com/support/index.htm**.

About Generic Import Templates

Image files are scanned into a workspace with a specific method, using import templates, and in a specific order according to folder hierarchy. This section lists principles of basic import templates used for importing scenes within the **Import Scenes** dialog box.

Generic - one file per scene

- Creates one scene per file.
- The number of layers per scene is dependent on the image file. For example, if the single image file contains three layers, the scene is created with three layers.
- Matching Pattern: **anyname**
- For the scene name, the filename without extension is used.
- **Geo-coded - one file per scene**: Reads the geo-coordinates separately from each readable image file.

Generic - one scene per folder

- All layers are taken from all image files.
- Creates a scene for each subfolder.
- Takes all image files from the subfolder to create a scene.
- If no subfolder is available the import will fail.
- The name of the subfolder is used for the scene name.
- **Geo-coded - one file per scene**: Reads the geo-coordinates separately from each readable image file.

Options

Images are scanned in a specific order in the preview or workspace. There are two options:

Select the check-box **Search in Subfolders**.

- Files in selected folder and all subfolders.
- Takes the first item in current folder.
- If this item is a folder, then steps into this folder and continues search there.

Clear the check-box **Search in Subfolders**.

- Only files directly in the folder.
- Alphabetical ascending.

For example, one might import the following images with this folder structure:

- [*selected folder*] > **[1]** > **[5]** > **1.tif** & **8.tif**
- [*selected folder*] > **[1]** > **[8]** > **5.tif**
- [*selected folder*] > **[1]** > **3.tif** & **7.tif**
- [*selected folder*] > **[3]** > **6.tif**
- [*selected folder*] > **2.tif** & **4.tif**

Supported Life Server Connectors

Available only for Definiens eCognition **Life** Server

Import Template (Connector)	Image Reader	File Formats	File Based	Win-dows	Linux	**EII Life** Portal[1]	**Cell** Portal[1]	**Tissue** Portal[1]
Applied Imaging		**.xml**	Yes	Yes	Yes	X		X
Bacus WebSlide		**.ini**	Yes	Yes	**No**	X		X
BD Pathway Bio Sciences		**.tif**	Yes	Yes	Yes	X	X	
Cellomics ArrayScan		**.dib** **.tif**	Yes	Yes	Yes	X	X	
Evotec Flex files	Opera II	**.flex**	Yes	Yes	**No**	X	X	
Evotec Opera	Opera I	**.tif**	Yes	Yes	Yes	X	X	

GE In Cell Analyzer 3000	GE Incell 3000	**.frm**	Yes	Yes	Yes	X	X	
GE In Cell Analyzer 1000	GE Incell 1000	**.tif**	Yes	Yes	Yes	X	X	
Generic One File per Scene	Generic Instruments[2]	All	Yes	Yes	Yes	X	X	X
Generic One Scene per Folder	Generic Instruments[2]	All	Yes	Yes	Yes	X	X	X
Maia Scientific MIAS		**.tif** and **.jpeg**	Yes	Yes	Yes	X	X	
Molecular Devices Discovery 1		**.tif**	Yes	Yes	Yes	X	X	
Molecular Devices MetaXpress Plate with Timepoint	Molecular Devices MetaXpress, Molecular Devices ImageXpress	**.tif**	Yes	Yes	Yes	X	X	
Zeiss Mirax Slide IO		**.ini**	Yes	Yes	**No**	X		X

[1] Select the appropriate portal when starting Definiens Developer.

→ Start Definiens Software on page 37

[2] Generic Import Templates may support additional instruments or image readers not listed here. For more information about unlisted import templates contact **Definiens** support via internet on the website **http://www.definiens.com/support/index.htm**.

→ Generic Import Templates on page 60

Supported Earth Server Connectors

Available only for Definiens eCognition **Earth** Server

Import Template (Connector)	File Formats	File Based	Windows	Linux	**EII Earth** Portal[1]
Arc/Info Binary Grid	**.adf**	Yes	Yes	Yes	X
Envisat ASR Files	**.N1**	Yes	Yes	Yes	X
Erdas Imagine Images	**.img**	Yes	Yes	Yes	X
Erdas LAN	**.lan** **.gis**	Yes	Yes	Yes	X
ESRI ASCII Grid Files	**.asc**	Yes	Yes	Yes	X
Generic - Image with common thematic[1]	All	Yes	Yes	Yes	X
Geo Coded - One File per Scene[2]	All	Yes	Yes	Yes	X
Geo Coded - One Scene per Folder[2]	All	Yes	Yes	Yes	X
GeoSoft Grid Exchange Format	**.gxf**	Yes	Yes	Yes	X
GeoSoft Grid File 2.0	**.grd**	Yes	Yes	Yes	X
Geotif	**.tif**	Yes	Yes	Yes	X
Gould LIPS	**.lip**	Yes	Yes	Yes	X
Idrisi Raster	.rst	Yes	Yes	Yes	X
Intergraph Raster	-	Yes	Yes	Yes	X
National Imagery Transmission	**.nitf**	Yes	Yes	Yes	X
NCSA Hierarchical Data Format	**.hdf4** **.hdf5**	Yes	Yes	Yes	X
NTIF 2.0	**.ntif**	Yes	Yes	Yes	X

PCIDSK	**.pcidisk**	Yes	Yes	Yes	X
ProSmart SAR Files		Yes	Yes	Yes	X
SPANSRaster		Yes	Yes	Yes	X

[1] If different portals are available, you can select one when starting Definiens Developer.

→ Start Definiens Software on page 37

[2] Generic Import Templates may support additional instruments or image readers not listed here. For more information about unlisted import templates contact **Definiens** support via internet on the website **http://www.definiens.com/support/index.htm**.

→ Generic Import Templates on page 60

5.2.3.2 Customize the Import of Scenes into a Workspace

Import multiple scenes from an existing file structure into a workspace and save the procedure as an import template.

→ Examples of Customized Imports on page 75

The customized import is based on a conceptual approach as follows:

First, the user defines a master file, which functions as a sample file and allows identification of the scenes of the workspace.

Second, the user defines the individual data that represent a scene by defining a search string. A search string is a file path pattern defined by variables (for example, folder names or layer names) and static parts (normal text). To ensure the correct combination of files, both the search strings for the master file as well as the search strings for the individual files should be edited to contain only static parts. The more information the system can extract from a search string, the more reliable the data import.

Perform a Customized Import

To perform a customized import, you first define a master file which works as sample file and allows identification of the scenes to be imported. After testing the suggested import configuration, you can modify various import options. The final customized import is ready to be saved as an import template that you can reuse for similar data structures.

Preconditions:

- The file structure of image data to be imported must follow a pattern.
- You need to create a workspace before importing scenes.

1. To perform a customized import of scenes to a workspace, do one of the following:
   - Go to the left pane of the **Workspace** window. Right-click a folder and, if available, choose **Customized Import** from the context menu.
   - Choose **File** > **Customized Import** on the main menu bar.

   The **Customized Import** dialog box opens on the **Workspace** tab.

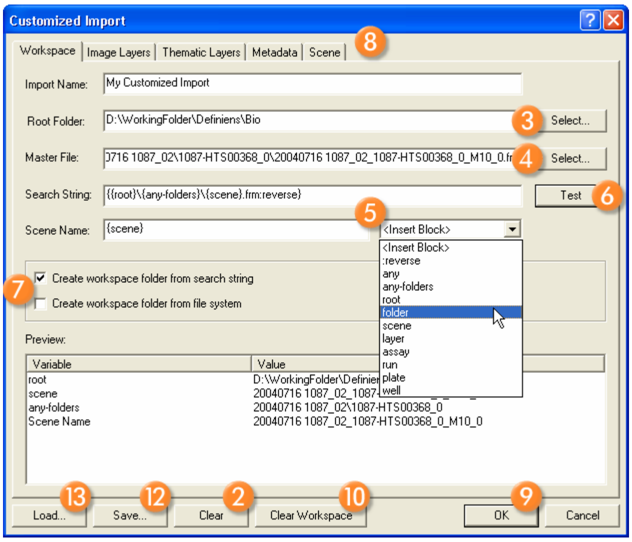


Figure 44: Customized Import dialog box.

Overview

Customized Import Dialog Box

*On the **Workspace** tab of the **Customized Import** dialog box, you define a master file, do some basic settings and test both. Only after selecting a master file can you access the other tabs that enable you to modify the import template:*

*On the **Image Layers** tab you can configure image layers of scenes to be imported.*

*On the **Thematic Layers** tab, you can configure thematic layers of scenes to be imported.*

*On the **Metadata** tab, you can load **Metadata** into the projects to be created.*

*On the **Scene** tab, you can configure image layers of scenes to be imported.*

2. Click the **Clear** button 2 before configuring a new import. Thus, you remove all present settings including any invisible ones.

3. **Select** the **Root Folder** 3 which is a folder under which all the image data you want to import is stored. The image data files to import can be structured into multiple subfolders. To allow a customized import, the structure of image data storage has to follow a pattern which will be analyzed in the next steps. The **Root Folder** text box displays the path.

4. **Select** a **Master File** 4 within the root folder or its subfolders. Depending on the file structure of your image data defined by your image reader or camera, the master file may be both a typical image file and a header file. While the **Master File** text box 5 displays the path below the root folder, the **Search String** text box displays a textual representation of the sample file path used as a pattern for the searching routine. The **Scene Name** 5 text box displays a textual representation of the name of the scene that will be used in the workspace window after import.

5. You have to edit the **Search String** 5 and the **Scene Name** 5 if the automatically generated ones do not exactly meet your needs. Place the curser in the text boxes and edit manually or by inserting blocks from the **<Insert Block>** drop-down list box 5. You have to comply with the following search string editing rules:

   - The search string has to start with **{root}**.

   - All static parts of the search string have to be defined by normal text.

- All variable parts of the search string can be defined by using blocks representing the search items which are sequenced in the search string:

Block	Description	Usage
:reverse	Starts reading from the end instead of the beginning.	{*part of a file name***:reverse**} is recommended for reading file names, because file name endings are usually fixed.
any	Represents any order and number of characters.	Used as wildcard character e.g. **{any}.tif** for TIFF files with an arbitrary name.
any-folders	Represents one or multiples of nested folders down the hierarchy under which the images are stored.	**{root}\{any-folders}\{any}.tif** for all TIFF files in all folders below the root folder.
root	Represents a root folder under which all image data you want to import is stored.	Every search string has to start with **{root}**.
folder	Represents one folder under which the images are stored.	**{root}\{folder}\{any}.tif** for all TIFF files in a folder below the root folder.
scene	Represents the name of a scene that will be used within the workspace after import.	**{root}\{scene}.tif** for TIFF files whose file names excluding the extension will be used as scene names.
layer	Represents the name of an image layer.	

- Use a backslash \ between a folder block and its content.
- Use {*block name***:n}** to specify number of characters of a searched item.
- For advanced editing, you can use regular expressions (a defined grammar used in complex pattern searching).
- Specific blocks for plates as used in cell biology:

Available only for Definiens eCognition **Life** Server

assay	Represents the name of an assay.	For plates only
run	Represents the run of an assay.	For plates only
plate	Represents the name of a plate.	For plates only
well	Represents the name of a well.	For plates only

- **Plate** and **well** blocks must be provided in the search string to allow proper usage of the **Plate View** window.
- If plate names are defined by folder names make sure that the folder name is included in the **Sample File** text box e.g. *path***AKT 10x***well name.xyz*.

6. Press the **Test** button 6 to preview the naming result of the **Master File** based on the **Search String**. If the **Preview** is not satisfying, go back to the previous steps.
7. Define the folder structure within the workspace by using the check boxes in the **Customized Import** dialog box. This structure determines in which folders and subfolders the projects with the scenes are located. This is important for easy access and for some specific analysis tools.

In the following table you can compare available folder structures listed with the corresponding check box settings:

Resulting folder structure in the workspace	**Create workspace folder from search string** check box 7	**Create workspace folder from file system** check box 7
Do not create any folders: All projects with the scenes are located directly in the workspace without any folder structure. Root Folder Sub-folder 1 Images of Date A Workspace Images of Date A	☐	☐
Copy the folder structure of the image data to import: Each folder of the import data is mapped in the workspace. Root Folder Sub-folder 1 Images of Date A Sub-folder 2 Images of Date B Images of Date C Images of Date D Workspace Sub-folder 1 Images of Date A Sub-folder 2 Images of Date B Images of Date C Images of Date D	☐	☑
Create folders according to the search string: Every search item found according to the search string, which corresponds to a logical item (e.g. a folder or a plate), is mapped to a separate workspace folder. Using this can be more satisfying as image files are recognized according to the search pattern instead of their location. Root Folder Sub-folder 1 Images of Date A Sub-folder 2 Images of Date B Images of Date C Images of Date D Workspace Folder 1\Date A Images of Date A Folder 2\Date B Images of Date B Folder 2\Date C Images of Date C Date D Images of Date D	☑	☐
Create many folders and subfolders: Each folder of the import data and additionally every search item according to the search string is located in a separate folder in the workspace. Root Folder Sub-folder 1 Images of Date A Workspace Sub-folder 1 Sub-folder1\Date A Images of Date A	☑	☑

8. Define the individual data that represents the scene. To ensure the correct combination of files, both the search strings for the master file as well as the search strings for the individual files should be edited to contain all static parts that are not recognized by the search string of the master file. To check and edit these settings go to the other tabs 8 of the **Customized Import** dialog box. Note, that you cannot access other tabs until a master file is defined.

→ Image Layers Import Options on page 67
→ Thematic Layers Import Options on page 70
→ Metadata Import Options on page 72
→ Metadata Import Options on page 72

9. To execute a customized import, click **OK** 9 to load the scenes to the workspace.

10. We recommend checking the executed import in the **Workspace** window. If you are not satisfied with the import, reopen the **Customized Import** dialog box and click the **Clear Workspace** button ⑩ to remove all imported scenes from the workspace. Then go back to step 2.

11. To save the settings of the **Customized Import** dialog box as an import template for similar data sets, reopen the **Customized Import** dialog box which displays the last settings.

12. Define a name in the **Import Name** text box and click the **Save** button ⑫ to store the settings as file. If you use the default folder for saving, the new import template will be easily available in the **Import Scenes** dialog box.

→ Import Scenes to a Workspace on page 58

13. To load an import template to the **Customized Import** dialog box, click the **Load** button ⑬. Note that the **Root Folder** and **Sample File** are not defined to allow the usage as import template.

Image Layers Import Options

On the **Image Layers** tab of the **Customized Import** dialog box, you can configure image layers of scenes to be imported. That way, you modify the import template concerning the following options:

- Add image files.
- Remove image layers and image files.
- Define a special search string for files.
- Select image format driver for importing.
- Modify image layer aliases.
- Define No-data handling.

Preconditions:

→ Perform a Customized Import on page 63

- Define a master file on the **Workspace** tab. Without a master file, you cannot access the **Image Layers** tab.
- Before modifying the image layer options, we recommend that you first test the customized import with the default settings of the **Workspace** tab.

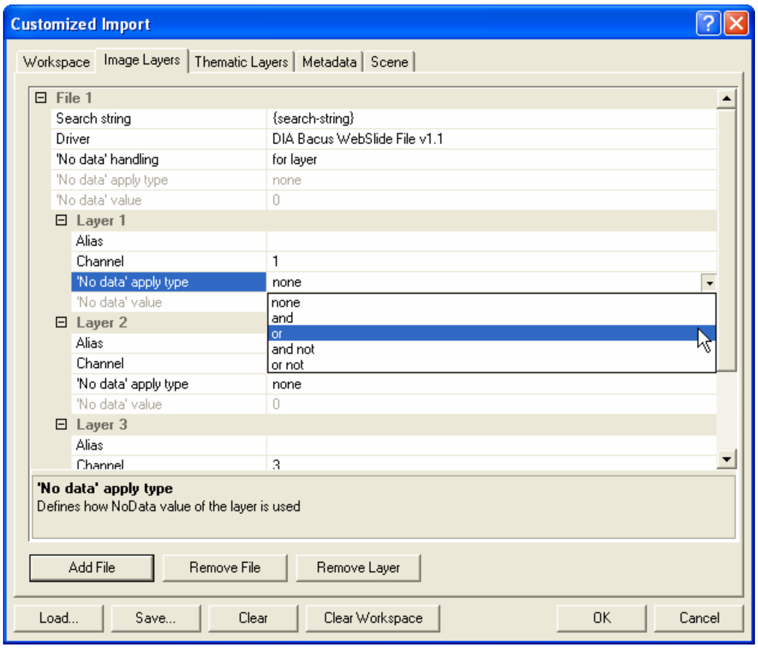


Figure 45: Image Layers tab of the Customized Import dialog box.

1. The **Image Layers** tab lists the **Files** to be imported with the included image **Layers**. Image layers are sequenced according to their **Channel** number.
2. To add files, click the **Add File** button.
3. To remove a file, select it and click the **Remove File** button. You cannot remove the last remaining file.
4. To remove a image layer, select it and click the **Remove Layer** button. You cannot remove the last remaining image layer of a file.

Other Options

Search String

You can define or modify the **Search string** for the image layers to be imported. In the corresponding the drop-down list you can select:

- **{search-string}** to use the scene string as defined on the **Workspace** tab.
- {*copy of the currently applied search string*} will not be updated while the scene string changes. Can be used for further modifications by editing.
- **From file** to load a file name used to adapt the search string. We recommend using this option. The software automatically detects the proper string based on the information provided by the search string. If the software fails to do so, this is an indication that the search string may be incorrect.

Driver

Keep the default **<auto>** to automatically select the appropriate image format **Driver**. If you require a special file format driver you can select it on the drop-down list.

Alias

The image layers are automatically named with numbers, in other words, **Layer 1**; **Layer 2**; **Layer 3**. Alternatively, you can define aliases for image layers. To do so, you have to initiate all image layers by adding them to the **Image Layers** list on the **Scene** tab. Do not forget to assign the correct channel index as follows.

→ About Layer Aliases on page 54

Channel

Display of the **Channel** number of the image layer as defined by the image file.

No-Data Settings

By default, **none** is selected, that means no-data-values are not defined. That means all pixels of all values can be analyzed later. If you want to exclude pixels with a specific gray value from a later image analysis, you can set the value of those pixels that are not to be analyzed.

→ Assign No-Data-Values on page 433

1. Set the **No-data handling** method:

   - Select **for file** to define the same no-data settings for all image layers of the file. Only the no-data settings in the **File** group are active.
   - Select **for layer** to define no-data settings for each image layers separately. Only the no-data settings in the **Layer** groups are active.

2. Set the **No-data apply type**. Select a method for assignment of no-data-values:

   - **none**: No-data-values are not defined. That means all pixels of all values can be analyzed later.
   - **and**: Select to assign only those overlapping no-data-areas that all images have in common. This corresponds to the **Intersection** check box of the **No-Data-Value** dialog box.
   - **or**: Select to assign the no-data-areas of all individual layers for the whole project, i.e. if a no data value is found in one layer, this area is treated as no-data-area in all other layers. This corresponds to the **Union** check box of the **No-Data-Value** dialog box.
   - **and not**:
   - **or not**:

3. Edit the **No-data-value** to a specific gray value defining those pixels to be excluded from later image analysis.

Thematic Layers Import Options

On the **Thematic Layers** tab of the **Customized Import** dialog box, you can configure thematic layers of scenes to be imported. That way, you modify the import template concerning the following options:

- Add and remove thematic layers.
- Define a special search string for thematic layers and attribute files.
- Modify thematic layer aliases.
- Select file format drivers for importing.
- Change the channel index values.
- Modify attribute table column aliases.
- Modify vector file parameters.

Preconditions:

→ Perform a Customized Import on page 63

- Define a master file on the **Workspace** tab. Otherwise, you cannot access the **Thematic Layers** tab.
- Before modifying the thematic layer options, we recommend that you first test the customized import with the default settings of the **Workspace** tab.

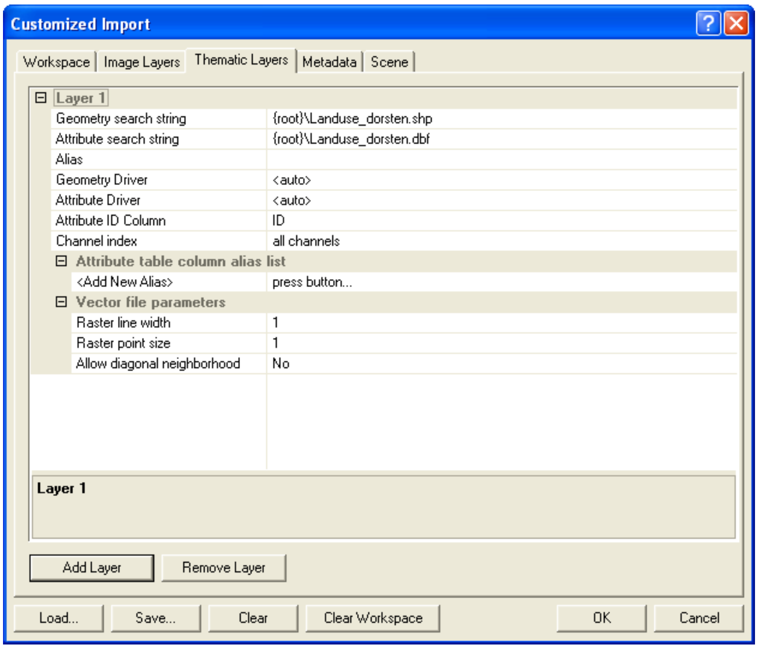


Figure 46: Thematic Layers tab of the Customized Import dialog box.

1. The **Thematic Layers** tab lists the thematic **Layers** to be imported with the related **Attributes**.
2. To add thematic layers click the **Add Layer** button.
3. To remove a thematic layer, select it and click the **Remove Layer** button.

Other Options

Geometry search string; Attribute search string

You can define or modify the search strings for the **Geometry** files containing the thematic layer information to be imported. Further you can define or modify the search strings for the **Attribute tables**. You need to do this only for special file formats; common shape file formats such as **.dbf** and **.csv** are found automatically. On the corresponding the drop-down list you can select:

- **{search-string}** to use the scene string as defined on the **Workspace** tab.
- **{***copy of the currently applied search string***}** will not be updated while the scene string changes. Can be used for further modifications by editing.
- **From file** to load a file name used to adapt the search string. We recommend using this option. The software automatically detects the proper string based on the information provided by the search string. If the software fails to do so, this is an indication that the search string may be incorrect.

Alias

You can define names for thematic layers and then address them later within other projects.

→ **About Layer Aliases** on page 54

Geometry Driver

Keep the default **<auto>** to select the appropriate file format **Driver** automatically. If you need a special driver, select it on the drop-down list.

Attribute ID Column

Generally, each attribute table contains a standard column ID containing the IDs for each object. In case you want to set another column as ID, enter it in the text box instead of the default which is **ID**.

Channel Index

By default, **All channels** are imported. If you want to load thematic layers which are image files, you may want to choose the image layer that contains the desired thematic information. Just enter the channel index value in the **Channel index** combo box to import the corresponding image layer, e.g. if you enter **2**, the second channel in the selected image file is imported.

Attribute table column alias list

In some geographic information systems (GIS) the names of attribute table columns can have only a limited number of characters. In order to display reasonable column names, an attribute table column alias list exists which maps the column aliases to the names.

You can modify attribute table column aliases:

1. Select **Add New Aliases** and click the ellipsis button. The **Enter Column Name** dialog box opens.


(ellipsis button)

2. Enter a name of an existing column name. A new item with this name is listed below the **Attribute table column alias list** group.
3. Select the new item and enter a new column alias.

Vector file parameters

Set parameters for the internal conversion of vector files to raster files:

- **Raster line width** for line shape files: You can edit the default of 1. Enter a number in pixels, which is used as the width with which a line shall be calculated with.
- **Raster point size** for point shape files: You can edit the default of 1 defining the pixel size in points.
- **Allow diagonal neighborhood** for line shape files: If your line shape file is converted into raster elements using 4-neighborhood or 8-neighborhood (diagonal) you can change the setting to **Yes**.

Metadata Import Options

On the **Metadata** tab of the **Customized Import** dialog box, you can load **Metadata** into the projects to be created and thus modify the import template with regard to the following options:

- Add and remove metadata.
- Define a special search string for metadata files.
- Select format driver to use for importing.
- Set conversions settings to include metadata in the feature tree.

Precondition: Define a master file on the **Workspace** tab. Otherwise, you cannot access the **Metadata** tab.

→ Perform a Customized Import on page 63

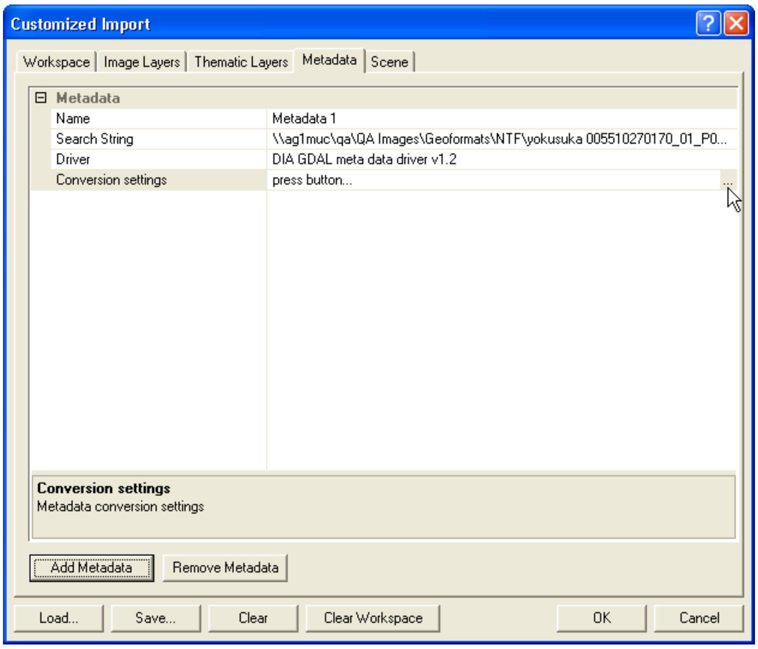


Figure 47: Metadata tab of the Customized Import dialog box.

1. The **Metadata** tab lists the metadata to be imported in groups.
2. To add a metadata group, click the **Add Metadata** button.
3. To remove a metadata group, select it and click the **Remove Metadata** button.

Other Options

Name

Edit the name of the metadata group.

Search String

Define the search string for the metadata files. In the corresponding drop-down list you can select:

- **{search-string}** to use the scene string as defined on the **Workspace** tab.
- **{***copy of the currently applied search string***}** will not be updated while the scene string changes. Can be used for further modifications by editing.
- **From file** to load a file name used to adapt the search string. We recommend using this option. The software automatically detects the proper string based on the information provided by the search string. If the software fails to do so, this is an indication that the search string may be incorrect.

Driver

If you know the file format driver applicable for your metadata, we recommend that you select it using the drop-down list. Otherwise, keep the default **<auto>** to automatically select an image format **Driver**.

Conversion Settings

When importing metadata you have to convert available external metadata to an internal metadata definition. Click the ellipsis button to open the **Metadata Conversion** dialog box.

(ellipsis button)

→ Convert Metadata to Provide it to the Feature Tree on page 52

Scenes Import Options

On the **Scene** tab of the **Customized Import** dialog box, you can configure image layers of scenes to be imported and thus modify the import template with regard to scene properties like geocoding, unit or site settings.

Precondition: Define a master file on the **Workspace** tab. Without a master file, you cannot access the **Metadata** tab.

→ Perform a Customized Import on page 63

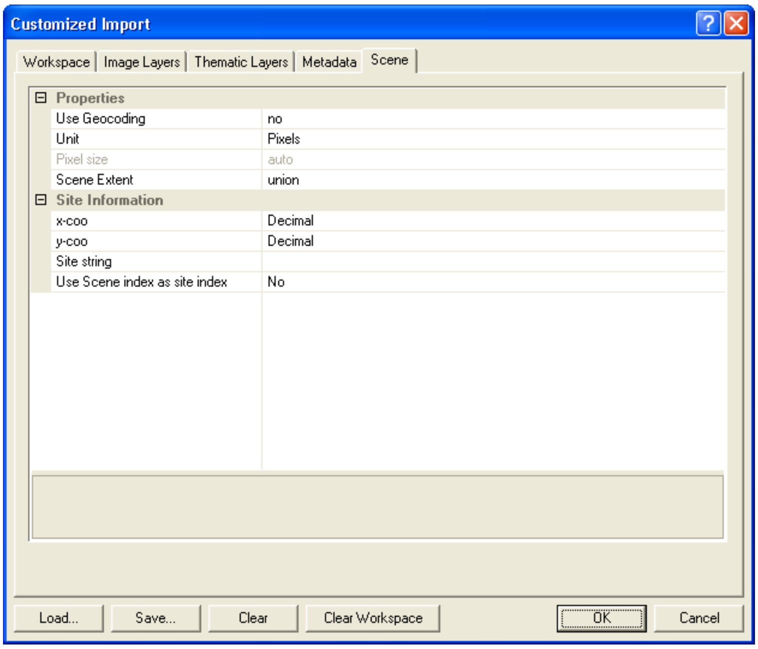


Figure 48: Scene tab of the Customized Import dialog box.

Properties

Use Geocoding

If your image data is geocoded you can make position coordinates available. In the drop-down list you can select from the following:

→ About Geocoding on page 50

- **No**: Geocoding information is not imported
- **Yes**: Geocoding information is imported
- **Automatic from file**: Automatic import depending on the existence of geocoding in the imported file.

Unit

For feature calculations, value display, and export, you may select another **Unit** on the corresponding drop-down list.

If you select **Auto**, the unit is applied according to the included geocoding information. If there is no geocoding information included the unit is pixel.

Pixel Size

For feature calculations, value display, and export, you can change the **Pixels size**. If you keep the default **Auto**, the unit conversion is applied according to unit of the coordinate system of the image data as follows:

- In case of included geocoding information, the pixel size is equal to the resolution.
- In other cases, pixel size is 1 unit of the coordinate system, for example **pixel**, **•m**, or **m**.

Scene Extent

Define the **Scene extent** if the loaded image layers have different sizes. On the drop-down list you can select from the following:

- **First scene only**: All images must have the same size.
- **Scene file**: The **Sample File** selected on the **Workspace** tab determines the extent of the scene. Image layers with larger sizes are cut off at the boundaries of the sample file. Use geocoding if you want to have the single image layers aligned correctly.
- **Union**: The scene has the maximal extent based on all input image layers.
- **Intersection**: The extent of the scene is determined by the overlapping area of all input image layers.

Available only for Definiens eCognition **Life** Server

Well Information for Plates

Well string

For the customized import of well, you can specify how well coordinates are parsed from well name. You can use **{x}** and **{y}** variables for specifying well coordinates or **{x0}** and **{y0)** variables for zero-based coordinates.

Examples:

Well Name	Well String	Using Regular Expressions
A - 2	**{x} - {y}**	**{x:1}{any:"[^A-Z0-9]*"}{y}**
c1r1	**c{x}r{y}**	
Zero-based indexing like **c0r2**	**c{x0}r{y0}**	

Site Information for Plates

For the customized import of sites, you have to add some site information to the search string.

x-coo; y-coo

Select if **Decimal** or **Hexadecimal** numbers are used for the X-coordinates or Y-coordinates.

Site String

Define the string for the site. Use **{x}** or **{y}** variables to identify the site index within scene name. Example: **001_003.tif** requires a site string **{x}_{y}.tif**

Use Scene Index as Site Index

Select **Yes** for automatic site recognition. Select **No** if you want to define the X-coordinates and Y-coordinates manually.

Examples of Customized Imports

To give you a practical idea of the principles of customized import, you can load prepared sample data for typical use cases.

Preparation for all use cases:

1. Go to the **Workspace** window and create a new workspace.
2. Choose **File > Customized Import** on the main menu bar.
3. The **Customized Import** dialog box opens.

You will browse to the sample data as described in the use cases. The sample data is stored in the **Examples** folder within the **Definiens Developer** installation directory. The default path is **C:\Program Files\Definiens Developer** *Version number***Examples \CustomizedImport**

Use Case 1: Single Layer Simple

Each scene consists of three image layer **.tif** files stored in one folder.

Figure 49: Use case 1: Folder structure (left) with contained files of the highlighted folder (right).

4. Click the **Clear** button before configuring a new import.
5. If there are scenes in the workspace, click the **Clear Workspace** button to remove them.
6. **Select** the **Root Folder** named **root folder usecase 1**. The default path is **C:\Program Files\Definiens Developer** *Version number***Examples \CustomizedImport\root folder usecase 1**. The **Root Folder** text box displays this path.
7. **Select** a **Master File**, for example **Image_A_Layer1.tif** within the subfolder **image A**. While the **Master File** text box displays the path below the root folder (**image A\Image_A_Layer1.tif**), the **Search String** text box displays a textual representation of the master file path used as a suggestion for the searching routine:
   {{root}\{any-folders}\{scene}.tif:reverse}.
8. Press the **Test** button to preview the naming result of the **Master File** based on the **Search String**.
9. In the **Preview** area the scene variable is found as **Image_A_Layer1** instead of **Image_A**. Because all static parts of the search string need to be identified, edit the **Search String** as follows:
   {{root}\{any-folders}\{scene}_Layer1.tif:reverse}.
10. Test again as described in step 8.
11. After the master file is defined, the scene needs to be set up. To do so, go to the **Image Layers** tab. Under **File 1** the settings of a master file are already predefined. Change here the search string to:
   For **Layer 1** use **{{root}\{any-folders}\{scene}_Layer1.tif:reverse}**.
   For **Layer 2** use **{{root}\{any-folders}\{scene}_Layer2.tif:reverse}**.
   For **Layer 3** use **{{root}\{any-folders}\{scene}_Layer3.tif:reverse}**.
   To define **Layer2**, click the **Add File** button to open the **Import Image Layer** dialog box. Navigate to **Image_A_Layer2.tif** and open it. Under **File 2** type **Layer 2** as **Alias**.
   For **Layer 3** proceed accordingly.

(ellipsis button)

12. Make sure that the **Create workspace folder from search string** check box is selected while the other is cleared.

13. Click **OK** to load the scenes **Image A** and **Image B** to the workspace. Examine the executed import in the **Workspace** window and open the created scenes. If not all image layers are displayed, check the **Image Layer Mixing**.

→ Edit the Image Layer Mixing on page 96

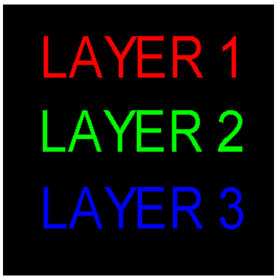


Figure 50: Imported sample scene.

If you are not satisfied with the import, reopen the **Customized Import** dialog box and go back to step 5.

Alternatively, you can load the sample import template to the **Customized Import** dialog box. Click the **Load** button and select the import template file named **Usecase 1.xml**. The path is **C:\Program Files\Definiens Developer** *Version number***Examples \CustomizedImport\Usecase 1.xml**.
Be sure to select the **Root Folder** (see step 6)and **Sample File** (see step 7) before you click **OK**.

Use Case 2: Single Layer Complex

Each scene consists of three image layer **.tif** files each stored in a different folder.

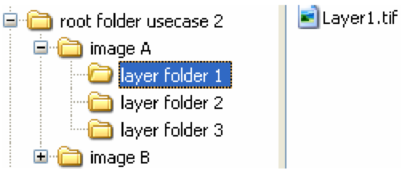


Figure 51: Use case 2: Folder structure (left) with file contained in the highlighted folder (right).

4. Click the **Clear** button before configuring a new import.

5. If there are scenes in the workspace, click the **Clear Workspace** button to remove them.

6. **Select** the **Root Folder** named **root folder usecase 2**. The default path is **C:\Program Files\Definiens Developer** *Version number***Examples \CustomizedImport\root folder usecase 2**. The **Root Folder** text box displays this path.

7. **Select** a **Master File**, for example **Layer1.tif** within the subfolder **layer folder 1**. While the **Sample File** text box displays the path below the root folder (**image A\layer folder 1\Layer1.tif**), the **Search String** text box displays **{{root}\{any-folders}\{scene}.tif:reverse}**.

8. Press the **Test** button to preview the naming result of the **Master File** based on the **Search String**.

9. The **Preview** is not correct because the scene is found as **Layer1** instead of **image_A**. To define the scene name and all static parts of the search string, you have to edit the **Search String** as follows: **{{root}\{scene}\layer folder 1\Layer1.tif:reverse}**.

10. After the master file is defined, the scene needs to be set up. To do so, go to the **Image Layers** tab. Under **File 1** the settings of a master file are already predefined. Change here the search string to:
For **Layer 1** use **{{root}\{scene}\layer folder 1\Layer1.tif:reverse}**.
For **Layer 2** use **{{root}\{scene}\layer folder 2\Layer2.tif:reverse}**.
For **Layer 3** use **{{root}\{scene}\layer folder 3\Layer3.tif:reverse}**.
Type **Layer 1** as **Alias**.
To define **Layer2**, click the **Add File** button to open the **Import Image Layer** dialog box. Navigate to **Image_A_Layer2.tif** and open it. Under **File 2** type **Layer 2** as **Alias**.
For **Layer 3** proceed accordingly.

(ellipsis button)

11. Make sure that the **Create workspace folder from search string** check box is selected while the other is cleared.

12. Click **OK** to load the scenes **Image A** and **Image B** to the workspace. Examine the executed import in the **Workspace** window and open the created scenes. If not all image layers are displayed, check the **Image Layer Mixing**.

→ Edit the Image Layer Mixing on page 96

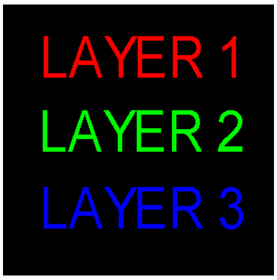


Figure 52: Imported sample scene.

If you are not satisfied with the import, reopen the **Customized Import** dialog box and go back to step 5.

Alternatively, you can load the sample import template to the **Customized Import** dialog box. Click the **Load** button and select the import template file named **Usecase 2.xml**. The path is **C:\Program Files\Definiens Developer** *Version number***\Examples \CustomizedImport\Usecase 2.xml**.
Be sure to select the **Root Folder** (see step 6)and **Master File** (see step 7) before you click **OK**.

Use Case 3: Multi-Layer and Layer Alias

Each scene consists of two **.tif** files: one with three image layers, one with a single layer both stored in one folder.

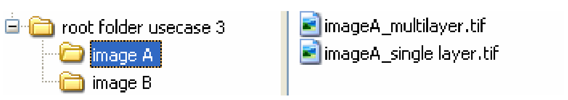


Figure 53: Use case 3: Folder structure (left) with file contained in the highlighted folder (right).

4. Click the **Clear** button before configuring a new import.

5. If there are scenes in the workspace, click the **Clear Workspace** button to remove them.

6. **Select** the **Root Folder** named **root folder usecase 3**. The default path is **C:\Program Files\Definiens Developer** *Version number***\Examples \CustomizedImport\root folder usecase 3**. The **Root Folder** text box displays this path.

7. **Select** a **Master File**, for example **imageA_multilayer.tif** within the subfolder **image A**. While the **Sample File** text box displays the path below the root folder (**image A\imageA_multilayer.tif**), the **Search String** text box displays **{{root}\{any-folders}\{scene}.tif:reverse}**.

8. Press the **Test** button to preview the naming result of the **Master File** based on the **Search String**.

9. The **Preview** is not yet satisfying because the scene is found as **imageA_multilayer** instead of **Image A**. You have to edit the **Search String** as follows: **{{root}\{any-folders}\{scene}_multilayer.tif:reverse}**

10. After the master file is defined, the scene needs to be set up. To do so, go to the **Image Layers** tab. Under **File 1** the settings of a master file are already predefined.

11. a.) If no image layer aliases are used set the layer properties as follows:

    Add to the predefined **File 1** another **File 2** by clicking the **Add File** button. For **File 2** change the search string as follows:

    - For **Layer 1** use **{{root}\{any-folders}\{scene}_multilayer.tif:reverse}**.
    - For **Layer 2** use **{{root}\{any-folders}\{scene}_single layer.tif:reverse}**.

    b.) If you want to use image layer aliases, set the layer properties as follows:

    Add to the predefined **File 1** another **File 2** by clicking the **Add File** button. For **File 2** change the search string as follows:

    - For **Layer 1** use **{{root}\{any-folders}\{scene}_multilayer.tif:reverse}**.
      Define the layer alias as **alias1**.
      Set the **Channel index** to **1**, defining that the first channel of this file is loaded.
    - For **Layer 2** use **{{root}\{any-folders}\{scene}_multilayer.tif:reverse}**.
      Define the layer alias as **alias2**.
      Set the **Channel index** to **2**, defining that the second channel of this file is loaded.
    - For **Layer 3** use **{{root}\{any-folders}\{scene}_multilayer.tif:reverse}**.
      Define the layer alias as **alias3**.
      Set the **Channel index** to **3**, defining that the third channel of this file is loaded.
    - For **Layer 4** use **{{root}\{any-folders}\{scene}_singlelayer.tif:reverse}**.
      Define the layer alias as **alias4**.
      Since this is a one layer file, you do not need to edit the channel index.

    Further change under the **File 1** group, the following **Alias** entries:

    - For the **Alias** of **Layer 1** type **alias1**.
    - For the **Alias** of **Layer 2** type **alias2**.
    - For the **Alias** of **Layer 3** type **alias3**.

    Under the **File 2** group, change the following **Alias** entries:

    - For the **Alias** of **Layer 1** type **alias4**.

12. Make sure that the **Create workspace folder from search string** check box is selected while the other is cleared.

13. Click **OK** to load the scenes **Image A** and **Image B** to the workspace. Check the executed import in the **Workspace** window and open the created scenes. Check the **Image Layer Mixing** to see if Layer 4 is loaded as well. If not, you can select

→ Edit the Image Layer Mixing on page 96

colors for **Layer 4** e.g. **R** and **G**.

Figure 54: Imported sample scene of Use Case 3.

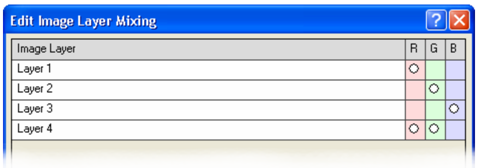


Figure 55: Sample setting in the Edit Image Layer Mixing dialog box for displaying all image layers.

If you are not satisfied with the import, reopen the **Customized Import** dialog box and go back to step 5.

Alternatively, you can load the sample import template to the **Customized Import** dialog box. Click the **Load** button and select the import template file named **Usecase 3 no alias.xml** or **Usecase 3 with alias.xml**. the path is **C:\Program Files\Definiens Developer** *Version number***Examples \CustomizedImport**
Be sure to select the **Root Folder** (see step 6)and **Sample File** (see step 7) before you click **OK**.

5.2.4 Change the View in the Workspace

The **Workspace** window provides different views of projects.

You can switch among the following:

- **List View**
- **Folder View**
- **Small Thumbnails**
- **Large Thumbnails**

You can also filter the columns displayed in the **List View** and the **Folder View.**

To switch views, click the drop-down arrow at bottom of the right pane of the **Workspace** window and select one of the options.

List View

In the **List View**, the right pane of the **Workspace** window lists all projects contained in the selected structure hierarchy of the tree view. This includes also projects contained in sub-structures. The position of a project within the structure is noted in the project name: Each path level is separated by a period. With the **List View** you have direct access to all projects in your workspace.

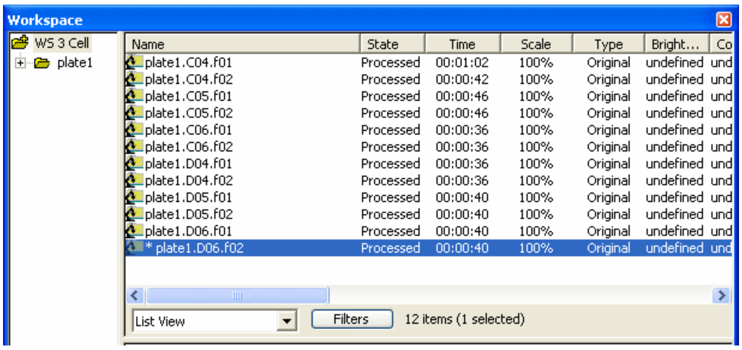


Figure 56: Workspace window with activated List View.

Folder View

In the **Folder View**, you only see the content of the selected structure, either folders or projects. Double-click a folder to see its contents. The **Folder View** displays structural elements to help you navigate.

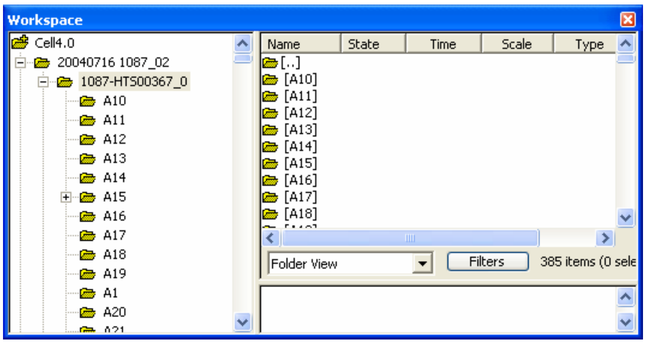


Figure 57: Workspace window with activated Folder View. (Image data courtesy of BioImage A/S).

Thumbnail View

Using the **Small** or **Large Thumbnails,** you can select any folder to see thumbnails of the contents. Double-click any thumbnail to open it in the project view. The open image is marked in the **Thumbnail** view with an asterisk (*) before the file name.

→ Edit Image Layer Mixing for Thumbnails on page 98

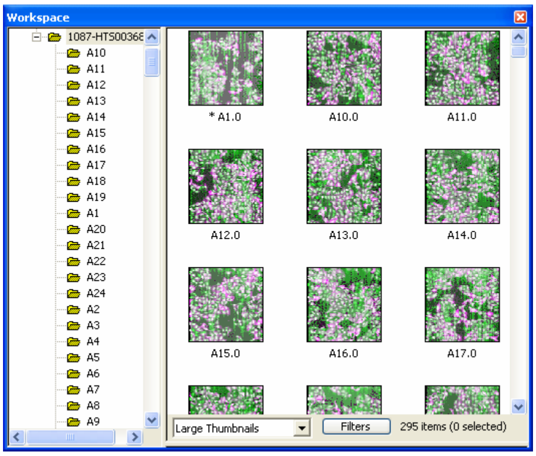
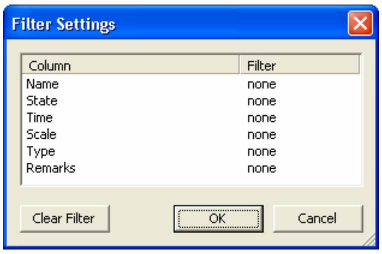


Figure 58: Large Thumbnails view in the Workspace window. (Image data courtesy of BioImage A/S).

Set Filters for the Workspace

Click the **Filters** button to open the **Filter Settings** window and create filters for the projects in the right pane of the workspace.



1. To create a filter, select the category you want to filter in the **Filter** column and type in the text for the projects you want to view. Click **OK.**
   The right pane displays a colored background and your filter is applied.
2. To remove a filter, select it in the **Filter** column and click the **Clear Filter** button.

5.2.5 Manage Folders in the Workspace Tree View

Add, move, and rename folders in the tree view on the left pane of the **Workspace** window. Depending on the import template, these folders may represent different items, such as **Assay**, **Well**, **Plate** or **Run** for a cell screening device.

1. To add an item, right-click a folder and select **Add [Item]**.
2. The new folder is displayed in the tree view of the **Workspace** window. You can edit the folder name. To rename a folder, right-click it and choose **Rename**.
3. Move folders to rearrange them by drag-and-drop operations.

5.2.6 Save a Workspace

Save the currently open workspace to a **.dpj** file.

To save a workspace, do one of the following:

- Choose **File** > **Save Workspace** on the main menu bar.
- Choose **File** > **Save Workspace As** on the main menu bar. The **Save Workspace** dialog box opens. Select a folder and enter a name for the workspace file (**.dpj**). Click the **Save** button to store the file.

Save Workspace

5.2.7 Close a Workspace

Close the currently open workspace.

1. To close a currently open workspace, do one of the following:
    - Choose **File** > **Close Workspace** on the main menu bar.
    - If available, create a new workspace.
    - Open an existing workspace.
2. If the workspace has changed, you are asked to save the modifications. Select **Yes** for saving and closing. Select **No** to close the workspace without saving.

→ Create a New Workspace on page 57

→ Open an Existing Workspace on page 84

5.2.8 Move a Workspace

Moving a workspace is easy because you can move the complete workspace folder and continue working with the workspace on the new location.

If file connections related to the input data are lost, the **Locate Image** dialog box opens. Browse to the new location and reconnect the image. This automatically updates all other input data files which are stored under the same input root folder. If you have loaded input data from multiple input root folders, you only have to relocate one file per input root folder to update all file connections.

Use the same procedure if input data has been moved.

We recommend that you not move any output files which are stored by default within the workspace folder. These are typically all **.dpr** project files and by default, all result files. However, if you do, you can modify the path of the output root folder under which all output files are stored.

→ Create a New Workspace on page 57

1. To modify the path of the output root folder choose **File** > **Workspace Properties** on the main menu bar. The **Workspace Properties** dialog box opens.

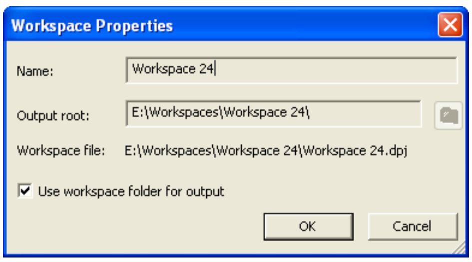


Figure 59: Workspace Properties dialog box.

2. Clear the **Use workspace folder** check box and change the path of the output root folder by editing it. Alternatively, you can click the ellipsis button and browse to an output root folder.

 (ellipsis button)

5.2.9 Open an Existing Workspace

Open an existing workspace from a file.

1. To open an existing workspace, do one of the following:

   - Click the **Open Workspace** button on the **File** toolbar.
   - Choose **File** > **Open Workspace** on the main menu bar.

   The **Open Workspace** dialog box opens.

 **Open Workspace**

2. Browse for an **Definiens** workspace file (**.dpj**).
3. Click **Open**. The folder tree of the workspace is displayed in the left pane (the tree view) of the **Workspace** window.

5.2.10 View a Project

Open a project to view and investigate it in the project view.

1. Go to the right pane of the **Workspace** window that lists all projects of a workspace.
2. To view a project, do one of the following:

   - Right-click a project and choose **Open** on the context menu.
   - Double-click a project.
   - Select a project and press Enter.

3. The project opens and is displayed in the project view. If another project is already open, it is closed before opening the other one.

For more information about opening and viewing projects, see the related instructional chapters.

→ **View Data in Projects** on page 93

→ **Open a Separate Project** on page 43

Open a Subset Only

If scenes are very large, you can open and investigate a subset of the project only.

1. Go to the right pane of the **Workspace** window that lists all projects of a workspace.
2. Right-click a project and choose **Open Subset**. The **Subset Selection** dialog box opens.
3. Define a subset and confirm with **OK**. The subset displays in the project view. This subset is not saved with the project and does not modify the project. After closing the project view of the subset, the subset is lost.

   However, you can save the subset as a separate project.

→ **Subset Selection** on page 461

→ **Save a Project** on page 55

5.2.11 Inspect the State of a Project

For control purposes you can view the state of the current version of a project.

1. Go to the right pane of the **Workspace** window that lists the projects.
2. Check the **State** of the current version of a project displayed beside its **Name**. The following states are possible:

→ Manage Projects in a Workspace on page 56

Processing States Related to User Workflow		
Created	Project has been created.	→ Create a New Project on page 44
Canceled	Automated analysis has been canceled by the user.	→ Cancel Image Analysis of Waiting and Currently Processing Projects on page 369
Edited	Project has been modified or manually edited by the user.	→ Modify a Project on page 54 → Manual Editing on page 391
Processed	Automated analysis has finished successfully.	→ Process Data on page 361
Skipped	Tile has not not selected randomly by the **submit scenes for analysis** algorithm with parameter **Percent of Tiles to Submit** defined smaller than 100.	→ Reference Book
Stitched	Stitching after processing has been successfully finished.	→ Stitch Tile Results on page 376
Accepted	Result has been marked by the user as accepted.	→ Accept and Reject Results on page 390
Rejected	Result has been marked by the user as rejected.	→ See above.
Deleted	Project was removed by the user. This state is visible in the **Project History**.	→ Inspect the State of a Project on page 85
Other Processing States		
Unavailable	A basic element of the software, the Job Scheduler, where the job was submitted is currently unavailable. It might be disconnected or restarted.	→ Start Automated Image Analysis on page 363
waiting	Project is waiting for automated analysis.	→ See above.
processing	Automated analysis is running.	→ Cancel Image Analysis of Waiting and Currently Processing Projects on page 369
FAILED	Automated analysis has failed. See **Remarks** column for details.	
TIMEOUT	Automated analysis could not be completed due to a timeout.	
CRASHED	Automated analysis has crashed and could not be completed.	

5.2.12 Inspect the History of a Project

Inspect detailed information for each version of a project. Inspecting older versions helps with testing and optimizing solutions. This is especially helpful when performing a complex analysis, where the user may need to locate and roll back to an earlier version.

1. To inspect the history of older project versions, go to the right pane of the **Workspace** window that lists projects. Right-click a project and choose **History** from the context menu. The **Project History** dialog box opens.

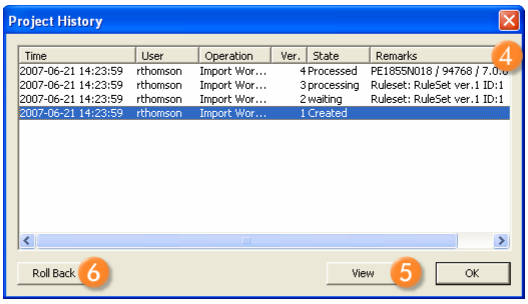


Figure 60: Project History dialog box.

2. All project versions (**Ver.**) are listed with related **Time**, **User**, **Operations**, **State**, and **Remarks**.

→ Inspect the State of a Project on page 85

3. Click **OK** to close the dialog box.

Options

4 Click a column header to change sorting by column.

5 To open a project version in the **Project View**, do one of the following:

- Select a project version and click **View**.
- Double-click a project version.

5.2.12.1 Rollback a Single Project

6 To restore an older version, choose the version you want to bring back and click the **Rollback** button of the **Project History** dialog box. The restored project version does not replace the current version but adds it to the project version list. The intermediate versions are not lost.

5.2.13 Rollback All Changes of Projects

Rollback all changes to restore one or multiple projects to the first version.

Beside the **Rollback** button of the **Project History** dialog box, you can proceed as follows:

→ Inspect the History of a Project on page 85

1. Do one of the following:

   - Select a project in the right pane of the **Workspace** window and select **Analysis** > **Rollback All** on the main menu bar.
   - Right-click a folder in the left pane of the **Workspace** window and select **Rollback All** on the context menu. Alternatively, you can select **Analysis** > **Rollback All** on the main menu bar.

   The **Rollback All Changes** dialog box opens.

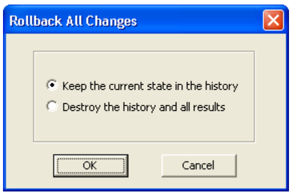


Figure 61: Rollback All Changes dialog box.

2. Select **Keep the current state in the history** if you want to keep the history when going back to the first version of the projects. The intermediate versions are not lost.
   Select **Destroy the history and all results** if you want to restart with a new version history after removing all intermediate versions including the results. In the **Project History** dialog box, the new version 1 displays **Rollback** in the **Operations** column.

Find Out More

Automatic Rollback

*In cases of some unexpected failure of processing, the project automatically rolls back to the last workflow state. This operation is documented as **Automatic rollback** in the **Remarks** column of the **Workspace** window and as **Roll Back** operation the **History** dialog box.*

5.2.14 Import Existing Project to Workspace

Import an existing project into a workspace. The project might be unprocessed or processed before import.

For creating a new project within a workspace, use the **Add Project** command.

→ Create a New Project Within a Workspace on page 87

1. To import an existing project into a workspace:
   - Go to the left pane of the **Workspace** window and select a folder. Right-click it and choose **Import Existing Project** from the context menu.
   - Choose **File > New Project** on the main menu bar.

   The **Open Project** dialog box opens.
2. Select one project (file extension **.dpr**) and click **Open**.
3. The new project is displayed in the right pane of the **Workspace**.

5.2.15 Create a New Project Within a Workspace

Create a new project within a workspace by importing a scene.

For adding multiple projects to a workspace, use the **Import Scenes** command.

→ Import Scenes to a Workspace on page 58

For adding an existing projects to a workspace, use the **Import Exiting Project** command.

→ Import Existing Project to Workspace on page 87

For creating a new project separately from a workspace, close the workspace and use the **New Project** command.

→ Create a New Project on page 44

1. To create a new project within a workspace, do one of the following:

   - Go to the left pane of the **Workspace** window. Right-click a folder and, if available, choose **Add Project** from the context menu.

   - Choose **File > New Project** on the main menu bar.

   The **Import Image Layers** dialog box opens.

Create New Project

2. Select image data.

   To open some specific file formats or structures, you have to proceed as follows:

   2.1. First select the correct driver in the **Files of type** drop-down list box.

→ Supported Image Formats on page 47

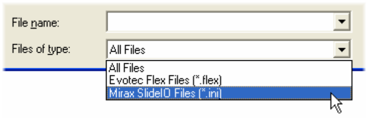


Figure 62: Select a Driver.

   2.2. Then select from the main file in the files area.

   If you select a repository file (archive file), another **Import Image Layers** dialog box opens in which you can select from the contained files.

→ Import Image Layers on page 448

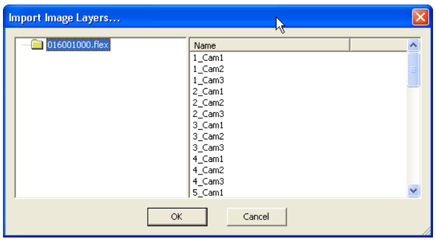


Figure 63: Import Image Layers dialog box listing items of a repository file.

   Press **Open** to display the **Create Project** dialog box.

3. All preloaded image layers will be displayed along with their properties in the **Create Project** dialog box. Now you can select a subset or modify the project default settings.

→ Create a New Project on page 44

4. Click **OK** to create a project. The new project is displayed in the right pane of the **Workspace**.

5.2.16 Work on Subsets and Copies of Scenes

If you have to analyze projects with scenes that exceed the processing limitations, you have to consider some preparations.

→ About Processing Limitations on page 370

Projects with scenes within the processing limitations can be processed normally, anyhow some preparation is recommended for accelerating the image analysis or if the system is running out of memory.

The following methods are available for manual and automated image analysis all of them working on some kind of scene subsets:

- Definition of a scene subset

→ Define a Scene Subset on page 371

- Working at different scales

→ Work at Different Scales on page 373

- Tiling and stitching of large scenes
- Tiling of large scenes

For automated image analysis we recommend developing rule sets that handle the above methods automatically. In the context of workspace automation, subroutines enable you to automate and accelerate the processing specially of large images.

→ Tile and Stitch Large Scenes on page 375
→ Tile Large Scenes on page 377
→ Automate the Workspace Processing on page 295

5.2.17 Remove Projects and Delete Folders

Remove selected projects from a workspace.

When removing projects, the related image data is not deleted as it is referenced by the projects.

1. For deleting, select one or more projects in the right pane of the **Workspace** window. Do one of the following:
   - Press Del on the keyboard.
   - Right-click the project item and choose **Remove** on the context menu.
   - Delete folders including all contained projects: Right-click a folder in the left pane of the **Workspace** window and choose **Delete** on the context menu.
2. Confirm the deletion. The project item disappears from the right pane of the **Workspace** window.

Tip

Undo Removing of Projects

*If you removed a project by mistake, just close the workspace **without** saving. After reopening the workspace, the deleted projects are restored to the last saved version.*

5.2.18 Copy Workspace Window

Copy the current display of both panes of the **Workspace** to clipboard. You can then paste it as a picture into documents.

1. Go to the left or right pane of the **Workspace**. Right-click a folder and choose **Copy to Clipboard** from the context menu.
2. Go to an office or imaging application and paste the copied display into a document.

5.2.19 Save Dialog Box List to File

Save the currently displayed list in the right pane of the **Workspace** window to a **.csv** file.

1. Go to the right pane of the **Workspace** window. Right-click a project and choose **Save list to file** from the context menu.

2. The list can be opened and analyzed in applications like **Microsoft® Excel**.

5.3 Manage Data in the Plate View

Available only for Definiens eCognition **Life** Server

For automated image analysis in cellular biology, you commonly use plates. Definiens Developer provides specific controls and tools to develop ruleware for cellular image analysis.

For easy-to-use image analysis, we recommend to use the **Definiens Architect** in combination with **Definiens Cellenger** application.

To open the **Plate View** window, choose **View > Windows > Plate View** on the main menu bar.

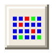


Plate View

5.3.1 Navigate Through a Plate

Navigate through a plate and specify wells to view images and layers of images or select them for analysis.

Available only for Definiens eCognition **Life** Server

In the **Navigation** tab, the plate is displayed in gray with all active wells circled. When you select one or more wells, the selected wells appear light gray, the rest of the plate is shaded. Inactive wells always are marked dark gray.

While holding the mouse pointer over one well, its coordinates and - if already processed - the value for the selected parameter are displayed in the **Selected wells** area. Otherwise, this area displays the number of selected wells.

Note
You cannot select wells with missing data.

To open the **Navigation** tab in the **Plate View** click **Navigate**.

The **Navigation** tab is the default tab of the **Plate View**.

Navigation tab

5.3.1.1 Select a single well

Select a single well to view the corresponding images and layers.

Available only for Definiens eCognition **Life** Server

1. To select a single well, click the well in the plate. The project of the selected well is displayed in the **Project View** window. The corresponding entry in the right **Workspace** pane is also selected.
2. To change the plate, select the plate number or name from the **Plate** list.
3. To browse through different fields of the selected image, click the up or down arrow of the **Site** list or enter the number of the desired image directly. The image in the **Project View** window changes as you browse through the layers. When selecting **All**, you can select from the **Result Statistics** list, what to display: Mean value, minimal or maximal value or the sum.
4. To deselect all selected wells, click on the plate between the wells.

5.3.1.2 Select Multiple Wells

Select multiple wells to analyze the corresponding scenes.

Available only for Definiens eCognition **Life** Server

1. To select multiple wells, do one of the following:
   - To select a row, click the row header character on the left side of the plate
   - To select a column, click the column header number on top of the plate.
   - To select all wells, click in the upper left corner of the plate.
   - To select a number of adjacent wells, drag over them. Each well you touch with the mouse will be selected.

The corresponding entries in the right **Workspace** pane are also selected. The **Wells selected** area shows the number of selected wells.

2. To change the plate, select the plate number or name from the **Plate** list.
3. To change the selected layer, click the up or down arrow of the **Site** list or enter the number of the desired layer directly. Select **All** to select all layers of a scene.
4. To deselect all selected wells, click on the plate between the wells.

Changing the selection of plates changes the display of the **Graph** simultaneously.

→ View Result Values in the Graph on page 384

5.3.2 Define Plate Layout

Define the wells used for negative and positive control or inactive wells, which will be excluded from the analysis as well as the results.

Available only for Definiens eCognition **Life** Server

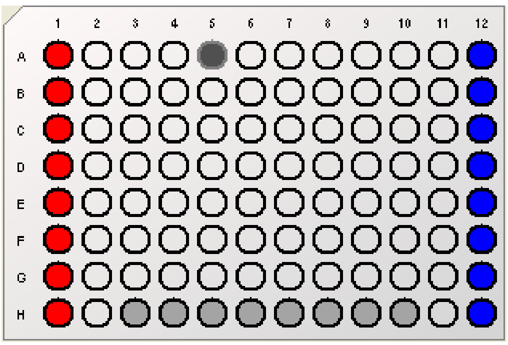


Figure 64: Plate map with marked wells.

1. To edit the layout of a plate, click **Layout** tab in the **Plate View**.
2. To appoint wells to be negative or positive control or set them inactive, select the desired wells.

Layout tab

3. Mark the wells as desired.
   - Click **Negative Control** or **Positive Control** to mark the selected wells respectively. Negative Control wells are displayed in red color, Positive control wells are blue.
   - Click **Inactive** to exclude the selected wells from the analysis. Inactive wells are displayed in gray color.
4. To remove marks from a well, select them and click **Clear marks**.

5.3.2.1 Save Plate Layout

You can save the layout of a plate to a **.dpl** file.

Available only for
Definiens eCognition **Life** Server

1. In the **Plate View**, define the layout as you whish.
2. Click **Save** to open the **Save plate layout** dialog box.
3. Browse to the desired folder and enter a name for the layout and confirm with **Save**.

5.3.2.2 Load Plate Layout

You can load an existing layout of a plate and also the input of compounds for the dose response curve. Plate layouts are saved as **.dpl** files.

Available only for
Definiens eCognition **Life** Server

→ **Input Compounds to Plate Rows** on page 385

1. In the Plate View, click **Load** in the **Layout** tab to open the **Open plate layout** dialog box.
2. Browse to the desired folder and select a layout file and confirm with **Open**.

6 View Data in Projects

Contents in This Chapter

You have several options for viewing and investigating your **Definiens** projects visually. Some of the visualization tools are dependent on the extent of analysis that has been performed.

→ View a Project on page 84
→ Use Workflow Views on page 40

1. You can explore individual image layers right from the start, after loading scenes, and throughout the image analysis, including the investigation of results. You can define the color composition for the display of image layers and set equalizing options.

→ Change Display of Image Layers on page 95

2. You can use different zoom and pan functions at any stage of image analysis.

→ Zoom Within Projects on page 101

3. After a basic segmentation that creates the first image objects, you can view their borders, called outlines.

→ View Outlines of Image Objects on page 108

4. Next, you classify image objects using various methods. You will commonly review the classification results extensively.

→ Change Display of Image Objects on page 109

   A specific visualization of features (image object attributes) enables you to preview the relevance of comprised information.

→ Feature View on page 111

5. For refinement of your image analysis, you can navigate visually through the different (if there is more than one) levels of the image object hierarchy. This enables you to to maintain information about the different image object levels and to perform more classifications.

→ Navigate Within Projects on page 114

6. Additionally, you can use several views of a project at the same time.

→ Use Different Viewing Modes on page 116

If you work with polygons you can display polygons and skeletons.

→ View Polygons on page 112
→ View Skeletons on page 114

6.1 Open a Project from a Workspace

Open a project to view and investigate it in the project view.

1. Go to the right pane of the **Workspace** window that lists all projects of a workspace.
2. To view a project, do one of the following:
   - Right-click a project and choose **Open** on the context menu.
   - Double-click a project.
   - Select a project and press Enter.
3. The project opens and is displayed in the project view. If another project is already open, it is closed before opening the other one.

For more information about opening and viewing projects, see the related instructional chapters.

→ View Data in Projects on page 93
→ Open a Separate Project on page 43

Open a Subset Only

If scenes are very large, you can open and investigate a subset of the project only.

1. Go to the right pane of the **Workspace** window that lists all projects of a workspace.
2. Right-click a project and choose **Open Subset**. The **Subset Selection** dialog box opens.
3. Define a subset and confirm with **OK**. The subset displays in the project view. This subset is not saved with the project and does not modify the project. After closing the project view of the subset, the subset is lost.

   However, you can save the subset as a separate project.

→ Subset Selection on page 461
→ Save a Project on page 55

6.2 Open a Separate Project

A separate project is a project that is not part of a workspace. You can open a separate project and investigate it in the project view.

Precondition: You need a **.dpr** file containing project. To view results, you need a **.dpr** file containing an analyzed project.

1. To view a project that is not part of a workspace, you must open it from a separate **.dpr** file. Choose **File** > **Open Project** on the main menu bar. The **Open Project** dialog box opens.
2. Select a project stored as **.dpr** file.
3. Click **open** and the project will display in the project view.


Open Project

If you try to open a second project in one **Definiens Developer** instance, the other one closes automatically. To open and view more than one project simultaneously, you have to start multiple **Definiens Developer** instances of Definiens Developer; each for opening another project.

→ Close a Project on page 56
→ Start Definiens Software on page 37

> **Note**
>
> If an existing project or its image data have been moved, the references to the image data may have got lost. If the **Locate** *image file* dialog box opens, the image data need to be relocated.

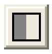
6.3 Change Display of Image Layers

Go to the **View Settings** toolbar and set display options for the display of your image.

As an alternative to the toolbar buttons you can use the **View Settings** window.

→ View Settings on page 463

6.3.1 Single Layer Grayscale


Single Layer Grayscale

Images are automatically assigned RGB (Red + Green + Blue) colors by default when an image with three or more image layers is loaded. Use the **Single Layer Grayscale** button on the **View Settings** toolbar to display the image layers separately in grayscale. In general, when viewing multilayered images, the grayscale mode for image display provides valuable information. To change from default RGB mode to grayscale mode go to the toolbar and press the **Single Layer Grayscale** button. Now you see image layer 1 only in the grayscale mode.

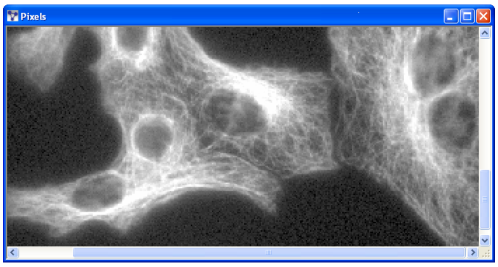
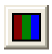


Figure 65: Gray scale project view of sample image (image data courtesy of EMBL Heidelberg). Layer 1 single gray scale view of microtubules.

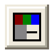
6.3.2 Three Layers RGB


Mix Three Layers RGB

Display three layers to see your image in RGB (Red+ Green + Blue) mode. By default, layer 1 is assigned to the Red channel, level 2 to Green, and level 3 to Blue. These are additively mixed to display the image in the **Project View**. To change these settings, go to the **Edit Image Layer Mixing** dialog.

> **Note**
>
> The color of an image area informs the viewer about the particular image layer, but not about the real color of the image area.

6.3.3 Show Previous Image Layer



Show Previous Image Layer

In **Grayscale** mode, this button displays the previous image layer. The number or name of the displayed image layer is indicated in the middle of the status bar at the bottom of the main window.
In **Three Layer Mix**, the color composition for the image layers changes one image layer up for each image layer. For example, if layers 2, 3 and 4 are displayed, the **Show Previous Image Layer Button** changes the display to layers 1, 2, and 3. If the first image layer is reached, the previous image layer starts again with the last image layer.

6.3.4 Show Next Image Layer

Show Next Image Layer

In **Grayscale** mode, this button displays the next image layer down.
In **Three Layer Mix**, the color composition for the image layers changes one image layer down for each layer. For example, if layers 2, 3, and 4 are displayed, the **Show Next Image Layer Button** changes the display to layers 3, 4 and 5. If the last image layer is reached, the next image layer begins again with image layer 1.

6.3.5 Edit the Image Layer Mixing

You can define the color composition for the visualization of image layers for display in the project view. In addition, you can choose from different equalizing options.

You can use different visualizations without changing the image data. Thus, the color of an image area informs the viewer about the particular image layer, but not about the real coloring of the image area.

> **Note**
>
> Changing the image layer mixing only changes the visual display of the image but not the underlying image data as such—it has no impact on the process of image analysis. Therefore, using this option enables you to better visualize the image according to your preferences without actually changing it.

When creating a new project, the first three image layers are displayed in red, green and blue.

1. To change the layer mixing, open the **Edit Image Layer Mixing** dialog box:

Edit Image Layer Mixing

   - Choose **View > Image Layer Mixing** on the main menu bar.
   - Double-click in the right pane of the **View Settings** window.

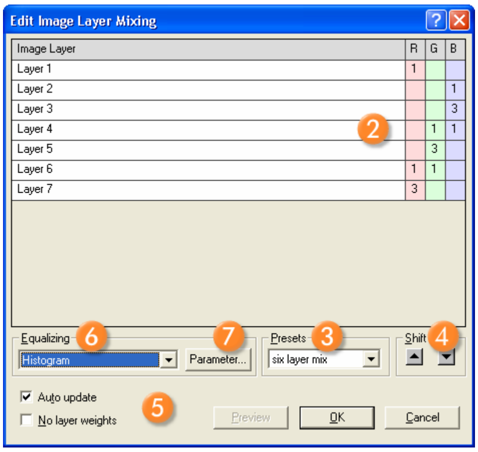


Figure 66: Edit Image Layer Mixing dialog box. Changing the layer mixing and equalizing options affects the display of the image only.

2. Define the display color of each image layer. For each **Image Layer** 2 you can set the weighting of the Red, Green and Blue color channel. Your choices can be displayed together as additive colors in the project view.

3. Choose among different layer mixes from several **Presets** 3. The **One layer gray** preset displays a layer in grayscale mode with the Red, Green, and Blue together. The **Three layer mix** displays layer 1 in the Red channel, layer 2 in Green and layer 3 in Blue. Choose **Six layer mix** to display additional layers.

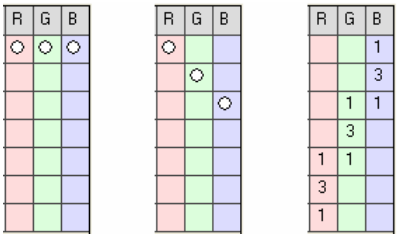


Figure 67: Layer Mixing presets (from left to right): One layer gray, Three layer mix, Six layer mix.

4. Change these settings to your preferred options with the **Shift** button 4 or by clicking in the respective R, G, B cell. One layer can be displayed in more than one color, and more than one layer can be displayed in the same color.

5. Individual weights can be assigned to each layer. Clear the **No layer weights** check box 5 and click the color in which you want the layer to be displayed. Left-clicking increases the layer's color weight while right-clicking decreases it.
The **Auto update** check box 5 refreshes the view with each change of the layer mixing settings. Clear this check box to show the new settings after clicking **OK**. With the **Auto Update** check box cleared, the **Preview** button becomes active.

6. Compare the available image **Equalization** methods 6 and choose one that gives you the best visualization of the objects of your interest.

→ About Image Equalization on page 98

Equalization settings are stored in the workspace and applied to all projects within the workspace. In the **Options** dialog box you can define a default equalization setting.

→ Options on page 452

7. Click the **Parameter** button 7 to changing the equalizing parameters, if available.

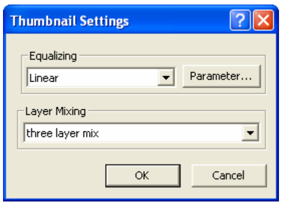
6.3.5.1 Edit Image Layer Mixing for Thumbnails

Change the way thumbnails display in the **Heat Map** window and in the **Thumbnail Views** of the workspace.

→ View Thumbnails in the Heat Map Window on page 380

To edit image layer mixing for thumbnails:

1. Right click on the **Heat Map** window or go to **View > Thumbnail Settings** to open the **Thumbnail Settings** dialog box.



2. Choose among different layer mixes in the **Layer Mixing** drop-down list.
   The **One layer gray** preset displays a layer in grayscale mode with the Red, Green, and Blue together. The **Three layer mix** displays layer 1 in the Red channel, layer 2 in Green and layer 3 in Blue. Choose **Six layer mix** to display additional layers.
3. Open the **Equalizing** drop-down box and select a method that gives you the best display of the objects in the thumbnails.

→ About Image Equalization on page 98

4. If you select an equalization method you can also click the **Parameter** button to changing the equalizing parameters.

6.3.5.2 About Image Equalization

Image equalization is performed after all image layers are mixed into a raw RGB (Red, Green, Blue) image. If, as is usual, only one image layer is assigned to each color, this approach is the same as applying equalization to the individual raw layer gray value images. On the other hand, if more than one image layer is assigned to one screen color (Red, Green or Blue), image equalization leads to higher quality results if it is performed after all image layers are mixed into a raw RGB image.

→ For details see the **Reference Book**

There are different modes for image equalization available:

None

No (**None**) equalization allows you to see the image data as it is what can be helpful at the beginning of rule set development when looking for an approach.

The output from the image layer mixing is displayed without further modification.

Linear Equalization

Linear equalization with **1.00%** is the default for new scenes. Commonly it displays images with a higher contrast as without image equalization.

Standard Deviation Equalization

With its default parameter of 3.0, **Standard deviation** renders a similar display as **Linear** equalization. Use a parameters around 1.0 for an exclusion of dark and bright outliers.

Gamma Correction Equalization

Gamma correction equalization is used to improve the contrast of dark or bright areas by spreading the corresponding gray values.

Histogram Equalization

Histogram equalization is well suited for **LANDSAT** images but can lead to substantial over-stretching on many normal images. It can be helpful in cases where you want to display dark areas with more contrast.

Manual Image Layer Equalization

Manual image layer equalization allows you to control equalization in detail. For each image layer individually, you can set the equalization method specifying the mapping function. In addition, you can define the input range by setting minimum and maximum values.

→ Manual Image Layer Equalization on page 100

Examples

Compare the following displays of the same scene:

Figure 68: Left: Three layer mix (red, green, blue) with Gamma correction (0.50). Right: One layer mix with Linear equalizing (1.00%)

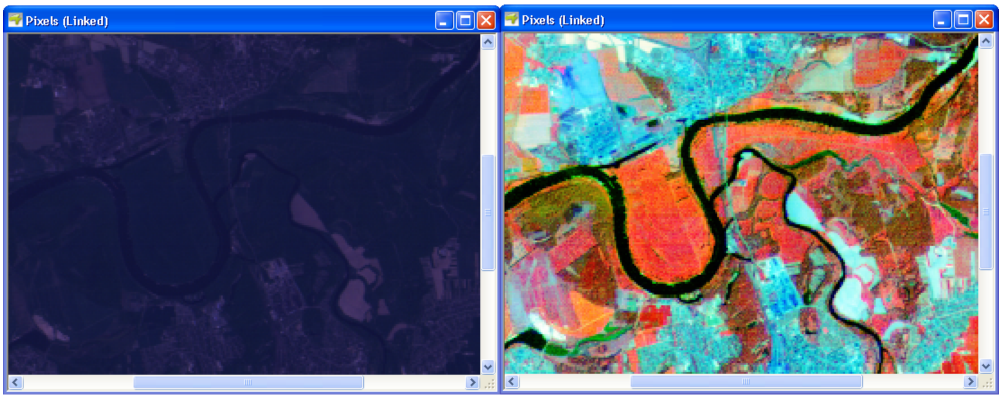


Figure 69: Left: Three layer mix (red, green, blue) without equalizing. Right: Six layer mix with Histogram equalization.
(Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.3.5.3 Manual Image Layer Equalization

Manual image layer equalization allows you to control equalization of image layer display in detail. You can select the method of equalization and limit it to a range of gray values that is important for the display of the objects in which you are interested.

1. Go to the **Equalizing** drop-down list box of the **Edit Image Layer Mixing** dialog box.

→ Edit the Image Layer Mixing on page 96

2. Select **Manual** and click the **Parameter** button. The **Image Layer Equalization** dialog box opens.

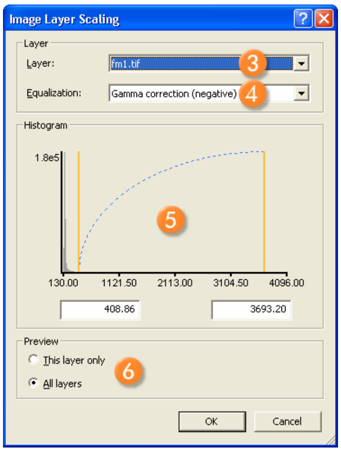


Figure 70: Image Layer Equalization dialog box.

3. Select one image **Layer** 3 for equalization. Only those image layers that you chose (with a dot or layer weight value) in the **Edit Image Layer Mixing** dialog box are listed.
4. Select an **Equalizing** 4 method from the drop-down list box:
   - **Linear**: Use to improve the contrast evenly over the selected range of gray values.
   - **Linear - inverse**: Same as **Linear** but with inverted gray values. This can be used, for example, in radiology to display a negative image positively, in other words, changing black to white and vice versa.
   - **Gamma correction (negative)**: Use to improve the contrast in brighter areas.
   - **Gamma correction (positive)**: Use to improve the contrast in darker areas.
   - **Gamma correction (negative) - inverse**: Same as **Gamma correction (negative)** but with inverted gray values. This can be used, for example, in radiology for displaying a negative image positively, in other words, changing black to white and vice versa.
   - **Gamma correction (positive) - inverse**: Same as **Gamma correction (positive)** but with inverted gray values. This can be used, for example, in radiology for displaying a negative image positively, in other words, changing black to white and vice versa.
5. The **Histogram** 5 displays the equalization method as function curve. You can modify minimum and maximum values in the histogram. Commonly you select a range of gray values that is important for the display of the objects of your interest. This allows you to limit the display on a defined range of gray values that is used for the selected equalization method 4. Likewise, you may exclude the display of disturbing gray values.

→ Check Image Layer Histograms on page 101

- Drag the orange marker.
- Enter a value in the text box.
- Click in the **Histogram** display and use arrow keys on your keyboard:
  ← and → to move the minimum value
  ↓ and ↑ to move the maximum value

6. The effect of the settings is updated as a preview in the project view. For **Preview** 6, you can switch between the display of all image layers or the selected one only.
7. Confirm with **OK** and go back to the **Edit Image Layer Mixing** dialog box.

6.3.6 Check Image Layer Histograms

The **Image Layer Histograms** dialog box provides information about the distribution of gray values on single image layers.

Depending on the image data, the **Histogram** uses different ranges:

Data Type	Lowest Scale Value	Highest Scale Value
8-bit	0	256
16-bit unsigned; 32-bit unsigned	0	2^n
16-bit signed; 32-bit signed	-2^n	2^n
32-bit float	Calculated by rounding the first relevant digit to the higher value by 1. Examples: 0.0000234 rounds to min=0.00002, max=0.0003 1964 rounds to min=1000, max=2000 –1964 rounds to min=–2000, max=–1000	

1. To open, choose **Tools > Image Layer Histograms** on the main menu bar.

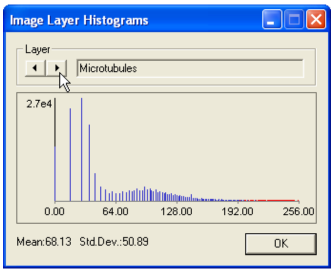


Figure 71: Image Layer Histograms dialog box.

2. Switch between the image layers by clicking the arrow buttons in the **Layer** group box.

6.4 Zoom Within Projects

To navigate within the display of a project, you can use the various zoom and pan functionalities.

To navigate the project display in the project view, do one of the following:

- Select a button from the **Zoom** toolbar:
- Click in the project view and use the mouse wheel.
- Right-click in the project view and select a specific cursors (pointers) from the context menu.
- On the main menu bar, choose **View > Cursor Mode** to use specific cursors (pointers).
- On the main menu bar, choose **View > Display Mode** to use direct zooming commands.

→ Zoom Toolbar on page 426

Figure 72: Zoom toolbar.

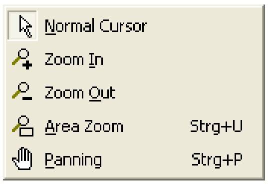


Figure 73: Context menu with zoom options.

Examples of different zoom settings:

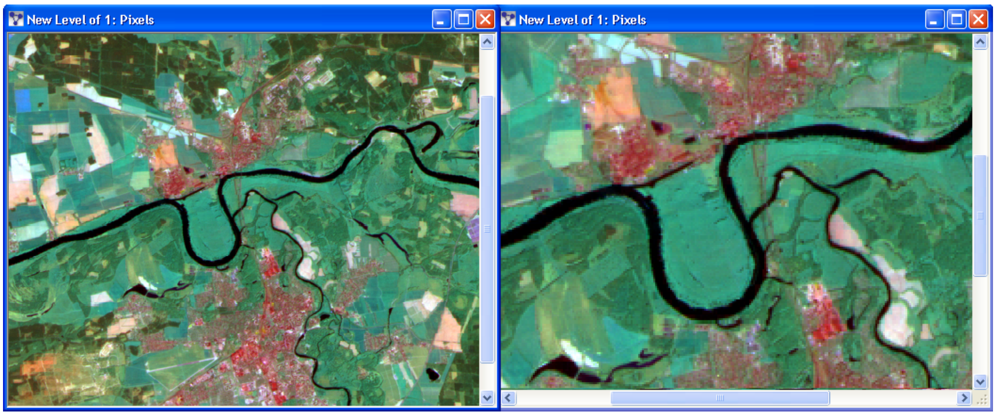


Figure 74: Left: Complete image Zoomed to Window. Right: After Zooming In resulting in zoom factor 100%.

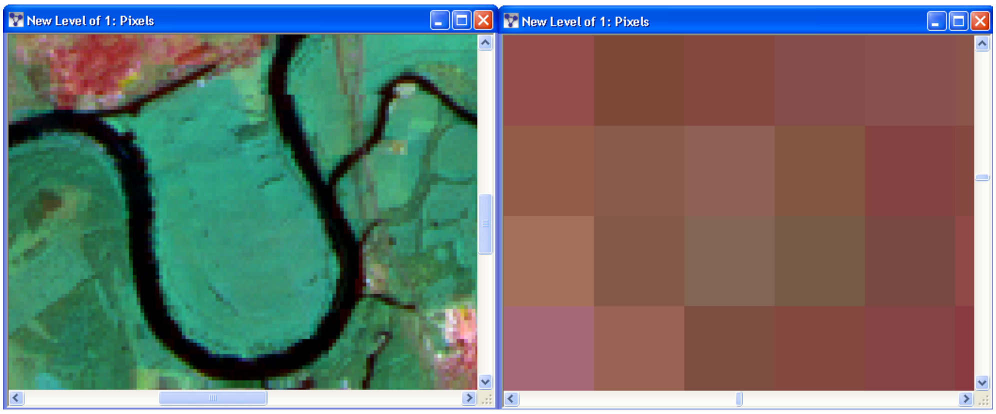


Figure 75: Left: After dragging an area with the Area Zoom. Right: Maximum zoom factor of 9600%.
(Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany)

6.4.1 Normal Cursor

The **Normal Cursor** is selected by default. Switch back from a zoom or pan mode to the **Normal Cursor**.


Normal Cursor

6.4.2 Zoom Using the Mouse Wheel

If your mouse has a scroll wheel, use it for zooming.

1. Click in a **Project View** window.
2. Scroll the mouse wheel to the front to zoom in.
3. Scroll the mouse wheel to the back to zoom out.

6.4.3 Area Zoom

Select the zoom area in the project view.

1. To activate the **Area Zoom** choose **View > Curser Mode > Area Zoom** on the main menu bar. The cursor changes its shape to magnifier glass.


Area Zoom

2. Drag a rectangle on the project view to zoom into the marked area.


Area Zoom curser shape

3. To deactivate the **Area Zoom** selection mode choose **View > Curser Mode > Normal Cursor**.


Normal Cursor

Figure 76: Area Zoom (highlighted in white). (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.4.4 Zoom In Center

Select the **Zoom In Center** button to zoom into the image from the middle of the displayed image.


Zoom In Center

6.4.5 Zoom In

Select the **Zoom In** button and click in the image to enlarge your image from a point. This means you can enlarge the specific area of interest.


Zoom In

6.4.6 Zoom Out Center

Select the **Zoom Out Center** button to zoom out of the image from the middle of the displayed image.


Zoom Out Center

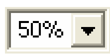
6.4.7 Zoom Out

Select **Zoom Out** and click into the image to zoom out from a point. This means you can specifically zoom out of the region of interest.


Zoom Out

6.4.8 Select Zoom Value

Select or enter a zoom value to change the display in the project view.



Zoom

6.4.8.1 Zoom to a Magnification Zoom Value

Zoom the display in the project view to different magnification zoom values representing zoom factors. Magnification zoom values are denoted like magnification factors used in microscopy.

Precondition: In the **Options** dialog box, the **Display scale** type is set on **Magnification**. Alternatively, the type can be set to **Auto** where using adequate data, such as is used in microscopy.

→ Options on page 452

1. Select a **Zoom** button. The display in the project view changes.

Zoom

2. Enter other zoom factors to the **Zoom** combo box.

Zoom

Find Out More

Zooming when Using the Magnification Scaling Mode

The magnification zoom value is denoted in a way similar to magnification factors. Note, that they do not represent the scale factor (referred to as magnification/resolution) of the displayed scene. That means, when changing the ***Zoom*** *value, the scale is not affected and the scene is not resampled.*

However, if you compare differently scaled copies of the same scene, you need different ***Zoom*** *factors to display them at the same size.*

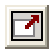
A zoom factor of 100% displays the project at its current scale (magnification/resolution).

→ ***About Scales*** *on page 49*

6.4.9 Zoom to Window

View the image in the same extent as the project view.

To zoom the whole image into the project view, do one of the following:



Zoom to Window

- Click the **Zoom to Window** button on the **Zoom** toolbar.
- Choose **View** > **Display Mode** > **Zoom to Window** on the main menu.

6.4.10 Zoom 100%

Click the **Zoom 100%** button to display the image at its original scale.

Zoom 100%

6.4.11 Panning

For moving within an image, use the **Panning** cursor. Alternatively you can use the **Pan Window**.

→ **Pan Window** on page 106

Panning

1. To activate the panning mode do one of the following:
   - Right-click in the project view and select **Panning** on the context menu.
   - Selecting **View** > **Cursor Mode** > **Panning** from the main menu.

   The cursor changes its shape.

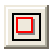
Panning curser shape

2. Drag the hand-shaped cursor around the project view to move to other areas of the image.

6.4.12 Pan Window

Move around the image within the project view by using the **Pan Window**.

1. To open the **Pan Window**, choose **View > Windows > Pan Window** on the main toolbar.


Pan Window

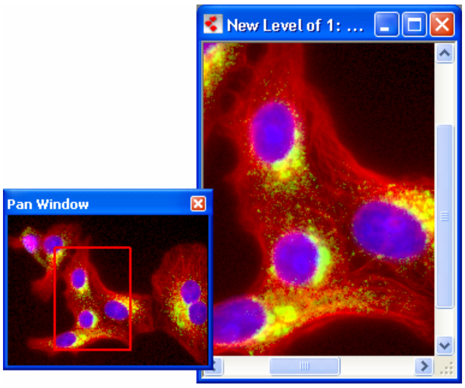


Figure 77: The Pan window and the project view window displaying sample data. (Image data courtesy of EMBL, Heidelberg.)

The red rectangle indicates the area of the image displayed in the project view.

2. To move to a different region of the image, go to the **Pan Window** and click the image outside the red rectangle. The displayed area in the project view will move. The clicked area of the image will be displayed in the middle of the project view.

6.4.13 Magnify the Image in a Separate Window

View a simultaneous magnification of the original image in the project view facilitating detail investigation.

The magnification is 5 times the zoom of the project view.

1. To open the **Magnifier** window, choose **View** > **Windows** > **Magnifier**.
2. Hold the cursor over any point of the image in the project view or the **Pan Window** to center the magnified view at that location.
   The magnified view is of the mixed image layers.


Magnifier

→ Pan Window on page 106

3. Move the cursor over the image and then stop to re-center the magnification.

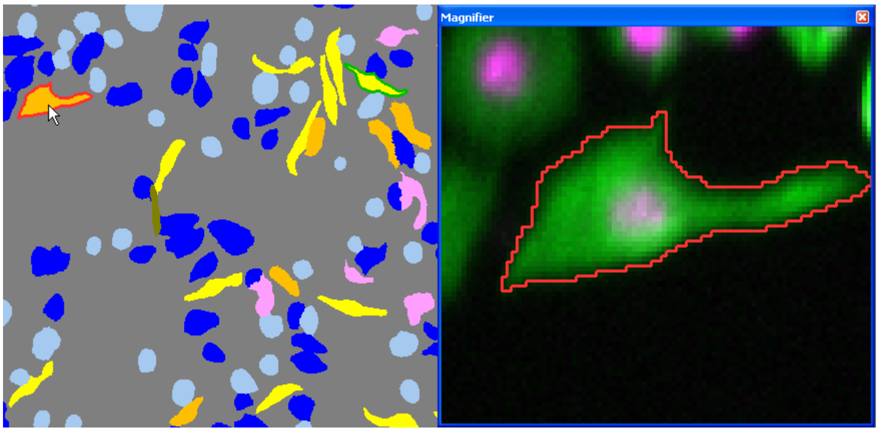


Figure 78: Magnifier Window with corresponding project view image. (Image data courtesy of Cenix BioScience GmhH.)

6.4.14 View a Scale Bar

Display a scale bar on the image in the project view. It helps you to estimate the size of depicted objects by visual comparison.

1. To display a scale, open the **View** menu in the main menu bar and select **Scale Bar,** then **Visible.**
   The scale displays in the corner of the project view, with the length of the scale in the unit of the project.

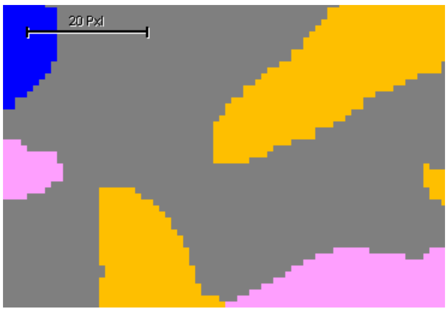


Figure 79: Scale Bar in project view. (Image data courtesy of Cenix BioScience GmhH.)

2. To change the location of the scale, open the **Scale Bar** menu again and choose one of the corners.
   Scale bar settings in the **View** menu apply to all projects in a workspace.
3. To set defaults for the scale, open the **Options** dialog box and make your selections in the **Display Scale Bar** and **Scale Bar default position** fields.

→ Options on page 452

6.5 Compare Rescaled Scene Copy with Original Scene

Compare a rescaled scene copy at a lower scale with the source scene at the original scale.

Precondition: You need to create at least one rescaled scene copy at an lower magnification/resolution.

→ Create a Rescaled Scene Copy at Another Magnification/Resolution on page 374

1. Go to the **Workspace** window and open a project with a rescaled scene.
2. Zoom in so that you can observe the pixel structure of the scene.
3. Click the **Image View or Project Pixel View** button on the **View Settings** toolbar to display of the scene at the original scale. Switch between the display of the scene at the original scale (button activated) and the rescaled magnification/resolution (button released).

Image View or Project Pixel View

6.6 View Outlines of Image Objects

Display the outlines of image objects and view the borders of image objects to explore them in detail.

To toggle between showing and hiding the outlines, click the **Show or Hide Outlines** button on the main toolbar.

Show or Hide Outlines

Depending on the active display mode, the outline colors vary:

→ Change Display of Image Objects on page 109

- Using the **View Layer** display mode, the outline color is the one specified in the **Edit Highlight Colors** settings.

View Layer

→ Edit Highlight Colors on page 444

- Using the **View Classification** display mode, the outlines take on the color of the respective classes.

View Classification

- Using the **View Samples** display mode, the outlines of the samples are colored according to the feature, while all other object outlines are black.

→ View Samples on page 112

- The outlines of a single image objects are visible when selected by clicking on them.

→ Select and Deselect Image Objects on page 115

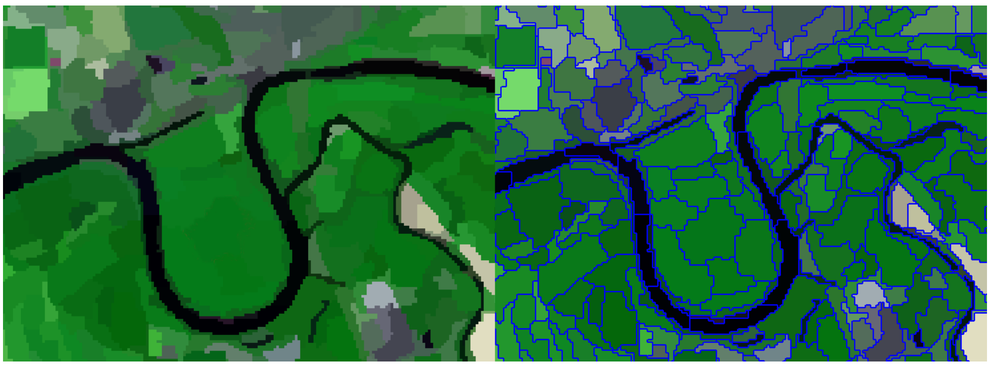


Figure 80: Sample project (Layer View with Object Mean View) with outlines hidden (left) and displayed (right).

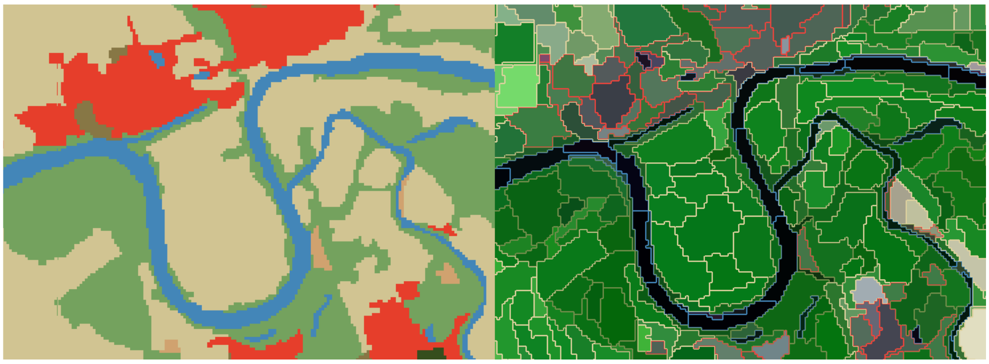


Figure 81: Sample project in Classification View. Note that the outlines take on the color of their respective class. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.7 Change Display of Image Objects

Go to the **View Settings** toolbar and switch between different display modes of image objects.

As an alternative to the toolbar buttons you can use the **View Settings** window.

→ View Settings on page 463

6.7.1 View Layer

View the current image layer.

When viewing a project, click the **View Layer** button on the **View Settings** toolbar to display the current image layer. Mixed image layer values of pixels or image objects make the image layer information itself visible.

 **View Layer**

6.7.2 View Classification

View the classification result.

Click the **View Classification** button to view the classification result. Each image object is colored according to its classification as specified in the Legend window. Unclassified objects are displayed as specified in the **Edit Highlight Colors** dialog box.

 **View Classification**

→ View Legend of Classes on page 380
→ Edit Highlight Colors on page 444

If you are using a split-screen viewing mode, you can use the transparency slider to adjust the transparency of the classification overlay.

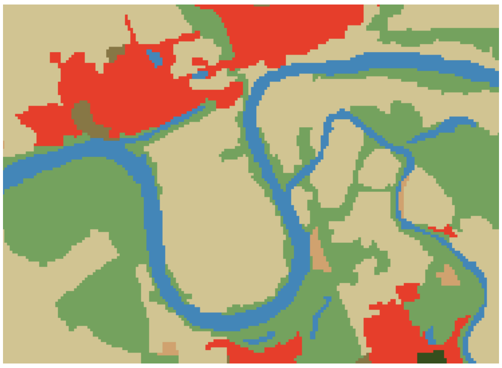


Figure 82: View Classification mode. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.7.3 Adjust Classification Transparency

Use the transparency slider to adjust the transparency of the classification overlay.

To use the slider, you must be viewing classifications and using **Pixel View** rather than **Object Mean View.**

→ Pixel View or Object Mean View on page 111

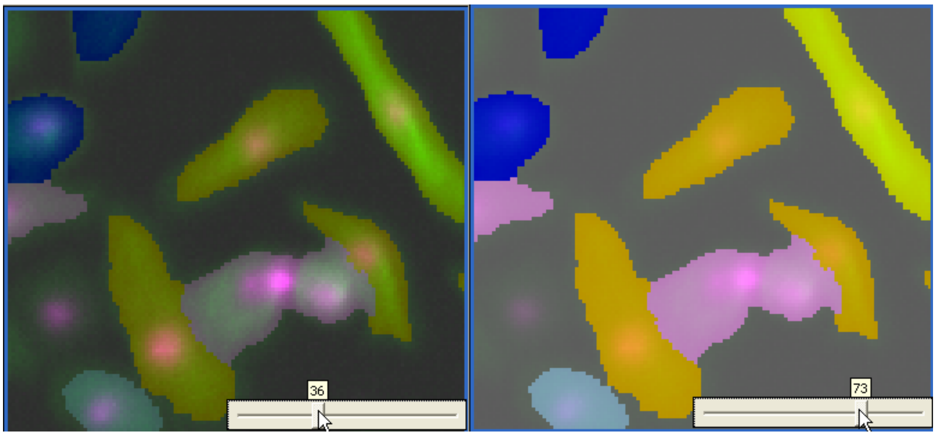


Figure 83: Composite view of the transparency slider (Image data courtesy of Cenix BioScience GmhH.)

1. Hold your mouse over the project view. The slider displays as a small box with a gradient in the right bottom corner of the view.
2. Hold your cursor over the gradient to expand it to a horizontal box with a slider control.
3. Click the control and move it to adjust the transparency of the classification overlay. A transparency scale from 0 to 100 displays directly above the control as you move it.

   You can choose whether to change the display instantly as you move the control by using the **Instant Render Update on Slider** option in the **Options** dialog box.

→ Options on page 452

> **Note**
>
> Projects created with prior versions of **Definiens Developer** will display with the current transparency settings. If you want to use the **export current view** algorithm and preserve the current transparency settings, access the **Algorithm parameters.** Then select **Click to capture current view settings** in the **Save current view settings** field. If you want to preserve the original transparency settings, do not select **Click to capture current view settings.**

6.7.4 Feature View

The **Feature View** button complements the **Feature View** display method.

Feature View

→ Get a Visual Overview of Feature Values on page 180

When selecting features in the **Feature View** window, the **Feature View** button (alternatively press Ctrl + F) is used to return to that feature after another view like **Layer View** or **Classification View** has been used.

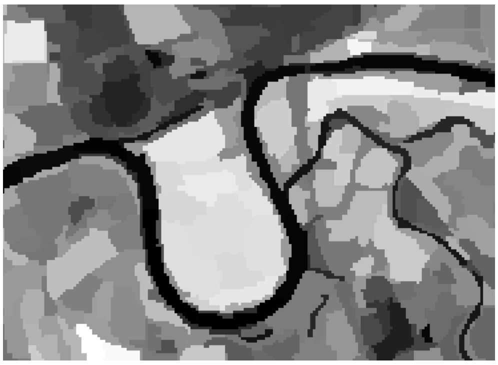


Figure 84: Feature View display method showing the mean values of the image's near infrared layer (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany).

6.7.5 Pixel View or Object Mean View

Switch between pixel view and object mean view.

Use **Layer**, **Classification**, **Sample View**, **or Feature View** to toggle between views:

Pixel View (pressed)

Object Mean View (released)

- **Pixel View** displays the image pixels.
- **Object Mean View** displays a fill color, calculated from the mean value of the pixels' display of each image object.

If the **Classification View** is active, the **Pixel View** is displayed semitransparently through the classification.

→ View Classification on page 109

If the **Feature View** is active, then the **Pixel View** button will not have an affect on the displayed image.

→ Feature View on page 111

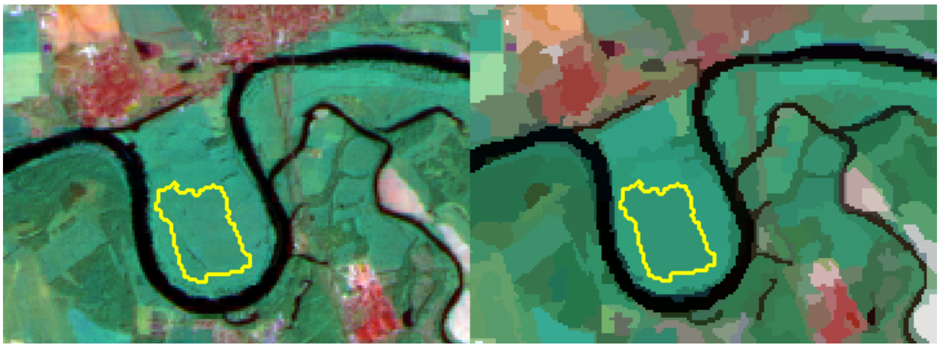


Figure 85: Sample project in Pixel View (left) and Object Mean View (right). The yellow outlines show a selected image object. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.7.6 View Samples

Switch between showing and hiding selected samples.

If sample image objects have been defined in a **Nearest Neighbor Classification**, they maybe revealed in their respective class color in the project view.

→ Use Samples for Nearest Neighbor Classification on page 219

1. To activate the **View Samples** mode go to the **View Settings** window.
2. On the left pane, click **Mode** and select **Samples** in the context menu. Sample objects are visualized in their respective class color.

→ View Settings on page 463

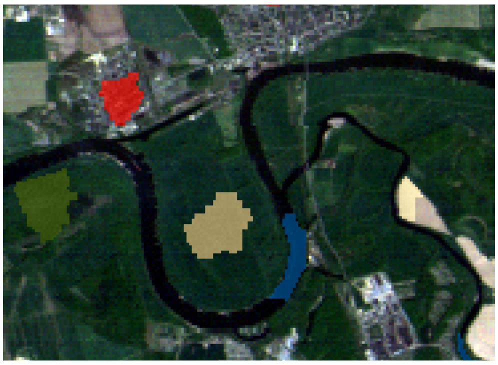


Figure 86: View Samples (Image courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany).

6.7.7 View Polygons

Show or hide polygons of image objects within a project. Polygons are available after the first segmentation of an image.

1. To display polygons, click the **Show/Hide Polygons** button on the **View Settings** toolbar
2. For further options open the **View Settings** window.

 **Show/Hide Polygons**

→ View Settings on page 463

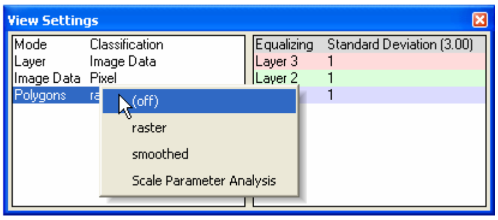


Figure 87: View Settings window with context menu for viewing polygons.

Click on **Polygons** in the left pane and select one of the following polygon display modes:

- **raster**: draws outlines along the pixel borders
- **smoothed**: draws generalized polygons with smoothed outlines
- **Scale Parameter Analysis**: shows the result of the scale parameter analysis in graduated colors

The polygons are displayed in the **Project View**.

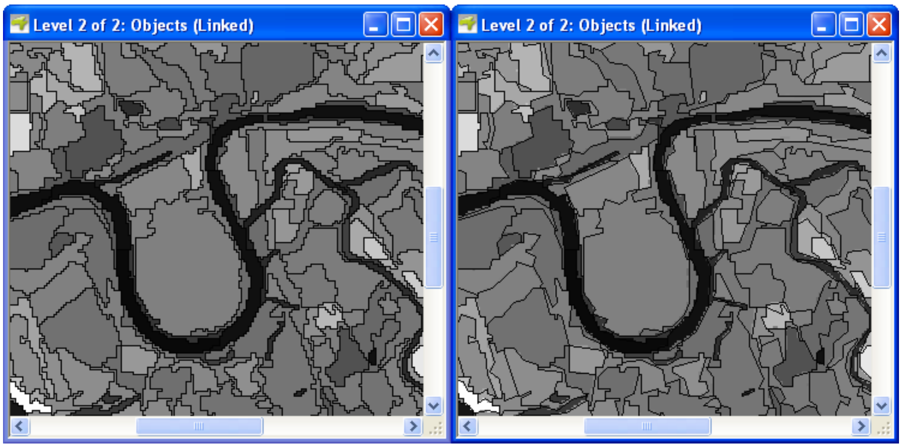


Figure 88: Different polygon display methods in the Project View. Top left: Raster outline mode. Top right: Smoothed outline mode.

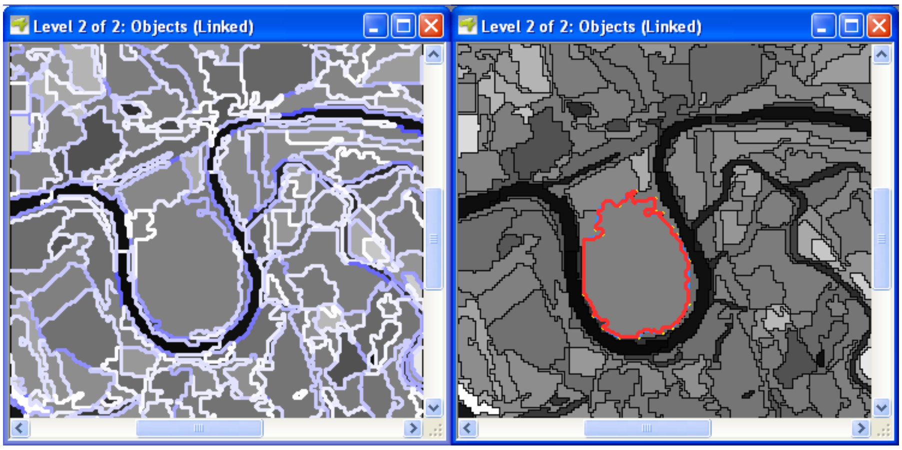


Figure 89: Different polygon display methods in the Project View. Bottom left: Result of scale parameter analysis. Bottom right: Selected image object.

Note

If the polygons cannot be clearly distinguished due to a low zoom value, they are automatically deactivated in the display. In that case, choose a higher zoom value to see the polygons.

If the polygon view is activated, any time you select an image object it will be rendered along with its characterizing polygon. This polygon is more generalized than the polygons shown by the outlines and is independent of the topological structure of the image object level. Its purpose is to describe the selected image object by its shape.

6.7.8 View Skeletons

Display skeletons of objects. Skeletons are automatically generated in conjunction with polygons. An object's skeleton can only be displayed if the object is selected.

→ Display Skeletons on page 294

Select an object and toggle between displaying or hiding skeletons with the **Show / Hide Skeletons** button (if desired, change the skeleton color under **View > Display Mode > Edit Highlight Colors**).

Show / Hide Skeletons

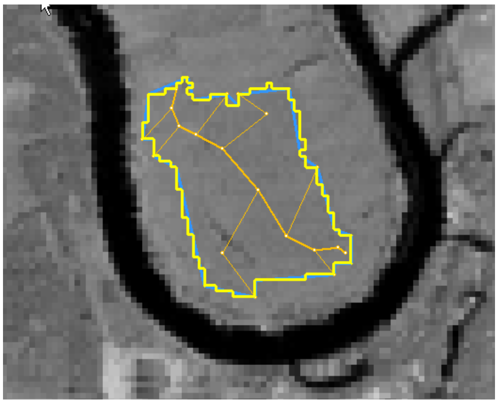


Figure 90: Sample project with one selected skeleton (outline color: yellow; skeleton color: orange)

6.8 Navigate Within Projects

Navigate within the image object hierarchy and select image objects. Further, you can link multiple project views.

6.8.1 Navigate Within the Image Object Hierarchy

Navigate through the different levels of the image object hierarchy in order to be able to maintain information on the different image object levels and to perform classifications.

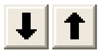
→ Image Object Hierarchy on page 26
→ Use Hierarchical Image Object Levels on page 193

1. To switch through the image object levels do one of the following:

- Select the desired level in the drop-down menu of the **Navigate** toolbar.

Select Level in Image Object Hierarchy

- Use the arrow buttons from the **Navigate** tool bar.



Next Level Down/Up in Image Object Hierarchy

2. The current level of the image object hierarchy is displayed in the status bar in the level field. The number after the slash represents the total number of levels contained in the image object hierarchy.

Figure 91: Level display in the status bar (here: TopLevel, level 1 out of a total of 2 levels)

Tip

Display a List of All Image Object Levels According to the Image Object Hierarchy

If want to get an overview on all image object levels and their sequence within the image object hierarchy, select ***Image Objects*** *>*  ***Delete Levels*** *on the main menu bar. The opening* ***Delete Level*** *dialog box lists the image object hierarchy with all image object levels.*

→ ***Delete Image Object Level*** *on page 197*

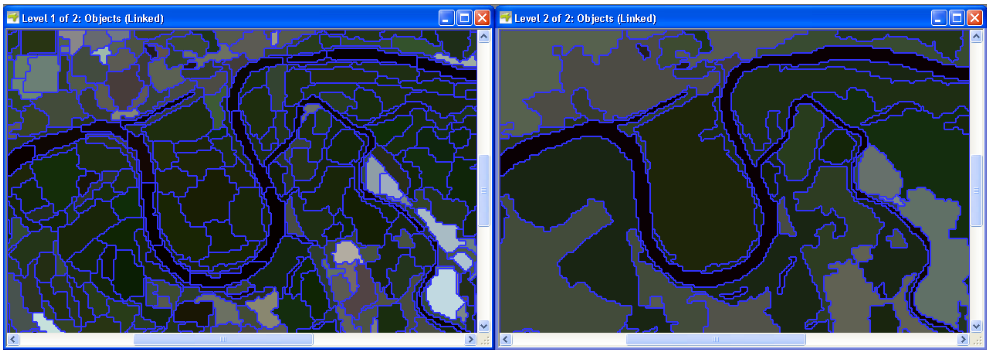


Figure 92: Sample project in level 1 (left) and level 2 (right). (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.8.2 Select and Deselect Image Objects

Select and deselect image objects by clicking on them.

1. To select an image object, do one of the following:

- Left-click the image object. The selected image object appears with a red outline. You can select a second image object by holding down the Ctrl key while clicking. The second image object appears with a green outline.
- Go to the **Image Object Table** and select an image object from the table by clicking it (this is only possible if the **Image Object Table** has been configured before)

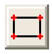
Image Object Table

→ Compare Multiple Image Objects by Using the Image Object Table on page 178

2. To select more than one object, choose from the **Manual Editing** toolbar:
   - **Polygon Selection** selects all objects that are within or touched by a polygon.
   - **Line Selection** selects all objects along a line.
   - **Rectangle Selection** selects all objects within or touched by a rectangle.

Polygon Selection

Line Selection



Rectangle Selection

3. To deselect an image object, do one of the following:
   - Click the same image object again.
   - Click another image object. This deselects the previous object and selects the new one.
   - Click inside the white space of the project view.
     This is only possible when the image is minimized enough to see the white space at the edge of the window. Depending on your operating systems settings, the mentioned area may have another color.

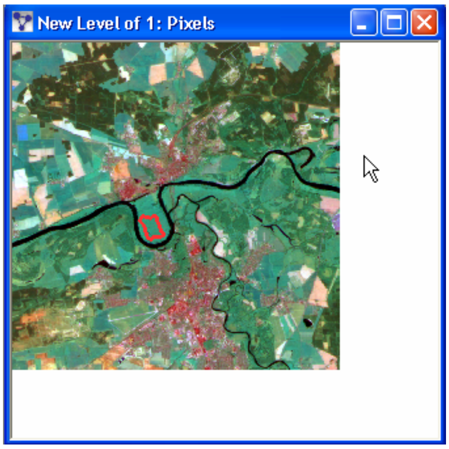


Figure 93: Clicking inside the white space of the project view deselects a selection.

6.9 Use Different Viewing Modes

Split the project view into two or four and then display different view settings and image object levels in each resulting project view or section of a single project view.

Any selected image objects will also be selected in all levels of all resulting project views.

The default method of viewing mode is one project view with the same view settings applied over the entire view.

1. To use the different viewing modes, you must first split a single project view into two or four display areas and then select a viewing mode. To split the project view, select **Window** on the main menu bar and then choose one of the following:

- Select **Split** to display movable cross-hairs in the project view. Click anywhere in the project view to create the inner frames of the separate sections or project views.
- Select **Split Horizontally** to display top and bottom sections or project views.
- Select **Split Vertically** to display right and left sections or project views.

2. Then select one of the following viewing modes under the **Window** menu:
   - **Independent View**: Unlink the project views and select independent pan and zoom values in each one.
   - **Side by Side View**: Link the project views, so the pan and zoom value is the same for all. Different view settings can be selected for each project view.
   - **Swipe View**: Change the view settings in wipeable sections of a single project view flexibly by dragging the view settings for one section across the project view.

6.9.1 Use Independent Project Views

Use two or four project views with independent view settings and independent panning and zoom values.

Precondition: You must first split the single project view using one of the splitting options on the **Window** menu of the main menu bar.

1. To display independent project views, choose **Window > Independent View** on the main menu bar.
2. Click any project view to make it the active project view, with a blue border.
    - Change any view setting and zoom values in the active project view without affecting the other project views.
    - You can also pan independently in the active window.
    - Each project view displays its own scroll bars, as needed.
3. Drag the inside frames of any project view to reveal or conceal more or less of a view. Drag an inside frame to the outer boundary of the window to remove it entirely and partially or totally consolidate the project views.
4. To reconsolidate the project views, select **Window > Split Horizontally** and/or **Window > Split Vertically.**

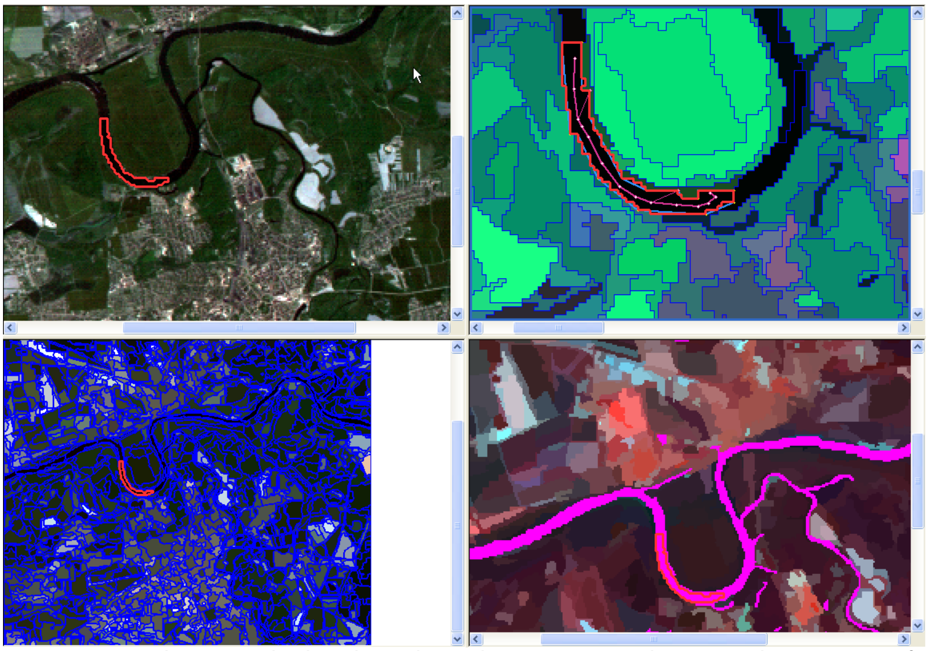


Figure 94: Sample project displayed in independent viewing mode. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.9.2 Use Linked Side-by-Side Project Views

Link panning and zoom values of two or four project views, each with separate view settings.

Precondition: You must first split the single project view using one of the splitting options on the **Window** menu of the main menu bar.

1. To view side-by-side project views with linked pan and zoom, select **Window > Side-by-Side View** on the main menu bar.
    - Any project view that you click becomes the active window, with a blue border and scroll bars that display as needed.
    - Use the active window to control the pan and zoom of the other project view(s).
    - Drag the inside frames of the project views to reveal or conceal more or less of each project view. Dragging a frame to the outer boundary of the window will remove it entirely.
2. Select another viewing mode to consolidate the views or to unlink them.

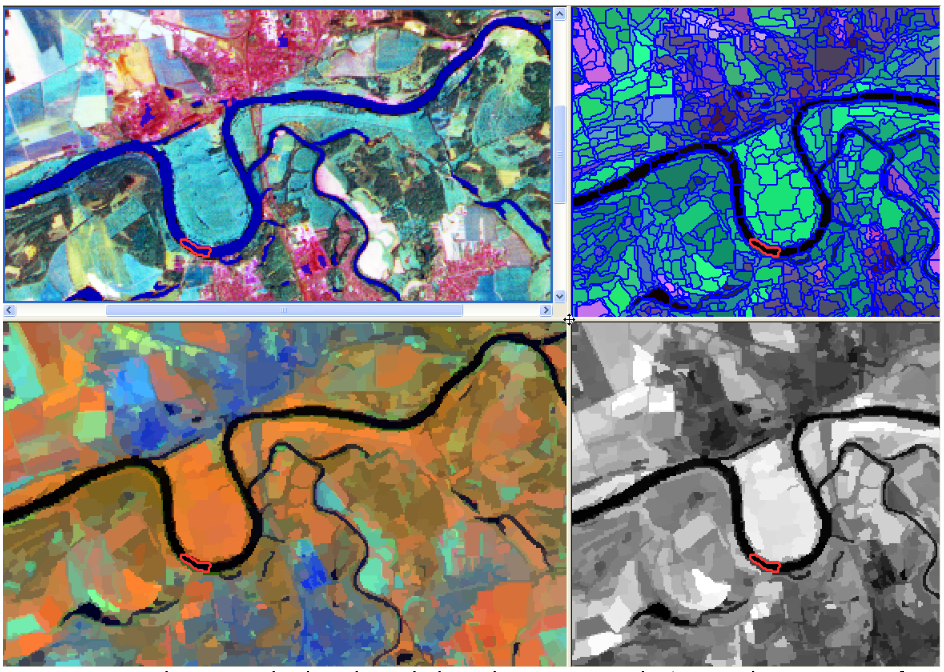


Figure 95: Sample project displayed in side-by-side viewing mode. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

6.9.3 Use Flexible Visualization with Swipe View

Apply the view settings and image object level for one section of a divided project view across the project view by dragging and resizing sections. Because the project view is not resampled, the display changes continuously as the sections are resized.

Precondition: You must first split the single project view using one of the splitting options on the **Window** menu of the main menu bar.

1. Select **Window >Swipe View** in the main menu bar. The active section of the project view has a blue border and scroll bars which display as needed.
2. Click in any section to activate it.
3. Select any desired independent view settings and/or image object level to display in the active section of the project view.
4. Drag any inside frame across the project view to the apply the view settings and image object level of the sections
   - Drag a horizontal frame to apply the view settings and levels of top sections or bottom sections.
   - Drag a vertical frame apply the view settings and levels of left sections or right sections.
   - Drag an inside corner diagonally to apply the view setting and level of one section to the other three.
5. Drag an inside frame to the project view boundary to delete the frame and partially or totally consolidate the sections of the project view.

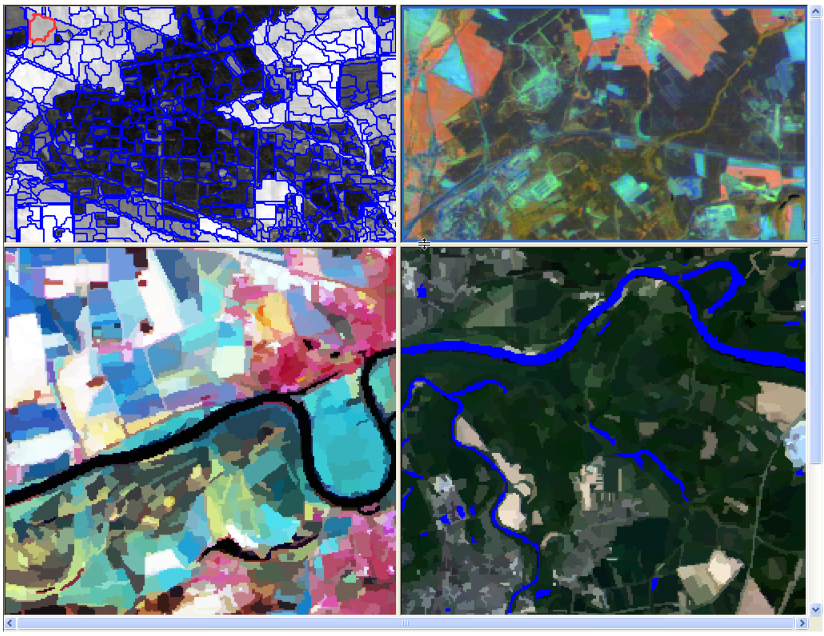


Figure 96: Sample project displayed in swipe viewing mode. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

7 Start Developing Rule Sets

This chapter introduces the technology and the development workflow and explains the basic tools. It enables you to create basic rule sets for simple image analysis tasks.

→ **How to Learn Developing Rule Sets** on page 121
→ **Introduction into Object-Oriented Image Analysis** on page 124
→ **Use Processes to Build Rule Sets** on page 130
→ **Create Image Objects by Segmentation** on page 157
→ **Get Information on Image Objects** on page 167
→ **Classify Image Objects** on page 190

To enable fast and easy learning, this chapter focuses on the main concepts and functions only. Two more chapters about rule set development follow.

→ **Three Chapters About Rule Set Development** on page 123

7.1 How to Learn Developing Rule Sets

In addition to learning the basic functions of handling data—which are loading, managing, viewing, processing, and exporting—you have to achieve some knowledge about developing rule sets with **Definiens Developer**.

→ **Workflow** on page 32

Before continuing, you may want to repeat the basic handling functions. Consult the **Quick Start** section for suggestions about effective learning paths.

→ **Get a Quick Start** on page 14

1. Start learning about developing **Definiens** rule sets by using the **Tutorial Basic Knowledge 1**. Using sample data, you perform a very simple process sequence. Thus, you get a basic idea about the common elements of the **Definiens Developer** user interface and the basic workflow of **Definiens** image analysis.

   → **Tutorials and Guided Tours** on page 470

   Go to the data folder **TutorialBasicKnowledge1**, which is installed in the **Definiens Developer** installation directory. The default path is **C:\Program Files\Definiens Developer** *Version number***\Examples \TutorialBasicKnowledge1**.

2. Repeat the **Basic Concepts** chapter of this User Guide to include unknown basic concepts related to development functions.

   → **Basic Concepts** on page 19

3. If you are using **<podname> Developer** for geographic image analysis, we recommend that you work through the guided tour **Mapping Impervious Surface** presenting a geographic example of image analysis. It includes sample data to follow in practice. This guided tour focuses on the basic steps involved in developing a rule set using **Definiens Developer** and submitting a set of images for batch processing.

   Guided tours are installed separately. Consult the **Installation Guide** for details.

   → **Installation Guide** on page 469

4. After reviewing starting instructions for creating basic rule sets, continue by learning how to benefit from all tools of the development environment. **The Develop Efficient Rule Sets** chapter provides detailed handing instruction to build efficient and reusable rule sets and other ruleware.

   → **Start Developing Rule Sets** on page 121
→ **Develop Efficient Rule Sets** on page 192

Simultaneously, you should consult the **About Strategies** chapter focusing on composing rule sets based on the **Definiens** Cognition Network Language (CNL). Learn how to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ **About Development Strategies** on page 324

7.1.1 Recommended Skills

Working with **Definiens Developer** requires some skills and previous knowledge for successful rule set development. Thus we assume the following user skills:

- Good skills in handling standard office software on **Microsoft Windows** systems.
- Basic knowledge about image analysis such as image data generation and common image data file formats.
- Basic software programming skills.

Further we highly recommend that users have a certain level of competence about the visual content of the image data to be analyzed. This knowledge about the information depicted and its meaning is a crucial basis for creating rule sets. Depending on industry and the images you analyze we recommend:

In Life Sciences

- Basic knowledge in biology may be sufficient, if developers have access to an expert in the field to refer for example in cases of ambiguous image content.

In Remote Sensing

- Basic remote sensing skills including knowledge about the analysis of aerial and satellite imagery.
- Basic knowledge about geoinformatics.

7.1.2 Main Learning Areas for Developing Rule Sets

When learning how to develop rule sets you have to work in two main areas: how to compose rule sets based on the **Definiens** Cognition Network Language (CNL) and how to use the tools for developing rule sets within the provided development environment. This is somewhat like cooking,with two distance main learning areas, one being the recipes and the other the tools, such as stove, pot and spoon.

How to Compose Rule Sets Based on the Definiens Cognition Network Language (CNL)

This User Guide provides you with concepts, instructions and strategies about developing rule sets.

These chapters enable you to begin composing CNL code into rule sets and collect your first development experiences. To build and strengthen advanced development competencies, you need to perform some learning-by-doing:

→ **Basic Concepts** on page 19
→ **How to Approach Developing a Rule Set** on page 324
→ **Generic Development Strategies** on page 337
→ **Find and Handle Image Structures** on page 351

How to Use the Tools for Developing Rule Sets

In addition to the basic functions of handling data—which are loading, managing, viewing, processing, and exporting—various functions and tools are available to build rule sets that suit your needs regarding both insight and efficiency.

→ **Workflow** on page 32

This User Guide instructs you in performing various function related tasks:

→ **Use Processes to Build Rule Sets** on page 130
→ **Create Image Objects by Segmentation** on page 157
→ **Get Information on Image Objects** on page 167
→ **Classify Image Objects** on page 190
→ **Use Hierarchical Image Object Levels** on page 193
→ **Focus Processes by Using the Domain Concept** on page 197
→ **Use Variables in Rule Sets** on page 257
→ **Use Customized Features** on page 269
→ **Reuse Process Sequences with Customized Algorithms** on page 276
→ **Working with Thematic Layers** on page 283
→ **Working with Polygons and Skeletons** on page 290
→ **Automate the Workspace Processing** on page 295
→ **Create Action Libraries** on page 306

While developing rule sets, the related Reference Book helps with details of available algorithms and features.

→ **Reference Book**

7.1.3 Three Chapters About Rule Set Development

This User Guide provides three chapters about developing rule sets:

- **Start Developing Rule Sets** introduces the technology and the development workflow and explains the basic tools. This chapter enables you to create basic rule sets for simple image analysis tasks.

→ **Start Developing Rule Sets** on page 121

- **Develop Efficient Rule Sets** instructs you in using the available tools to build efficient and reusable rule sets and other ruleware. This chapter enables you to benefit from the tools of the development environment.

→ **Develop Efficient Rule Sets** on page 192

- **About Development Strategies** focuses on composing rule sets based on the **Definiens** Cognition Network Language (CNL). This chapter enables you to develop rule sets that suit your needs concerning both insight and efficiency and supports you in handling common problems.

→ **About Development Strategies** on page 324

7.1.4 Definiens Training

Get up to speed quickly—or enhance your current expertise—with **Definiens** training courses.

The **Definiens** training team has developed a hands-on curriculum designed specifically to hone the skills of users.

Workshops are held in **Definiens**' classrooms around the world and at customers' sites. Training courses can be tailored to satisfy your unique needs, thereby maximizing the training effect.

To get more information about training, see **Definiens** webpages or contact us via email.

→ http://www.definiens.com/services/training.php
→ training@definiens.com

7.2 Introduction into Object-Oriented Image Analysis

This section introduces into the technology of object-oriented image analysis and gives you a brief overview on the basic development workflow.

→ Workflow on page 32

Via cross references, each subsection guides you to related detailed instructions.

7.2.1 About Image Analysis with Image Objects

The aim of an image analysis with **Definiens Developer** is image object classification that enables the generation of numerical values and statistics. **Definiens** image analysis is based on a simple principle:

The automatic detection of objects in a digital image starts with cutting the image into pieces called image object primitives that are rather homogeneous inside and have shape and size such that they can serve as building blocks for further processing. This step is called segmentation.

→ Create Image Objects by Segmentation on page 125

Next, image objects are labeled according to attributes, such as shape, color, and embedding. This step is called classification. The next step is another segmentation: the classified image objects to yield even more meaningful image objects. It is a central property of object based image analysis that you can restrict any operation—segmentation or classification—to image objects of one or more classes.

→ Classify Image Objects on page 127

This workflow of iterative segmentation and classification is continued until the desired image objects are isolated. Each segmentation should be designed to facilitate the subsequent classification and vice versa.

→ About the Interplay between Segmentation and Classification on page 128

7.2.1.1 About the Fractal-Hierarchical Approach

The language behind **Definiens** software, the Cognition Network Language (CNL), is a special implementation of a more general type of technology, **Definiens** Cognition

Network Technology (CNT). **Definiens** Cognition Network Technology (CNT) is a fractal-hierarchical way of describing the world.

Independent of the system that one is observing, structures and dynamics of different scales are always involved. Not one of these scales but all together represent the whole and its specific characteristics. Different structures at the same scale, as well as structures between different scales show symmetry and symmetry-break.

Understanding that the embedding hierarchy is a ubiquitous phenomenon, we conclude that a simulation of complex systems as well as images needs to fulfill the following conditions:

1. It represents all scales that are of relevance for the specific question simultaneously.

2. It describes meaningful relations between the structures and dynamics on both the same scale and between different scales.

In the **Definiens** software the fractal-hierarchical approach is realized in the form of the hierarchical image object domain.

7.2.2 Create Image Objects by Segmentation

→ Create Image Objects by Segmentation on page 157
→ Scene on page 22

The fundamental step of any **Definiens** image analysis is a segmentation of a scene—representing an image—into image object primitives. Thus, initial segmentation is the subdivision of an image into separated regions represented by basic unclassified image objects called image object primitives.

For successful image analysis, defining object primitives of suitable size and shape is of utmost importance. As a rule of thumb, good object primitives are as large as possible, yet small enough to be used as building blocks for the objects to be detected in the image. Pixels are the smallest possible building block, however pixels have limited information. To get larger building blocks, different segmentation methods are available to form contiguous clusters of pixels that have larger property space.

Commonly, in image processing, segmentation is the subdivision of a digital image into smaller partitions according to given criteria. In contrast, within the **Definiens** technology, each operation that creates new image objects is called segmentation, no matter if the change is achieved by subdividing or by merging existing objects.

Different segmentation algorithms provide several methods of creating of image object primitives.

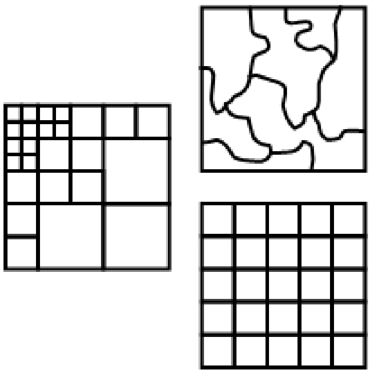


Figure 97: Different segmentation results: Multiresolution (above), Quadtree (left), Chessboard (below).

Segmentation Creates a New Image Object Level

The new image objects created by segmentation are stored in what is called a new image object level. Each image object is defined by a contiguous set of pixels, where each pixel belongs to exactly one image object. Each of the subsequent image object related operations like classification, reshaping, re-segmentation, and information extraction is done within an image object level. Simply said, image object levels serve as internal working areas of the image analysis.

→ Image Object Level on page 24

7.2.3 Get Information on Image Objects

While developing a rule set for image analysis you need information about the attributes of image objects, such as their color or shape. Additionally, you need information on the environment of image objects, for example average brightness of all neighbors of the image object in question. There are different methods for access and display of that information.

→ Get Information on Image Objects on page 167

The goal is to recognize the complex visual information which is contained within images and to translate this into processes for image analysis. Obtaining comprehensive knowledge about the features of an image and its image objects is usually good preparation for rule set development and this knowledge is also needed in most of the subsequent workflow steps. Thus, a thorough assessment and prior knowledge of the domain semantics of the image are essential preparations that will guide the detection of objects of interest.

In **Definiens** technology, an image object attribute is called a **feature**. There are simple, inherent image object features that inform about the shape or color of a single image object. They are calculated from the pixels that are a part of the image object. Other features return values based on how the image object relates to other image objects in its horizontal and vertical neighborhood. More features provide information on the classification or on the texture. Additionally, there exists functionality for assembling customized features using existing features.

→ About Features as a Source of Information on page 168

Thus, features represent a source of information concerning image objects. They help you to:

- **Investigate single image objects**: The **Image Object Information** window allows you to look at a selection features for one image object at a time.

→ Investigate Single Image Objects by Using the Image Object Information on page 174

- **Compare multiple image objects**: In the **Image Object Table** window you can list several features for many image objects, with one row for each image object, one column for each feature. You can sort the image objects by any feature. You can also navigate though the image using the table: the image object clicked in the table will be highlighted in the image.

→ Compare Multiple Image Objects by Using the Image Object Table on page 178

- **Compare feature values**: The **Feature View** display method and the **Feature View** window are functionalities that will color-code each image object based on the value returned from the feature selected.

→ Compare Feature Values by Using the Feature View on page 179

- **Analyze the correlation of two features**: The **2D-Feature Plot** window allows you to compare for all image objects two features in a separate graph, where to each axis a feature is assigned. Image objects are displayed as dots of the color representing their classification in that graph.

→ Analyze the Correlation of Two Features by Using the 2D Feature Space Plot on page 188

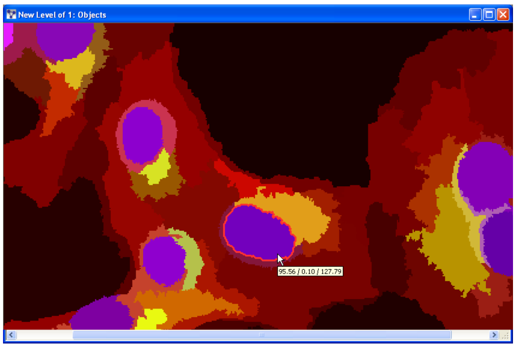


Figure 98: Project view of selected image object values. (Image data courtesy of EMBL Heidelberg).

7.2.4 Classify Image Objects

After image objects have been created in your scenes, you classify them to give them both a meaning and a label. Information contained in image objects is used as a filter for classification.

→ Classify Image Objects on page 190

Based on a classification of image objects, you can analyze and interpret complete images. To perform a classification, appropriate classes need to be defined. During classification, the image objects are analyzed according defined criteria and assigned to classes that best meet the defined criteria.

7.2.5 About the Interplay between Segmentation and Classification

When creating rule sets you always start with a segmentation procedure to create an initial set of image objects. These image object primitives should be as meaningful as possible. Afterwards, the process sequence commonly follows a general pattern.

The pattern consists of alternating between segmentation (or reshaping) and classification of defined image objects. Each segmentation step should create image objects that are most functional for subsequent classification. In turn, each classification step should sort image objects in a way that is most functional for subsequent segmentation.

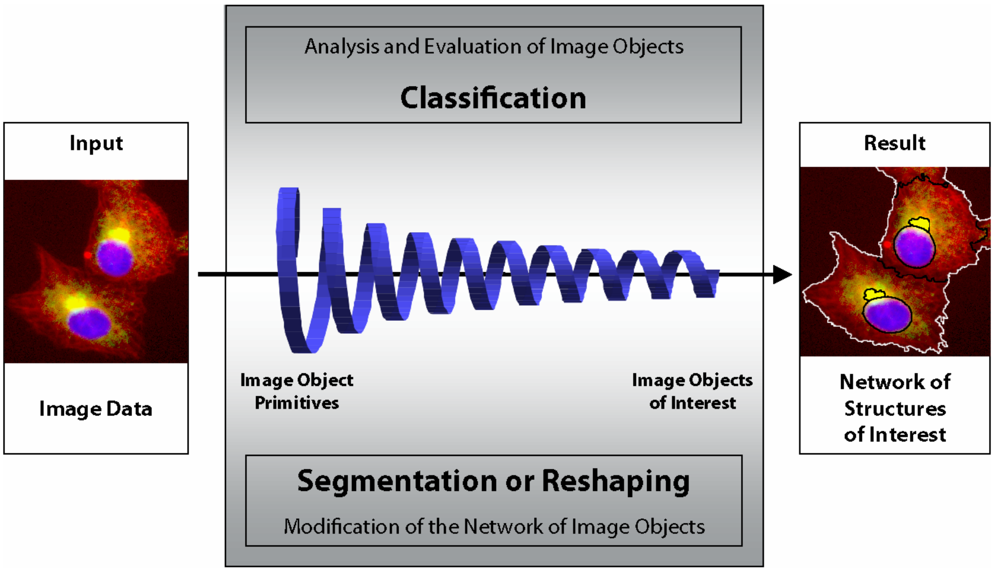


Figure 99: The image analysis workflow iterates the segmentation (or reshaping) and classification of image objects.

For **segmentation**, basic segmentation algorithms can be used, like **chessboard segmentation**, **quadtree segmentation**, or **multiresolution segmentation**. However, later during the process, more knowledge-based reshaping algorithms will be necessary.

For **classification**, the simple **assign class** algorithm enables you to classify based on a single object feature and a threshold, whereas classification allows a fuzzy-logical combination of different features. More complex algorithms classify based on the embedding of an image object in another image object or in an area of image objects of a specific class.

Find Out More

Developing Rule Sets for Image Analysis

Generally, you advance in small steps to develop robust rule sets for image analysis tasks. In fact, you can develop a short process working quick and perfect for a limited number of images. However, step-by-step rule sets are usually more flexible and can be applied to a series of images which always have some differences to consider.

Comparable to playing golf, you may try to hit any hole by a single, well-trained strike. However, it is wiser to reach the targets by multiple standard strikes.

→ ***How to Approach Developing a Rule Set*** *on page 324*

7.2.6 Use Hierarchical Image Object Levels

Although you can perform some image analysis on a single image object level, the full power of the **Definiens** object oriented image analysis unfolds when using multiple image object levels. On each of these levels, a number of objects is defined by the image objects on the level below that are considered their sub-objects. In the same manner, the lowest level image objects are defined by the pixels of the image that belong to them.

→ Use Hierarchical Image Object Levels on page 193

In this hierarchical structure, the image object hierarchy, each image object provides access to information about its neighbors, sub- and super-objects at all times. By connecting image objects vertically, access to scale and advanced texture properties is possible. The image object hierarchy allows the representation of image information at different spatial resolutions simultaneously.

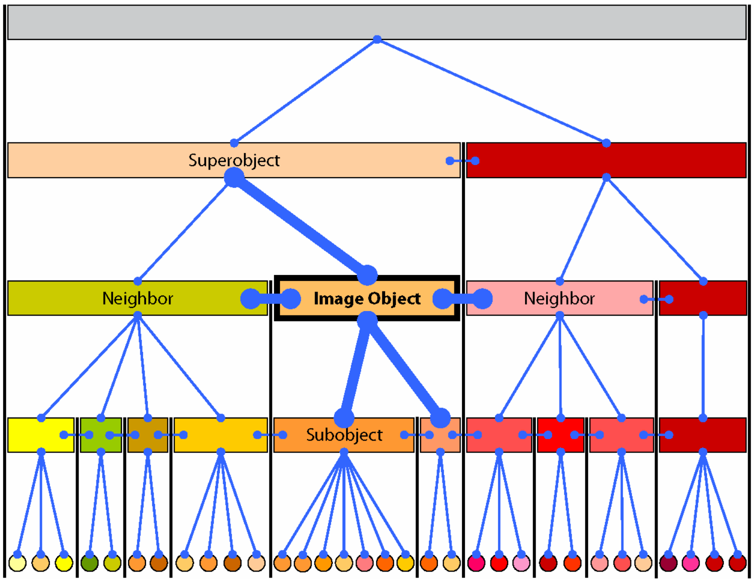


Figure 100: Within the image object hierarchy, each image object is linked to its neighbors, its superobject, and its subobjects.

7.2.7 About the Development Workflow

The process of detecting objects in an image consists of a sequence of segmentation and classification procedures. The sequence is defined in the rule set, which represents the program code. It consists of a list of processes, which represent commands, organized in a tree-like structure.

The typical workflow of the rule set development consists of three iterative activities:

- Use one the different ways of retrieving information about image objects.

→ Get Information on Image Objects on page 126

- Edit one or more lines in the rule set.

→ Use Processes to Build Rule Sets on page 130

- Run a branch of the rule set process tree for testing.

→ Execute Processes for Testing on page 151

At the end you should have a rule set, that—when executed entirely on a image data—detects the image objects of interest as defined by the image analysis task.

If you are already experienced in rule set development, you may continue with the extended workflow of rule set development focusing the strategic approach.

→ How to Approach Developing a Rule Set on page 324

7.2.7.1 Use Processes to Build Rule Sets

Processes are the fundamental tool for editing and sequencing development functions.

→ Use Processes to Build Rule Sets on page 130

Learning about rule set development starts by understanding processes and using the **Process Editor** dialog box for editing processes.

→ Edit a Single Process on page 131

The **Process Tree** window is the central place where rule sets are organized.

→ Organize a Process Sequence on page 141

Other common windows provide information that you need for developing the rule set. These are the **Class Hierarchy** window, the **Image Object Information** window, and the **Feature View** window.

→ Get Information on Image Objects on page 167

7.3 Use Processes to Build Rule Sets

A single process represents an individual operation of an image analysis routine. Thus, it is the main working tool for developing rule sets. A single process is the elementary unit of a rule set providing a solution to a specific image analysis problem.

→ About the Development Workflow on page 129

Every single process has to be edited to define an algorithm to be executed on an image object domain.

Combine single processes in a sequence by building a rule set. You can organize a process sequence with parent and child processes which are executed in a defined order. You can also load an existing rule set, save your rule set, and execute individual processes or the complete rule set.

Developing a rule set does not require you to write any code, rather, it is a concept applicable within the graphical user-interface.

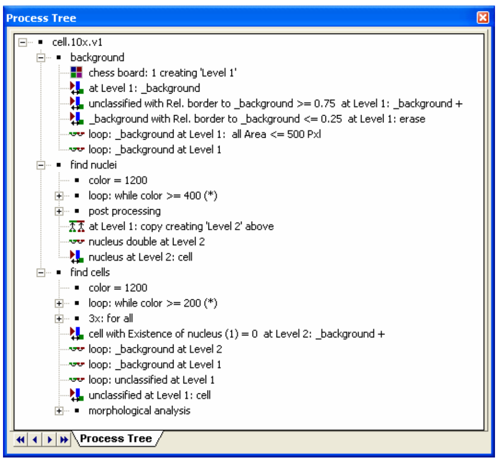


Figure 101: Process Tree window.

Note

In **Definiens Developer**, the term **Process** is used for both a single process and a process sequence.

In this section you learn:

- How to edit a single process

→ Edit a Single Process on page 131

- How to organize single processes in a process sequence using the **Process Tree** window

→ Organize a Process Sequence on page 141

- How to execute processes for testing.

→ Execute Processes for Testing on page 151

- About available algorithms that can be executed by processes

→ Algorithms Overview on page 155

7.3.1 Test Project

For better understanding of the topics in this chapter, we recommend opening a test project which is included within the installation. The test project is provided so that you might experiment with different basic methods and settings useful in analyzing information concerning image objects.

Test Project

1. To open the test project, launch **Definiens Developer**, create a **New Project** calling it **Test Project**, and select the image data. The path is:
   C:\Program Files\Definiens Developer
   *Version number***Examples\TestProjects\Cells.tif**

   New Project

2. Make sure the following windows are open:
   - **Image Object Information**
   - **Feature View**
   - **Process Tree**

   Image Object Information

   Feature View

   Process Tree

3. If you are an experienced user of a previous **Definiens** product, then you may append one of the segmentation processes and begin to explore the new **Image Object Information** features and **Image Layers**.
4. Inexperienced or new users should work their way through this section of the users guide to gain an understanding of how to build processes for automated image analysis.

7.3.2 Edit a Single Process

A single process executes an algorithm on an image object domain what is a set of image objects.

For process creation and modification you mainly use the following dialog boxes:

- **Edit Process** dialog box for creating and modifying process

→ Create a Single Process on page 132

→ Modify a Single Process on page 140

- **Process Tree** window for sequencing processes

→ Organize a Process Sequence on page 141

7.3.2.1 Create a Single Process

Create a single process using the **Edit Process** dialog box. This section gives you an overview about creating and editing a single process.

A single process executes an algorithm on an image object domain. A image object domain is a set of image objects which can be unclassified or classified. Every process loops through this set of image objects one by one and applies the algorithm to each single image object.

Precondition: While editing processes, keep the **Process Tree** window open.

Process Tree

1. To open the **Edit Process** dialog box, do one of the following:

   - Go to the **Process Tree** window, which might be empty or already contains a process. Right-click the **Process Tree** window and select **Append New** from the context menu.
   - Choose **Process** > **Process Commands** > **Append New** on the main menu bar.
   - Type Ctrl + A on the keyboard.

   The **Edit Process** dialog box opens.

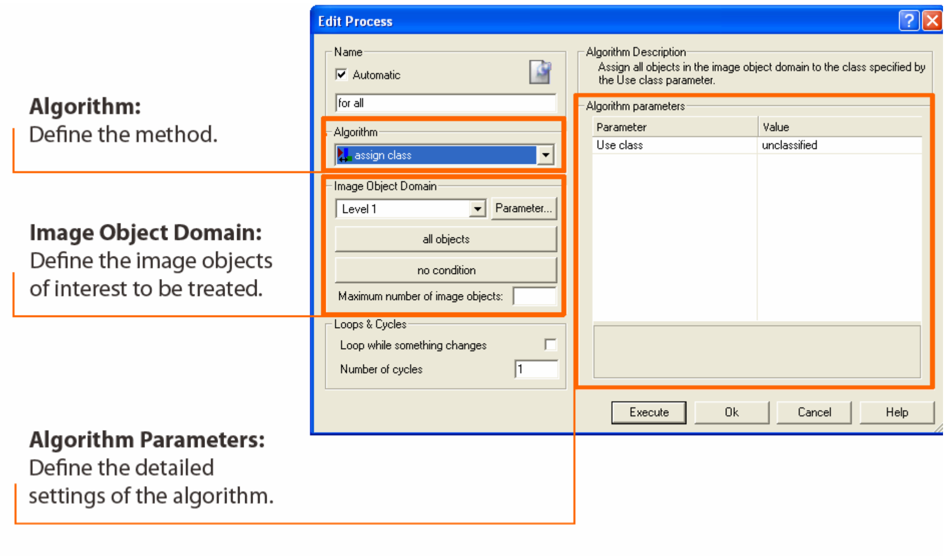


Figure 102: Edit Process dialog box with highlighted group boxes.

   The **Edit Process** dialog box offers three group boxes (main areas) for process creation:

   - **Algorithm**
   - **Image Object Domain**
   - **Algorithm Parameters**

2. Define the method of the process in the **Algorithm** group box. For information on specifying the settings see **Define the Algorithm**.

→ Define the Algorithm on page 133

3. Define the image objects of interest on which an algorithm should be performed in the **Image Object Domain** group box. For information on specifying the settings see **Define the Image Object Domain**.

→ Define the Image Object Domain on page 135

4. Define the individual settings of the algorithm in the **Algorithms Parameters** group box. We recommend that you do this after you decided about the image object domain (step 3). For information on specifying the settings see **Define the Algorithm Parameters**.

→ Specify Algorithm Parameters on page 134

5. To confirm the process, do one of the following:

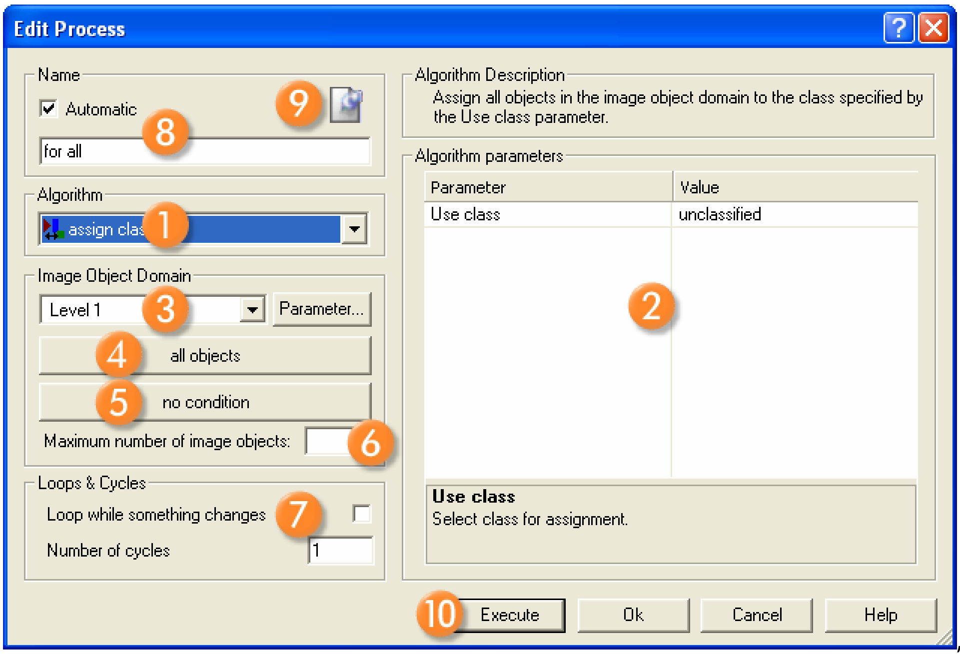
   - Click **OK**. The confirmed process will be noted in the **Process Tree** window. The process will not be executed.
   - Click **Execute** 10 to confirm and execute the process.
   - Press Enter on the keyboard to confirm and execute the process.

Additional process settings can be done concerning repeated execution of processes, called looping.

→ Repeat Process Execution on page 138

Apart from automatic naming of processes, you can edit self-explaining process and comments.

→ Name the Process on page 139



Define the Algorithm

Define the algorithm that a single process will execute on the image object domain.

1. Define the method of the process in the **Algorithm** 1 drop-down list box. For detailed descriptions of available algorithms see the separate **Reference Handbook**.

→ Reference Handbook

2. In the **Algorithm** 1 drop-down list box you find by default a limited selection of algorithms. To select more rarely used algorithms, click **More** which is the last item in the drop-down list. The **Select Process Algorithms** dialog box opens.

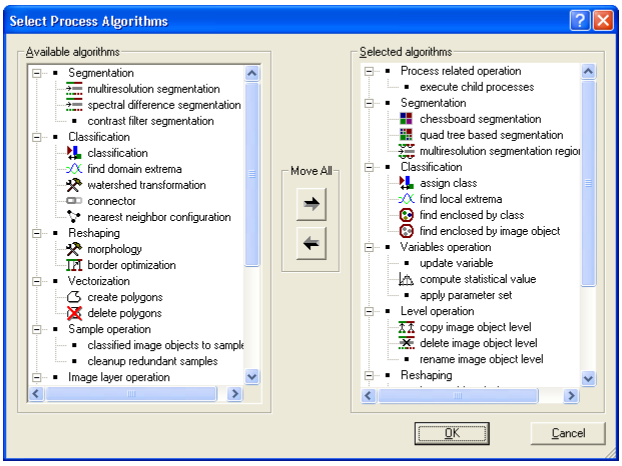
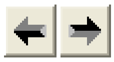


Figure 103: Select Process Algorithms dialog box

3. The list on the left offers all additionally available algorithms. Do one of the following:
   - Click an algorithm from the left list to select it for the current process. Click **OK** to go back to the **Edit Process** dialog box with the selected algorithm in the **Algorithm** 1 drop-down list box.
   - Double-click an algorithm from the left list to add it to the **Algorithm** 1 drop-down list box. It will be added to the list on the right. Continue with step 4.
4. The list on the right itemizes all algorithms to offer in the **Algorithm** 1 drop-down list box. Double-click an algorithm on the right to remove it from the **Algorithm** 1 drop-down list box. It will be returned to the list on the left.
5. To move all list items at once, use the **Move All** buttons in the middle.


Move All

6. Click **OK** to go back to the **Edit Process** dialog box with the modified **Algorithm** 1 drop-down list box.

Specify Algorithm Parameters

Depending on the chosen algorithm, you have to specify different parameters.

1. Define the individual settings of the algorithm in the **Algorithms Parameters** 2 group box. If available, click a plus sign (+) button to expand the table to access additional parameters.


(expand)

2. To edit **Values** of **Algorithm Parameters**, select the parameter name or its value by clicking. Depending on the type of value, change the value by one of the following:
   - Edit many values directly within the value field.
   - Click the ellipsis button located inside the value field. A dialog box opens allowing you to configure the value.


(ellipsis button)

- Click the drop-down arrow button placed inside the value field. Select from a drop-down list to configure the value.


(drop-down arrow button)

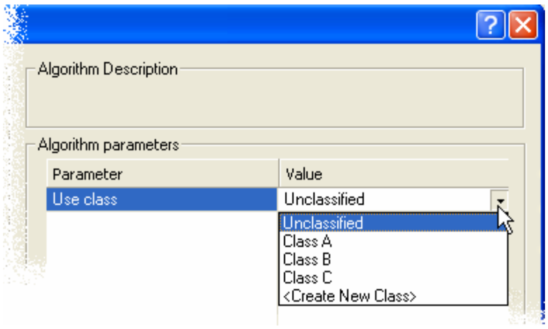


Figure 104: Select an Algorithm Parameter for editing values.

For details about the various algorithm parameters see the separate **Reference Handbook**.

→ **Reference Handbook**

Define the Image Object Domain

Each process executes an algorithm on an image object domain you have to define.

The **Image Object Domain** is a filter to narrow down the area of interest and thereby the number of image objects to be classified.

The image object domain consists of three major components to set:

- Image object level: Select an existing level or let the process create a new level. See step 1 below.
- Classes: If you already classified image objects, you can select classes to focus on the image objects of these classes. See step 2 below.
- Conditions: You can use a threshold condition to further narrow down the number of image objects. As part of the image object domain, you can use conditions to select a group of image objects for the classification. You can either use a condition alone for classification or to reduce the number of image objects for other classification methods. See step 3 below.

→ **Use Thresholds for Classification with Class Descriptions** on page 212

- Additionally, you can enter the maximum number of image objects to be considered. See step 4 below.

Technically, the image object domain is a set of image objects. Before a classification, the image object domain is a set of unclassified image objects. Every process loops through this set of image objects one by one and applies the algorithm to each single image object.

Find Out More

About Image Object Levels

Image objects are organized in one or multiple image object levels per image. An image object level serves as a layered working area of processes; here you commonly apply most algorithms.

An image object level can be created by segmentation algorithms from the underlying pixel level or from an existing image object level. Further you can create an image object level by duplicating an existing image object level.

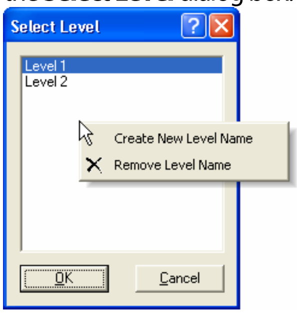
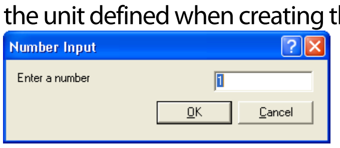
→ ***Algorithms Overview** on page 155*

Multiple image objects levels of an image perform an image object hierarchy.

→ ***Image Object Hierarchy*** *on page 26*

*Keep in mind the important difference of **image object level** and **image layer**: While an image object level serves as working area, the image layer represents the data base of of an image analysis.*

1. To specify the **Image Object Domain** ③, select an image object level or another general domain in the drop-down list box. The following general domains are available:

Domain	Usage
no image object	For algorithms that do not use pixels or image objects (e. g. **Execute child process** or **Update process variable**)
pixel level	Select to apply the algorithm on the pixel level (e. g. segmentation algorithms).
image object level (No Level selected) alternatively: *name of the currently active image object level*	Select to apply the algorithm on the currently active image object level. Click the **Parameters** ③ button to select another image object level in the **Select Level** dialog box.  Figure 105: Select Level dialog box. If the chosen algorithm creates a new image object level, you may want to avoid the automatic naming of new image object levels. In that case you can define a new name beforehand. It is assigned to the image object level during process execution. Right-click in the **Select Level** dialog box and select **Create New Level Name**. In the opening dialog box you can enter the name of the new image object level.
current image object	Select to apply the algorithm to the current internal selected image object of the parent process.
neighbor obj. (dist=0)	To apply the algorithm to all neighbors of the current internal selected image object of the parent process. The size of neighborhood is defined by the distance value, the spatial distance. Default value **0** adds all direct neighboring image objects. Press the **Parameters** ③ button to set the distance in the **Number Input** dialog box. The unit is pixel, regardless to the unit defined when creating the project.  Figure 106: Number Input dialog box.

→ **Use Hierarchical Image Object Levels** on page 193

→ **Edit an Image Object Level or Level Variable** on page 196

→ **Spatial Distance** on page 171

sub- objects (down=1)	Apply the algorithm to all subobjects of the current internal selected image object of the parent process. The number of levels down the image objects level hierarchy is defined by the down value. Press the **Parameters** 3 button to set the steps down in the **Number Input** dialog box.
super objects (up=1)	Apply the algorithm to the superobject of the current internal selected image object of the parent process. The number of levels up the image objects level hierarchy is defined by the up value. Press the **Parameters** 3 button to set the steps up in the **Number Input** dialog box.

→ Level Distance on page 171

→ Level Distance on page 171

2. To specify the **Image Object Domain**, you can select one or more classes for further narrowing down the image objects of interest.

   2.1 Click the broad **All Objects** 4 button (alternatively displaying the names of selected classes) to open the **Edit Classification Filter** dialog box.

→ Edit Classification Filter on page 443

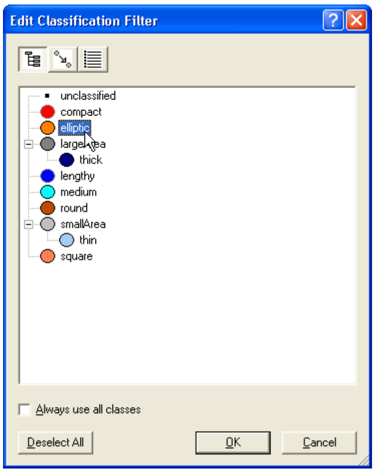


Figure 107: Edit Classification Filter dialog box.

   2.2 Select the domain the algorithm will be applied to:

   - unclassified image objects
   - one class
   - multiple classes. To select, click the **List** button. Hold the Ctrl key while selecting multiple classes with the cursor.

   2.3 The selected classes operate as a filter to the image object domain. After confirmation with **OK** the selected classes are displayed on the broad button 4. Default setting is to use all image objects of the domain.

3. To specify the **Image Object Domain**, you further can define a condition based on a feature. This allows you to limit the execution of the selected algorithm to only those image objects (of the selected classes (step 2) on the selected image objects or image object level (step 1)) that fulfill the condition.

   3.1 Click the broad **No Condition** 5 button (alternatively displaying the names of selected features) to open the **Select Single Feature** dialog box.

   3.2 Select a feature to filter the image object domain for a selected feature. After choosing a feature the **Edit Threshold Condition** dialog box opens where you can define a threshold condition.

→ Select Single Feature on page 459

→ Features Overview on page 168

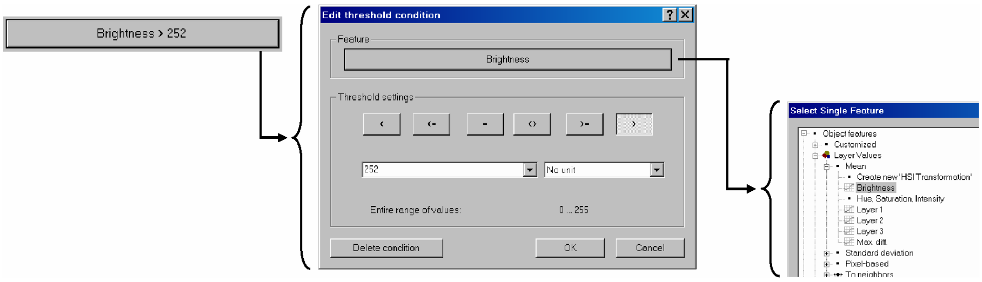


Figure 108: Click the broad buttons for modication of the items that are displayed on the button.

In the **Feature** group box, the feature that has been selected to define the threshold is displayed on a broad button. To select a different feature, click this button.

3.3 In the **Threshold settings** group box, select a logical operator. Enter the number defining the threshold. With **Definiens Developer**, you can alternatively select a process variable, if one exists. For constants, you can define the unit to be used. The feature range is displayed below.

→ About Operators on page 245

3.4 Click **OK** to close and return to the **Edit Process** dialog box. The defined feature threshold condition is now displayed on the broad button (5).

Tip

Defining a threshold conditions is available within class descriptions too.

→ ***Use Thresholds for Classification with Class Descriptions*** *on page 212*

4. To specify the **Image Object Domain**, you additionally can define a **Maximum number of image objects** (6).

Repeat Process Execution

Processes including all associated child processes can be re-executed.

Repeated process execution is called looping.

1. In the **Loops & Cycles** (7) group box you can set whether the process and all its child processes will be re-executed, called looping. Select the check box and the process will loop so long as any image objects are changed. Note, that changes may not only occur at the end of a loop because changes are defined as any modifications to variables (**Definiens Developer** only) or image objects. Therefore you should avoid such process settings that can cause endless looping or repeated execution with identical results.

2. Alternatively set a specific **Number of cycles** the process will loop.

Note

If **Loop while Something Changes** is enabled, or if the number of cycles in **Loops & Cycles** is greater than 1, the process has two cascading loops. The outer one repeats the whole process whereas the inner one automatically loops over all image objects of the domain and the child processes applying the algorithm to each object.

Name the Process

Define the name of a single process and edit additional information.

1. By default, **Names** 8 of processes are generated automatically based on the algorithm type, the image object domains and algorithm parameters. Anyhow, you can edit the process name.

> **Note**
>
> Editing the name of a process will disable automatic naming. To come back to automatic generation of process names, select the **Automatic** 8 checkbox again.

2. Add a comment to your process, which can be displayed in the **Process Tree** window. Click the **Comment** 9 button and edit it in the **Comment** dialog box.


Comment

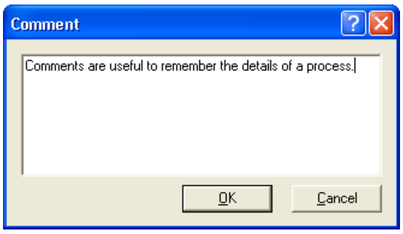


Figure 109: Comment dialog box

You can also see and add process comments without opening the process and view the comments in rule set documentation.

→ Document a Process Sequence on page 149

If you want to hide comments in the **Process Tree** window, you can change the settings in the **Process Editing** area of the **Options** dialog box.

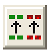
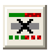
→ Options on page 452

7.3.2.2 Undo Process Editing Operations

Go back to a previous state in process editing by using the undo function. From there you can go forward by using the redo function. This allows you to go backward or forward in the operation's history.

You can undo or redo the creation, modification, and removal of:

- Processes
- Classes
- Customized features
- Variables

> **Note**
>
> The undo function does not include image object level operations such as  **Copy Current Level** or  **Delete Level**.

Note

If referenced rule set items such as a class or a variable are deleted and this action is undone, than only the object itself is restored but not its references.

1. To undo the last operation, select the **Undo Process Editing** button.
2. To redo the last undone operation, select the **Redo Process Editing** button.
2. To undo (or redo) multiple previous operations, click the drop-down list button right of the **Undo Process Editing** button or the **Redo Process Editing** button on the **Tools** toolbar. A list of operations is displayed below the buttons:

Undo Process Editing

Redo Process Editing

Undo drop-down list

Redo drop-down list

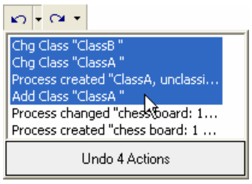


Figure 110: List of available operations below the Undo button.

- Locate the cursor over the **Undo Process Editing** button to select an earlier operation to which you would like to roll back. This enables you to go backwards in the operation's history.
- Locate the cursor over the **Redo Process Editing** button to select a later operation to redo. All intermediate operations will be undone. This enables you to go forward in the operation's history.

3. Toggle the previous operations until you are sure of your selections. With the next operation, the remaining redo operations—listed below the **Redo Process Editing** button—are deleted.

Options

The undo function is enabled by default in the **Options** dialog box. You can assign a minimum number of undo items, and a maximum of memory allowed for undo items. The minimum number of undo items has priority over the maximum memory setting.

→ Options on page 452

To minimize memory usage you can disable the undo function completely.

7.3.2.3 Modify a Single Process

When developing rule set you may want to modify single processes allowing you to improve the results of image analysis.

1. Go to the **Process Tree** window and select an process on the process tree.

Process Tree

2. To open the **Edit Process** dialog box, do one of the following:
   - Right-click the process and select **Edit Process** from the context menu.
   - Choose **Process** > **Process Commands** > **Edit Process** on the main menu bar.
   - Type Ctrl + Enter .

   The **Edit Process** dialog box opens.
3. Change the settings. For details see the **Create a Single Process** section.

→ Create a Single Process on page 132

7.3.2.4 Execute a Single Process

Executing a single process helps you testing a process while developing rule sets. The execution of a single process includes all existing child processes.

Precondition: Delete existing image object levels that are related to the process.

> **Note**
>
> When testing and modifying processes, you commonly want to execute a single process that has already been executed. Before you can do this, you have to delete the related existing image object levels. In most cases, you have to delete all existing image object levels and execute the whole process sequence from the beginning.
>
> To delete image object levels, choose **Image Objects** > **Delete Levels** on the main menu bar.
>
> If the execution of a process is not possible at the selected position in the process tree, the **Message Console** displays a warning message.
>
> → **Delete Image Object Level** on page 197

→ **Execute Processes for Testing** on page 151

To execute a single process do one of the following:

- Right-click a process and choose **Execute** from the context menu.
- Choose **Process** > **Process Commands** > **Execute** on the main menu bar.
- Select the process in the Process Tree window and press F5 on the keyboard.
- If the **Edit Process** dialog box is open, click **Execute**.

> **Note**
>
> While the execution of a single process includes all existing child processes, a process also can be executed on a selected image object only. To do this, choose **Process Menu** > **Process Commands** > **Execute on Selected Object** from the main menu bar.

7.3.3 Organize a Process Sequence

Organize a process sequence consisting of single processes that are sequenced and grouped together in a hierarchical structure to perform as rule set an image analysis routine.

Each process can contain any number of sub processes. This allows the grouping of processes into functional units. Furthermore, this functionality allows generating complex workflows, restricting child processes to certain domains or tying child processes to conditions.

Processes are organized in the **Process Tree** window Here you can arrange, save, load and execute them. The **Process Tree** window provides the development environment. The process tree defines the structure and the flow control of the image analysis.

To open the **Process Tree** window do one of the following:

- Choose **Process** > **Process Tree** on the main menu bar.

Process Tree

- Choose **View > Process Tree** on the main menu bar.

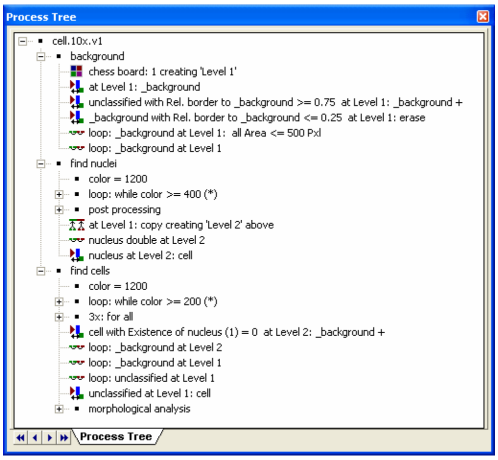


Figure 111: Process Tree window.

7.3.3.1 Organizational Processes

As single processes are the elementary units of rule sets, they have to be organized to build a solution to a specific image analysis problem. You can organize single processes in different structures:

- Sequence: Append processes to build a sequence.
- Group: Insert processes on a lower hierarchical level to build a group.
- Loop: Re-execute single processes or groups to loop several times. For details about Looping see the **Repeat Process Execution** section.

→ Repeat Process Execution on page 138

1. To organize processes go to the **Process Tree** window.

 Process Tree

2. Right-click a process in the **Process Tree** window. From the context menu you can choose **Append New** to arrange them sequential. This will build a successive step.

   Alternatively you can choose **Insert Child** to start a new group on a lower hierarchical level. This will build a subordinate child process.

   The **Edit Process** dialog box opens.

3. Edit the process.

   When inserting a child process, we recommend that you begin the new process group with an process using the algorithm **Execute child process**. This allows you easy handling and labeling of child processes.

→ Create a Single Process on page 132

4. In the process tree structure, an appended process is added below; an inserted process is added indented below.

Find Out More

About Grouping Processes in the Process Tree

Single processes are arranged in a process tree. The order in the process tree represents the sequence of processing. That means that sequenced processes are processed from top down.

The structure of the process tree is comparable to a family. According to the connection mode a process can be:
- a ***parent process*** *on a higher hierarchy level.*
- an inserted ***child process*** *on a lower hierarchy level*
- an appended ***sibling process*** *on the same hierarchy level.*
Similar to family members, any process can be a parent process, child process and sibling process at the same time. See figure below.

To give the process tree a clear structure, you should arrange processes in groups.

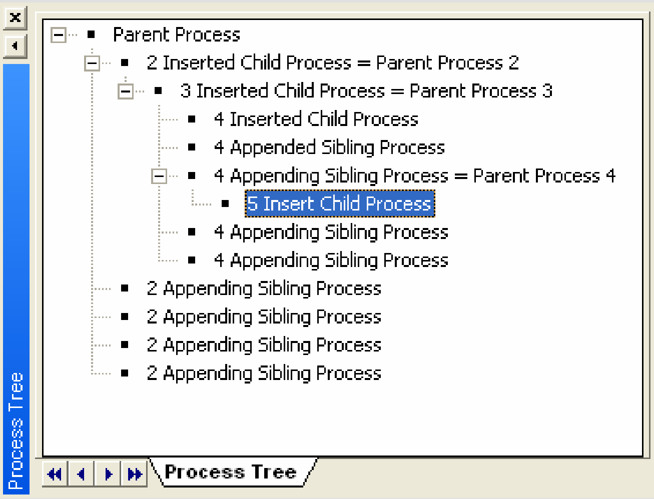


Figure 112: Process Tree displaying a prototype of a process tree. The processes are named according to their connection mode.

The ***Insert Child*** *context menu item includes a new process subordinate to the selected process. Inserted child processes will be executed as a part of its superordinated parent process, if the algorithm of the parent process supports the execution of child processes. The typical algorithm of the parent process is* ***Execute child processes****.*

After creating a new parent process, you always add the first subordinate process with the menu item ***Insert Child.***

Unlike ***Insert Child****, the menu item* ***Append New*** *adds a new sibling process in the same hierarchy level after the selected process.*

7.3.3.2 Drag-and-drop in the Process Tree

Use drag-and-drop operations in the **Process Tree** dialog box to modify process structures.

Thereby you have to bear in mind that a process can be connected to the process tree in three modes. According to the connection mode a process can be:

- a parent process on a higher hierarchy level.
- an inserted child process on a lower hierarchy level
- an appended sibling process on the same hierarchy level

1. Go to the **Process Tree** dialog box.
2. Left-click a process and drag and drop it to another process. It will be appended as a sibling process on the same level as the target process.

3. Right-click a process and drag and drop it to another process. It will be inserted as a child process on a lower level than the target process.

7.3.3.3 Load a Process

Load an existing processes from file to a currently open project.

When loading processes, you can choose between two methods for handling loaded rule set items (which are classes, variables and customized features) of the same name:

- Automatic renaming by **Load and Append Rule Set.**
- Automatic adaptation of properties by **Load Rule Set.**

Instead of appending a loaded process, you can choose to merge it.

Typically, you will use **Load Rule Set** when developing rule sets and you want to perform some manual editing after processing. To continue the processing after manual editing, you load and merge the following processes, for example, for exporting.

→ Manual Editing on page 391

In addition, the difference between the **Load Rule Set** and the **Load and Append Rule Set** is relevant if the following conditions are fulfilled:

- There are rule set items with the same name in both the existing processes and the process to load.
- The properties of rule set items of the same name are different.

In those cases, the properties of loaded rule set items of the same name will be adapted to the existing ones.

> **Note**
>
> If the properties of the loaded rule set items of the same name are different from the existing ones, they are changed automatically.

Load and Merge a Rule Set

1. To load and merge a process, do one of the following:

- Go to the **Process Tree** window and right-click a process and select **Load Rule Set** from the context menu.


Process Tree

- Choose **Process > Load Rule Set** on the main menu bar.

  The **Load Process** dialog box opens.

2. Browse to a **.dcp** file containing a process and **Open** it.

   If there are conflicts in the aliases of the image layers, the **Assign Image Layer Alias** dialog box opens. For details see the **Assign Image Layer Alias** section.

→ Assign Image Layer Alias on page 433

> **Note**
>
> To avoid loading conflicts concerning the aliases of the image layers you can edit them. Choose **Process > Edit Aliases > Image Layers Aliases** in the main menu bar.

Likewise you can avoid loading conflicts concerning the aliases of thematic layers. To edit them, choose **Process** > **Edit Aliases** > **Thematic Layers Aliases** in the main menu bar.

→ Manage Layer Aliases on page 450

3. The process will be added to the process tree.

   You can choose whether to automatically save your rule sets when you close a project by using the **Keep Ruleset on closing Project** option in the **Options** dialog box.

→ Options on page 452

Load and Append Rule Set

1. To load and append a rule set, choose **Process** > **Load and Append Rule Set** on the main menu bar. The **Load Process** dialog box opens.
2. Browse to a **.dcp** file containing a process and **Open** it.

   In cases of loading conflicts concerning the names of the image layers, the **Edit Image Layer Alias** dialog box opens. For details see the **Edit Image Layer Alias** section.

→ Assign Image Layer Alias on page 433

3. The rule set will be added to the process tree.

A process file or **.dcp** file contains processes and different rule set items to which the processes refer. These rule set items are classes, variables and customized features.

Usually, the loaded process does not includes any rule set items that have the same name as one of the existing rule set items in the **Process Tree** window. However, if there exist any rule set items with the same name in the existing and the loaded processes, the loaded rule set items are renamed. A number will be added to the name of the loaded rule set item allowing to distinguish them from the existing ones. In this case a message informs you about automatic renaming actions.

7.3.3.4 Save a Process or Rule Set

Save processes to file for reuse. You can save a single process or the whole rule set.

Process files have the extension **.dcp**.

Save a Process

To save a process, right-click it in the **Process Tree** window and select **Save As** from the context menu. The **Save Process** dialog box will open to enable you to save the process to a file.

Save a Rule Set

To save an entire rule set, including the structure, under a new name do one of the following:

- Select **Save Rule Set** in the context menu of the **Process Tree** window. The **Save Process...** dialog box will open to enable you to save the rule set to a file
- Open the **Process** menu in the main menu bar and select **Save Rule Set.** The **Save Process** dialog box will open to enable you to save the rule set to a file.

  You can use the **Save rule set minimized** option in the **Options** dialog box to choose whether to save features with the rule set.

→ Options on page 452

7.3.3.5 Duplicate Process

Duplicate a process including all child processes for building your process structure and for comparing processes with different settings

Process Tree

1. To duplicate a process, go to the **Process Tree** window and do one of the following:
   - In the **Process Tree** window, right-click a process and select **Copy** from the context menu.
   - Choose **Process > Process Commands > Copy** on the main menu bar.
1. To paste a process, right-click on a process in the **Process Tree** window or on a process snippet in the **Snippets** window and select **Paste** in the context menu.
2. The new process will be appended to any selected process or simply added to an empty window.

7.3.3.6 Delete a Process or Rule Set

Delete a single process or the whole rule set.

Delete a Single Process

Process Tree

1. To delete a single process from the project, go to the **Process Tree** window and do one of the following:
   - Right-click a process and select **Delete** from the context menu.
   - Choose **Process > Process Commands > Delete** on the main menu bar.
2. The selected single process will be deleted without reconfirmation if the selected process has no child processes.

Delete a Complete Rule Set

1. To eliminate the whole processes sequence including structure, classes, variables (**Definiens Developer** only) and customized features from the project, go to the **Process Tree** window and do one of the following:
   - Right-click a process and select **Delete Rule Set** from the context menu.
   - Choose **Process > Delete Rule Set** on the main menu bar.

Caution

Delete Rule Set eliminates all classes, variables (**Definiens Developer** only) and customized features, in addition to all single processes,

2. The rule set will be deleted after reconfirmation.

7.3.3.7 Find and Replace Rule Set Items

Within a rule set, you can search for all occurrences of one of the following rule set items:

- Feature
- Feature variable
- Class
- Class variable
- Image object level
- Level variable
- Scene variable
- Object variable
- Image layer
- Thematic layer

You can browse and edit found rule set items or replace them by rule set items of the same category. This is helpful especially for maintaining large rule sets and for development in teams.

1. To open the **Find and Replace** window, do one of the following:
    - Choose **Process** > **Find and Replace** on the main menu bar.
    - Choose **View** > **Window** > **Find and Replace** on the main menu bar.
    - Select and right-click a class within the **Class Hierarchy** window and choose **Find Class** from the context menu.
    - Select and right-click a feature within the **Image Object Information** window and choose **Find** from the context menu.
    - Select and right-click a feature within the **Feature View** window and choose **Find** from the context menu.

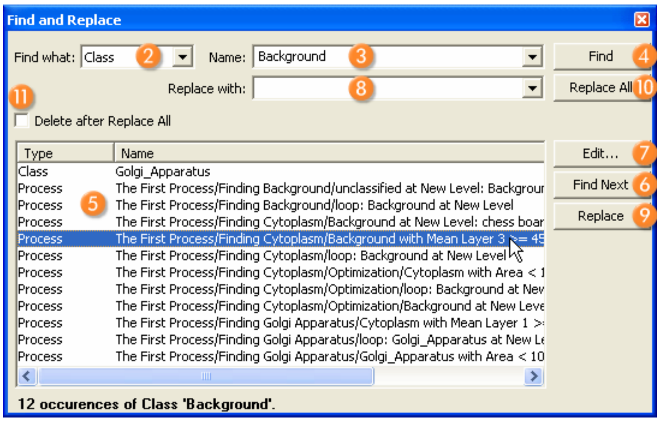


Figure 113: Find and Replace window with a sample search of the class Background.

2. In the **Find what** drop-down list box 2, select a category of rule set items you want to find.
3. In the **Name** combo box 3 type or select a rule set item within the category.

4. Click the **Find** button 4 to list the found rule set items.

5. Now you can select a found rule set item in the list 5. The corresponding processes are highlighted in the Process Tree window.

6. Click the **View Next** button 6 to browse the found rule set items.

7. To edit a rule set item, double-click it or select it and click the **Edit** button 7. The appropriate editing dialog opens, for example, the **Edit Process** dialog box for a found process.

8. For replacing rule set items as selected in the name combo box, go to the **Replace with** combo box 8. Type or select a rule set item within their category to replace with.

9. To replace a single rule set item, select it and click the **Replace** button 9. The list is updated.

10. To replace all rule set items, select it and click the **Replace All** button 10. The list is updated.

11. Select the **Delete after Replace All** check box 11 if you want to delete the replaced rule set item as selected in the **Replace with** combo box 8.

7.3.3.8 Save and Reuse Process Snippets

A process snippet is a part of a rule set, consisting of one or a sequence of processes. You can organize and save process snippets for reuse in other rule sets.

Drag-and-drop functionality makes it easy to copy them between the **Process Tree** window and the **Snippets** window.

To reuse snippets in other rule sets, you can export and save them in a snippets library.

1. To create snippets, first open the **Snippets** window by doing one of the following:

   - Choose **View** > **Windows** > **Snippets** on the main menu bar.

   - Choose **Process** > **Snippets** on the main menu bar.

Rule Set Snippets

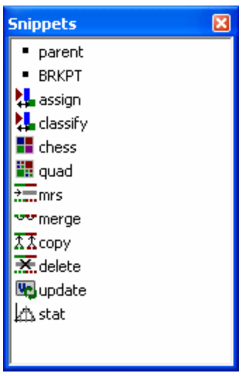


Figure 114: Snippets window.

2. By default, the **Snippets** window displays the following frequently-used algorithms that you can drag into the **Process Tree** window:

→ Reference Handbook

   - **parent:** execute child processes

   - **BRKPT:** add a breakpoint (not an algorithm)

- **assign:** assign class
- **classify:** classification
- **chess:** chessboard segmentation
- **quad:** quadtree based segmentation
- **mrs:** multiresolution segmentation
- **merge:** merge region
- **copy:** copy image object level
- **delete:** delete image object level
- **update:** update variable
- **stat:** compute statistical value

3. Drag a process of your choice from the **Process Tree** window to the **Snippets** window. You can drag any portion of the process tree along with its child processes, if any. Alternatively, you can right-click process or snippets for copying and pasting.
4. You can also copy snippets from the **Snippets** window to any position of the **Process Tree** window.

Note

You cannot add customized algorithms to the **Snippets** window, but snippets can include references to customized algorithms.

5. To save all listed snippets in a snippets library, right click in the **Snippets** window and select **Export Snippets**. All process snippets are saved as a snippet **.slb** file.
6. To import Snippets from a snippets library, right-click in the **Snippets** window and select **Import Snippets**.

Options

7. Rename the processes in the **Snippets** window by single clicking twice on the name and entering a new name. However, when you paste it back in the **Process Tree** window it reverts to the original name.
8. The contents of the **Snippets** window remain until deleted. To delete, right click on a process snippet and select **Delete or Delete All** on the context menu.

7.3.3.9 Document a Process Sequence

Insert comments in a process sequence to use in rule set documentation.

→ Document a Rule Set on page 153

You can use the **Rule Set Documentation** window to view and store all the processes in your rule set including your comments.

1. To document a rule set item (such as a process, class or expression) select it in the window where the rule set item displays, which is one of the following:

- **Process Tree** window
- **Class Hierarchy** window
- **Class Description** window.

2. Move your mouse over the lower right corner of the window to display the **Comment Window** button.


Comment Window

3. Click the button to open the comment editing field and add a comment. The editing field is not available unless you have selected a rule set item. You can resize the window if necessary by dragging the borders.
Your comment is automatically added to the rule set item as soon as you select another rule set item or another window.
4. You can also paste text into the editing area from the clipboard by using the context menu. Right-click at the insertion point and select **Paste** from the context menu.
5. To move up or down in the list of rule set items, click the up and down arrow buttons next to the editing field. This enables you to add comments without opening the rule set items for editing.

6. To remove a comment, use the **Undo** button or delete the comment from the comment editing field.

→ Undo Process Editing Operations on page 139

7. To view, store and copy your comments along with classes, processes and customized features, use the **Rule Set Documentation** window.

→ Document a Rule Set on page 153

Tip

*You can also open the **Edit Process** window for any process and enter a comment using the **Comment** dialog box. These comments will display in the **Rule Set Documentation** window also.*

→ *Name the Process on page 139*

7.3.3.10 Encrypt and Decrypt Rule Sets

Encrypt your rule sets to prevent reading and modification by others.

Encrypt a Rule Set

1. To encrypt rule set, first load it to the **Process Tree** window.
2. Open the **Process** menu in the main menu bar and select **Encrypt Rule Set** to open the **Encrypt Data** dialog box.
3. Do one or both of the following to encrypt the rule set and set a decryption code:
   - Enter a password that you will use to decrypt the rule set, and confirm it.
   - Enter a license ID.
     This option enables rule set developers to limit use of the rule set to specific types of Definiens licensees.

   The rule set will display only the parent process, with a lock icon next to it. If you have more than one parent process at the top level, each of them will have a lock next to it. You will not be able to open the rule set to read or modify it, but you can append more processes to it and they can be encrypted separately if you wish.

> **Note**
>
> If the rule set is part of a project and you close the project without saving changes, the rule set will be decrypted again when you reopen the project.

Decrypt a Rule Set

1. To decrypt a rule set, first load it to the **Process Tree** window.
2. Open the **Process** menu in the main menu bar and select **Decrypt Rule Set** to open the **Decrypt Data** dialog box.
3. Enter the password that was used to encrypt the rule set. The lock icon will disappear and you will be able to read and modify the processes.

7.3.4 Execute Processes for Testing

While developing rule sets, you should execute single processes or process groups for testing the effects.

→ About the Development Workflow on page 129
→ Execute a Single Process on page 141

Depending on the selected process, you can execute one of the following:

- A single process.
- A process group which is a process including all child process.
- A whole rule set, commonly organized below the highest process in the **Process Tree** window.
- Defined sections of processes.

In addition, you can execute a process on one selected image object only instead of the whole image object domain.

Process Tree

1. To execute a process, go to the **Process Tree** window and select a process.
2. Do one of the following:
   - Right-click the process and select **Execute** from the context menu.
   - Choose **Process** > **Process Commands** > **Execute** on the main menu bar.
   - Press F5 on the keyboard.

   The process and all children will be executed.

7.3.4.1 Exclude Process from Execution

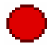
For testing purposes, you can exclude some child processes from execution.

- You can deactivate processes to exclude them from processing.
- You can set a breakpoint to stop the processing of the subsequent processes.

1. To exclude child processes from execution, go to the **Process Tree** window and select a process.
2. Go to one of these process commands:

- Right-click the process to display the context menu.
- Choose **Process > Process Commands > ...** on the main menu bar.

3. If **Active** is checked on the context menu or in the **Process Commands** menu, the process is active. Click **Active** to inactivate the process. The process will be displayed in brackets. Click again to reactivate.
4. Click **Breakpoint** on the context menu to set a breakpoint to the process. Processing will stop at the selected process. Click again to remove the breakpoint.



Breakpoint

7.3.4.2 Update the Project View While Processing

For tracking the execution of multiple process, you can update the project view.

1. To update the project view while processing, go to the **Process Tree** window and select a process.
2. Go to the process commands:
   - Right-click the process to display the context menu.
   - Choose **Process > Process Commands > ...** on the main menu bar.
3. Click **Update View** to turn on updating. Click again to turn off.

7.3.4.3 Execute a Process on a Selected Image Object

Typically you will execute processes on the image object domain as defined in the **Edit Process** dialog box. Alternatively you can execute on a single image object:

1. Select an image object
2. Right-click a process and choose **Execute on Selected Object** from the context menu.
3. The process including all child process will be executed on the one image object only.

7.3.4.4 Improve the Processing Performance

To control the processing performance you can use the **Process Profiler** to determine the processes that slow down the execution of your rule set. You can then modify your processes or replace them with alternative, less time-consuming processes, thus eliminating the performance bottlenecks. Remember to test your rule set again to see if the new results are still satisfactory.

To open the **Process Profiler** window, choose one of the following from the main menu bar:

- **View > Process Profiler**
- **Process > Process Profiler**

Process Profiler

Execute a process and view the profiling results under the **Report** tab.

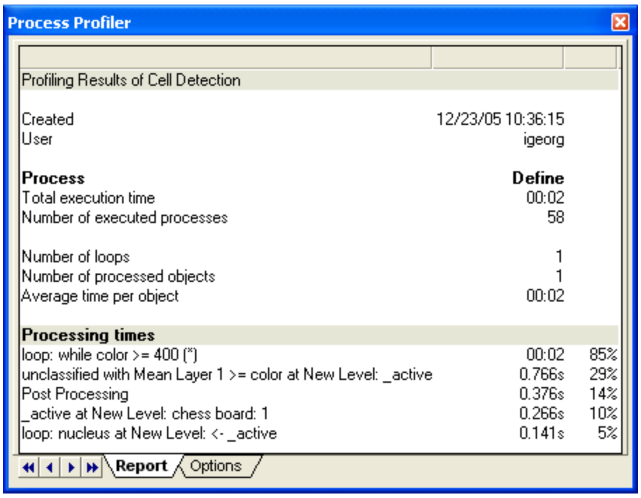


Figure 115: The Process Profiler window.

By default, the slowest five processes are shown in the **Report**. Under the **Options** tab, you can change the profiling settings.

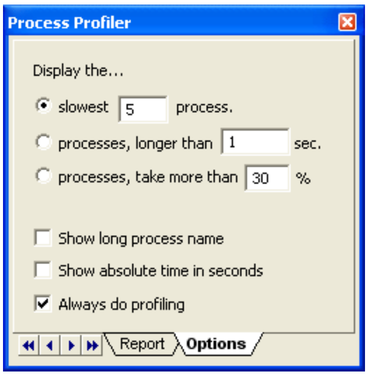


Figure 116: The Options tab of the Process Profiler window.

7.3.5 Document a Rule Set

Use the **Rule Set Documentation** window to view, store, and edit documentation for your rule set.

1. To open the **Rule Set Documentation** window, choose one of the following in the main menu bar:
   - **Process** > **Rule Set Documentation**
   - **View** > **Windows** > **Rule Set Documentation**
2. Click the **Generate** button to display a list of rule set items in the window, including classes, customized features and processes.The window also displays your comments to classes, class expressions, customized features and processes. Each comment is set off by a double backslash.
3. You can add comments to rule set items and then click the **Generate** button again to view your new comments.
4. If you wish, you can edit the documentation directly in the **Rule Set Documentation** window.
   Your changes in the window will not be added to the rule set. Your changes are

→ Document a Process Sequence on page 149
→ Document a Class on page 243
→ Document an Expression on page 249

deleted when you click the **Generate** button but they are preserved if you **Save to File** or **Copy to Clipboard.**

5. To save the contents of the **Rule Set Documentation** window to a file, click the **Save to File** button and choose a location and file type (either text or Rich Text Format).
6. To copy the contents of the **Rule Set Documentation** window to the clipboard, click the **Copy to Clipboard** button. Then you can paste the contents into a text editor.
7. To copy only part of the contents of the **Rule Set Documentation** window to the clipboard, highlight the text, right-click and select **Copy** from the context menu.
8. To paste text from the clipboard into the **Rule Set Documentation** window, right-click at the insertion point and select **Paste** from the context menu.

7.3.6 Process Tree Context Menu Options

Right click in the **Process Tree** window to select on the context menu:

Commands	Usage	
Edit	Open the **Edit Process** dialog box. Edit the selected process.	→ Edit a Single Process on page 131
Execute	Execute the selected process.	→ Execute a Single Process on page 141
Execute on Selected Object	Execute the process on the selected object only.	→ Execute a Single Process on page 141
Append New	Open the **Edit Process** dialog box. Insert a new sibling process following the selected process.	→ Create a Single Process on page 132
Insert Child	Open the **Edit Process** dialog box. Insert a new process as a child process of the selected process.	→ Create a Single Process on page 132
Copy	Copy the selected process and any child processes.	→ Duplicate Process on page 146
Paste	Paste a copy of a process along with any child processes into another position in the rule set, or into the **Snippets** window	→ Duplicate Process on page 146
Load Rule Set	Open the **Load Process** dialog box. Browse to a rule set (***.dcp** file) and append it at the end of the rule set (if any) in the **Process Tree** window.	→ Load a Process on page 144
Save Rule Set	Open the **Save Process** dialog box. Save the entire rule to file.	→ Save a Process or Rule Set on page 145
Delete Rule Set	Delete the entire rule set.	→ Delete a Process or Rule Set on page 146
Save As	Open the **Save Process** dialog box. Save the selected process to a new file.	→ Save a Process or Rule Set on page 145
Delete	Delete the selected process and any child processes from the **Process Tree** window	→ Delete a Process or Rule Set on page 146

Create Customized Algorithm	Open the **Customized Algorithm Properties** dialog box. Create a customized algorithm based on a process sequence.	→ Create a Customized Algorithm on page 278
Edit Customized Algorithm	Open the Customized Algorithm Properties dialog box. Available only if there is a customized algorithm in the **Process Tree** window.	→ Modify a Customized Algorithm on page 281
Edit Customized Algorithm Default Parmeters	Open the **Default Parameters** dialog box. Available only if there is a customized algorithm in the **Process Tree** window.	→ Create a Customized Algorithm on page 278
Active	Set to inactive to exclude a process from execution.	→ Execute Processes for Testing on page 151
Breakpoint F9	Insert a breakpoint for the selected process. Stop the execution of processes at this process.	→ Execute Processes for Testing on page 151
Update View	Update the project view while processing.	→ Execute Processes for Testing on page 151

7.3.7 Algorithms Overview

Definiens Developer provide numerous algorithms useful in image analysis. In this section you will find a brief overview of major subdivisions of algorithms. For detailed information have a look at the **Reference Handbook**.

→ Reference Handbook

Process Related Operations Algorithms

The Process Related Operation algorithms are used to control other processes.

Segmentation Algorithms

Segmentation algorithms are used to subdivide the entire image represented by the pixel level domain or specific image objects from other domains into smaller image objects.

Definiens provides several different approaches to this well known problem ranging from very simple algorithms like chessboard and quadtree based segmentation to highly sophisticated methods like multiresolution segmentation or the contrast filter segmentation.

Segmentation algorithms are required whenever you want to create new image objects levels based on the image layer information. But they are also a very valuable tool to refine existing image objects by subdividing them into smaller pieces for a more detailed analysis.

Basic Classification Algorithms

Classification algorithms analyze image objects according defined criteria and assign them each to a class that best meets the defined criteria.

Advanced Classification Algorithms

Advanced classification algorithms classify image objects that fulfill special criteria like being enclosed by another image object or being the smallest or the largest object in a hole set of object.

Variable Operation Algorithms

Variable operation algorithms are used to modify the values of variables. They provide different methods to perform computations based on existing variables and image object features and store the result within a variable.

Reshaping Algorithms

Reshaping algorithms modify the shape of existing image objects. They execute operations like merging image objects, splitting them into their subobjects and also sophisticated algorithm supporting a variety of complex object shape transformations.

Level Operation Algorithms

Level operation algorithms allow you to add, remove or rename2 entire image object levels within the image object hierarchy.

Interactive Operation Algorithms

Interactive operation algorithms are used for user interaction with the user of actions in **Definiens Architect**.

Vectorization Algorithms

> ***Tip***
>
> *Vectorization algorithms available in earlier versions have been removed because polygons are available automatically for any segmented image. You can use the algorithm parameters in the* ***set rule set options*** *algorithm to change the way polygons are formed.*

Sample Operation Algorithms

Use sample operation algorithms to perform sample operations.

Image Layer Operation Algorithms

Image layer operation algorithms are used to create or to delete image object layers. Further you can use the image layer operation algorithms to apply filters to image layers at the pixel level.

Thematic Layer Operation Algorithms

Thematic layer operation algorithms are used to transfer data from thematic layers to image objects and vice versa.

Export Algorithms

Export algorithms are used to export table data, vector data and images derived from the image analysis results.

Workspace Automation Algorithms

Workspace automation algorithms are used for working with subroutines of rule sets. These algorithms enable you to automate and accelerate the processing of workspaces with especially of large images. Workspace automation algorithms enable multi-scale workflows, which integrate analysis of images at different scales, magnifications, or resolutions.

Customized Algorithms

Customized algorithms enable you to reuse process sequences several times in one or different rule sets. Based on a developed process sequence, representing the developed code, you can create and reuse your own customized algorithms.

→ Reuse Process Sequences with Customized Algorithms on page 276

In contrast to duplicating a process, the main advantage of creating customized algorithms, is that when you want to modify the duplicated process you need to perform the changes to each instance of this process. However, with customized algorithms you only need to modify the customized algorithm template and the changes will take effect to every instance of this algorithm.

7.4 Create Image Objects by Segmentation

The fundamental step of any **Definiens** image analysis is a segmentation of a scene—representing an image—into image object primitives. Thus, initial segmentation is the subdivision of an image into separated regions represented by basic unclassified image objects called image object primitives.

→ Scene on page 22

For successful image analysis, defining object primitives of suitable size and shape is of utmost importance. As a rule of thumb, good object primitives are as large as possible, yet small enough to be used as building blocks for the objects to be detected in the image. Pixels are the smallest possible building block, however pixels have limited information. To get larger building blocks, different segmentation methods are available to form contiguous clusters of pixels that have larger property space.

Commonly, in image processing, segmentation is the subdivision of a digital image into smaller partitions according to given criteria. In contrast, within the **Definiens** technology, each operation that creates new image objects is called segmentation, no matter if the change is achieved by subdividing or by merging existing objects.

Different segmentation algorithms provide several methods of creating of image object primitives.

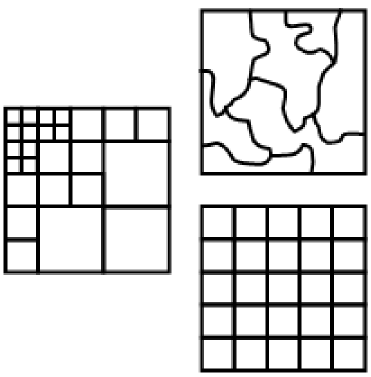


Figure 117: Different segmentation results: Multiresolution (above), Quadtree (left), Chessboard (below).

Segmentation Creates a New Image Object Level

The new image objects created by segmentation are stored in what is called a new image object level. Each image object is defined by a contiguous set of pixels, where each pixel belongs to exactly one image object. Each of the subsequent image object related operations like classification, reshaping, re-segmentation, and information extraction is done within an image object level. Simply said, image object levels serve as internal working areas of the image analysis.

→ Image Object Level on page 24

7.4.1 About Image Object Primitives

As the result of a segmentation, object primitives are created. They are unclassified basic image objects. They serve as both information carrier and building blocks at the starting point of the image analysis.

Information Carrier

Image object primitives contain information about their spectral characteristics, their shapes, their positions and textures, as well as information about their neighborhoods. Thus, they are an essential prerequisite for the subsequent image analysis steps.

Building Blocks

Image object primitives serve as the building blocks used in the steps of any image analysis. As building blocks, they can be merged, cut, or classified. As the image analysis progresses, image object primitives commonly are merged to build larger image objects. Ideally, they do not need to be split anymore into smaller segments.

7.4.2 What Are Useful Image Object Primitives?

Image object primitives serve as information carriers and as building blocks for all further image analysis. Therefore, when starting an image analysis, it is important to invest enough time on testing and improving the segmentation to be used.

A convenient approach for segmentation of new projects is to run different segmentations with different parameters until the result is satisfying.

If the data sets are too large for easy handling, try to work with a representative subset to speed up the process. Once suitable segmentation parameters have been established, they should be tested on the whole data set.

Shape and size are both quality criteria of image object primitives.

→ Define a Scene Subset on page 371

Rules of Thumb for the Generation of Image Object Primitives

A. The average size of image object primitives should be as large possible, yet small enough show the contours of the structures that interest you. Keep in mind that too-small image object primitives can be fused after classification.

B. The desired contours should be represented with the lowest possible number of image object primitives. If the structures of interest are represented by fragments rather than by the forms of the entire structures, there will be more object primitives than necessary.

To assess the quality of a segmentation result, switch between **Pixel View** and **Object Mean** viewing options while observing the abstraction effect of object primitives. If a lot of the original image information gets lost or is changed, the image object primitives do not fulfill their roles sufficiently as defined above .

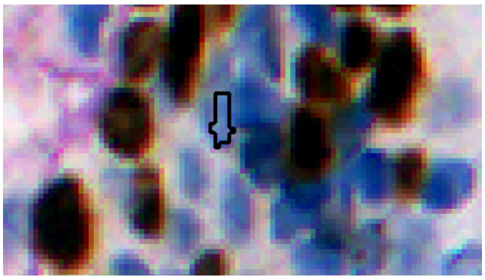
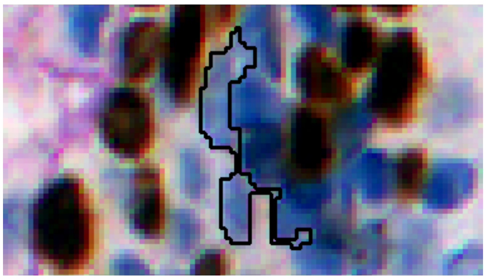
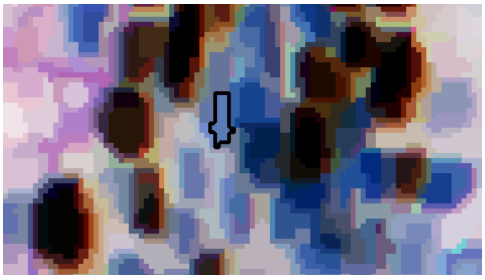
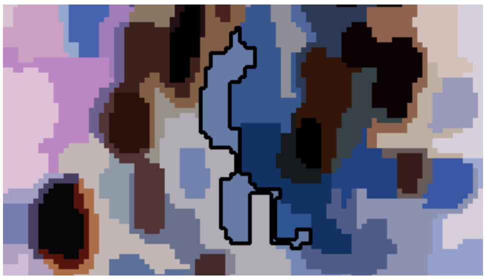
Useful Image Object Primitives	**Poor Image Object Primitives**
Below: Example of a segmentation resulting in useful image object primitives. 	Below: Example of a segmentation resulting in image object primitives that are too large. 
Switch between project view settings to examine the quality of a segmentation. Select single image objects for detailed investigation.	Switch between project view settings to examine the quality of a segmentation. Select single image objects for detailed investigation.
 Above: Useful image object primitives: objects of interest like nuclei can be recognized.	 Above: Image object primitives that are to large: some objects of interest are not recognized, that means some information is lost.

Image Data Courtesy of Novartis Pharma AG, Pathology/Toxicology EU, E. Persohn.

Pixel View (button pressed)

⇕ Project view settings

Object Mean View (button released)

7.4.3 Segmentation Principles: Top-down and Bottom-up

Commonly, the term segmentation means subdividing entities, for example objects, into smaller partitions. In image processing, the term **segmentation** is used different: Here, a segmentation is each operation that creates new image objects or alters the morphology of existing image objects according to given criteria. That means, a segmentation can be a subdividing operation, or a merging operation, or a reshaping operation.

There are two basic segmentation principles used:

→ When to Use Which Segmentation? on page 337

- Cut something big into smaller pieces, which is a top-down strategy

or

- Merge small pieces to get something bigger, which is a bottom-up strategy.

Top-down Segmentation

Top-down segmentation means cutting objects into smaller objects. It can—but does not have to—start from the entire image as one object.

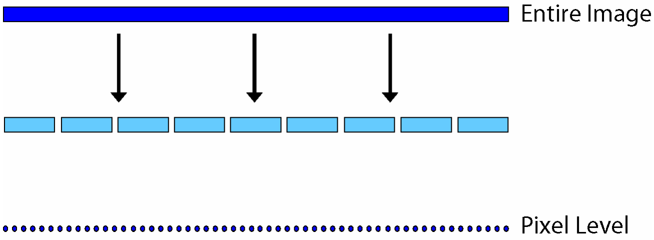


Figure 118: Top-down segmentation.

Chessboard segmentation and the quadtree segmentation are two examples.

→ Chessboard Segmentation on page 160
→ Quad Tree Based Segmentation on page 161

Bottom-up Segmentation

Bottom-up segmentation means assembling objects to create a larger objects. It can—but does not have to—start with the pixels of the image.

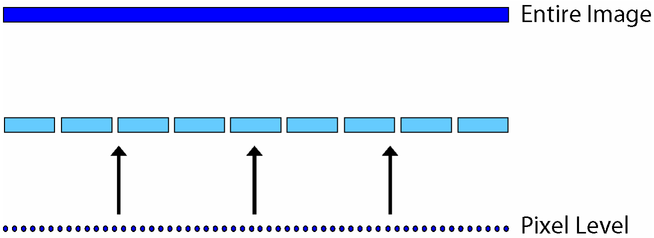


Figure 119: Bottom-up segmentation.

Examples are the multiresolution segmentation and the classification based segmentation.

→ Multiresolution Segmentation on page 162
→ Segmentation by Reshaping Algorithms on page 165

7.4.3.1 Chessboard Segmentation

Chessboard segmentation is the simplest segmentation algorithm. It cuts the scene, or later the dedicated image objects, into equal squares of a given size. Thus it is a top-down segmentation algorithm.

→ Reference Book

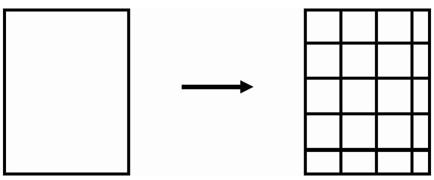


Figure 120: Chessboard segmentation.

Because the chessboard segmentation algorithm produces simple square objects, its main usage focuses on the cutting of images and image objects.

The following are some typical uses.

- Tiling and Stitching:
  Images that are too large to be handled in one piece are cut into equal squares; each square is then treated individually. Afterwards, the tile results are stitched

→ Tile and Stitch Large Scenes on page 375

together.

- Refining relatively small image objects that have been found already:
Apply a chessboard segmentation with a small square-size parameter to those image objects for further detailed analysis.

- Applying a new segmentation to an image object:
Let us say you have an image object that you want to cut into multiresolution-like image object primitives. Because multiresolution segmentation is a bottom-up technique, you must first apply chessboard segmentation with a small square size, such as 1 (one). Then you can use those square image objects as starting image objects for a multiresolution segmentation.

You can use the **Edit Process** dialog box to define the size of squares.

→ Create a Single Process on page 132

Square Size 1

Use square size 1 (one) to generate pixel-sized image objects. The effect is that for each pixel you can investigate all information available from features.

→ About Features as a Source of Information on page 168

Medium Square Size

In cases where the image scale (resolution/magnification) is higher than necessary to find regions or objects of interest, you can use a square size of 2 or 4 to reduce the scale.

Use a square size of about one-twentieth to one-fiftieth of the scene width for a rough detection of large objects or regions of interest. You may perform such a detection at the beginning of an image analysis procedure.

Large Square Size

You can select a large square size, for example, one-fifth of the scene width, if scenes are very large. That way you cut it into smaller tiles for further analysis on selected tiles only. Analysis of each tile separately enables a better performance compared with processing the complete scene.

7.4.3.2 Quad Tree Based Segmentation

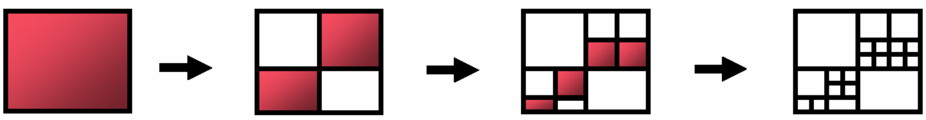
The quad tree based segmentation algorithm creates squares of differing sizes by cutting. Thus it is a top-down segmentation algorithm.

→ Reference Book

You can define an upper limit of color differences within each square. This limit is called **Scale** parameter. After cutting an initial square grid, the quad tree based segmentation continues as follows:

- Cut each square into four smaller squares if the homogeneity criterion is **not** met.
Example: The maximal color difference within the square object is larger than the defined **Scale** value.

- Repeat until the homogeneity criterion is met at each square.



Legend: ■ Criterion not met □ Criterion met

Figure 121: Quad Tree based segmentation.

Typical for a quad tree based segmentation outcome is that in regions with high homogeneity the squares are larger than in regions of inhomogeneity.

Typically you have to try different scale parameter to find adequate image object primitives.

Compared to multiresolution segmentation, quad tree based segmentation is a reasonable trade between much better processing performance and still-useful results.

The following are examples of typical uses:

- Images with a well-separated background/foreground situation, for example, some cellular assays.
- Images with a significant proportion of background that is easy to separate. To increase the processing performance significantly for this type of image, apply a quad tree based segmentation before separating the background. Then use a different segmentation for the non-background areas of interest.

7.4.3.3 About Merging Procedures of Bottom-up Segmentations

This section introduces merging as the basic component of bottom-up segmentations.

Top-down segmentation algorithms like chessboard and quad tree based segmentation split image objects. In each internal step, image objects are split into two or four parts.

In contrast, bottom-up segmentation strategies like multiresolution segmentation and classification-based segmentation perform a **local pairwise merging** according to their specific logic. Is this way they create local decisions based on asymmetrical procedures.

Local pairwise merging distinguishes between the following image object types:

- **Seed** is the active image object that is specified through the object domain.
- **Candidate** all adjacent image objects that are potential merging partners.
- **Target** is the new image object that would potentially result from merging seed and candidate.

The goal of local pairwise merging is to find the best fitting candidate for a local merger. For all merge candidates, the potentially resulting structures (targets) are compared in detail with the input structure (seed and candidates).

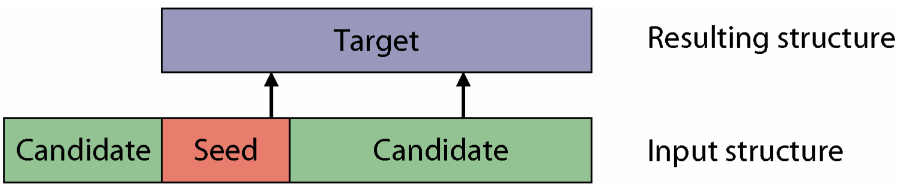


Figure 122: Image object types of local pairwise merging procedures.

7.4.3.4 Multiresolution Segmentation

The multiresolution segmentation algorithm consecutively merges pixels or existing image objects. Thus it is a bottom-up segmentation algorithm based on a pairwise region merging technique.

Multiresolution segmentation is an optimization procedure which, for a given number of image objects, minimizes the average heterogeneity and maximizes their respective homogeneity.

→ About Merging Procedures of Bottom-up Segmentations on page 162
→ Reference Book

The segmentation procedure works according the following rules, representing a mutual-best-fitting approach:

A. The segmentation procedure starts with single image objects of 1 (one) pixel size and merges them in several loops iteratively in pairs to larger units as long as an upper threshold of homogeneity is not exceeded locally.

This homogeneity criterion is defined as a combination of spectral homogeneity and shape homogeneity.

→ About the Homogeneity Criterion on page 164

You can influence this calculation by modifying the scale parameter. Higher values for the scale parameter result in larger image objects, smaller values in smaller image objects.

B. As the first step of the procedure, the seed looks for its best-fitting neighbor for a potential merger.

C. If best-fitting is not mutual, the best candidate image object becomes the new seed image object and finds its best fitting partner.

D. When best fitting is mutual, image objects are merged.

E. In each loop, every image object in the image object level will be handled once.

F. The loops continue until no further merger is possible.

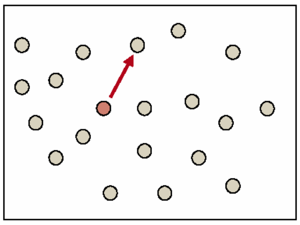


Figure 123: Each image object uses the homogeneity criterion to determine the best neighbor to merge with.

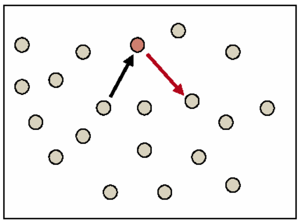


Figure 124: If the first image object's best neighbor (red) does not recognize the first image object (grey) as best neighbor, the algorithm moves on (red arrow) with the second image object finding the best neighbor.

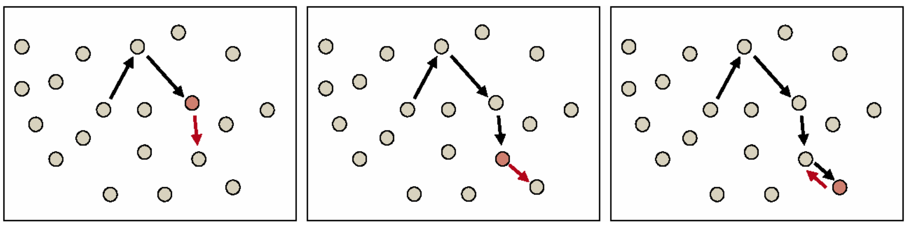


Figure 125: This branch-to-branch hopping repeats until mutual best fitting partners are found.

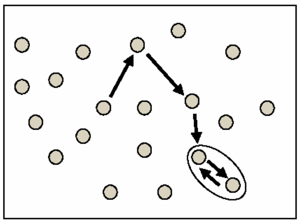


Figure 126: If the homogeneity of the new image object does not exceed the scale parameter, the two partner image objects are merged.
The procedure continues with another image object's best neighbor. The procedure iterates until no further image object mergers can be realized without violating the maximum allowed homogeneity of an image object.

With any given average size of image objects, multiresolution segmentation yields the best abstraction and shaping in any application area. However, it has higher memory requirements and significantly slower performance than some other segmentation techniques and therefore is not always the best choice.

About the Homogeneity Criterion

The homogeneity criterion of the multiresolution segmentation algorithm measures how homogeneous or heterogeneous an image object is within itself.

The homogeneity criterion is calculated as a combination of color and shape properties of both the initial and the resulting image objects of the intended merging. Here the color homogeneity is based on the standard deviation of the spectral colors. The shape homogeneity is based on the deviation of a compact or a smooth shape.

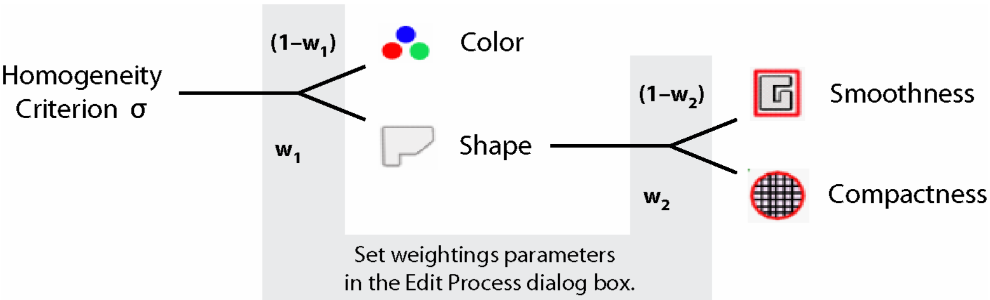


Figure 127: Weighted components of the homogeneity criterion.

The homogeneity criterion can be customized by weighting shape and compactness criteria. The weighting of color and smoothness criteria are derived from the complementary weighting:

- You can set the weighting $\mathbf{w_1}$ of shape within the homogeneity criterion between 0.1 and 0.9.
- The weighting of color assumes a value such that color and shape sums up to 1. That means, the weighting of color is $\mathbf{(1–w_1)}$.
- You can set the weighting $\mathbf{w_2}$ of compactness within the shape criterion between 0.1 and 0.9.
- The weighting of smoothness within the shape criterion assumes a value such that smoothness and compactness sums up to 1. That means, the weighting of smoothness within the shape criterion is $\mathbf{(1–w_2)}$.

- Based on the weighting settings w_1 and w_2, the weighting of compactness within the enclosing homogeneity criterion is $w_1 * w_2$. For smoothness the weighting within the homogeneity criterion is $w_1*(1-w_2)$.
- For each image object a homogeneity value is calculated expressing its homogeneity based on these weighted criteria.

> **Note**
>
> The multiresolution segmentation algorithm criteria **Smoothness** and **Compactness** are not related to the features **Smoothness** respectively **Compactness.**

For each image object its homogeneity value is multiplied by its pixel area. The resulting value, called the area weighted homogeneity criterion, is used to find best partners for merging. It is also used within the calculation of the multiresolution segmentation algorithm for comparison to the scale parameter.

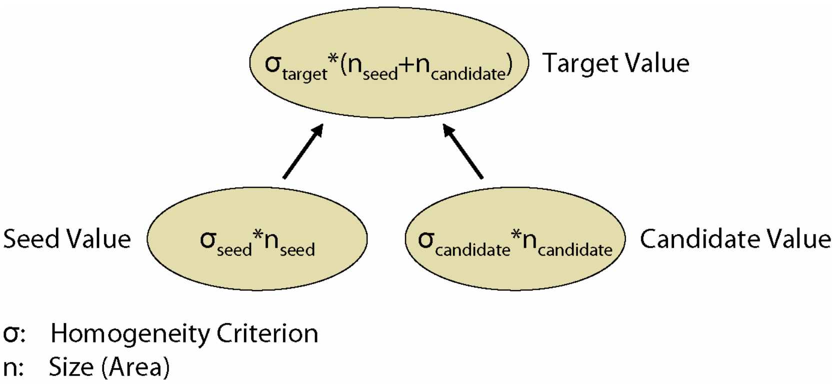


Figure 128: The area weighted homogeneity values that are compared.

7.4.3.5 Segmentation by Reshaping Algorithms

All algorithms listed under the **Reshaping algorithms** group technically belong to the segmentation strategies. They have in common that they require the existence of image objects..

Reshaping algorithms can not be used for initial finding of image object primitives; they require image objects and thus can not be not used on a plain image. However, they are fine for getting closer to regions and image objects of interest.

> **Note**
>
> Sometimes reshaping algorithms are referred to as classification-based segmentation algorithms, because they commonly use information about the class of the image objects to be merged or cut. However, this is not consistently true, and so **Definiens Developer** uses the term in the user interface to minimize irritations.

The two most basic algorithms in this group are: → Reference Book

- **Merge region** algorithm
- **Grow region** algorithm

The more complex **Image object fusion** algorithm is a generalization of the above-mentioned ones with additional options.

Merge Region

The **merge region** algorithm merges all neighboring image objects of a class to one large object. The class to be merged is specified in the image object domain.

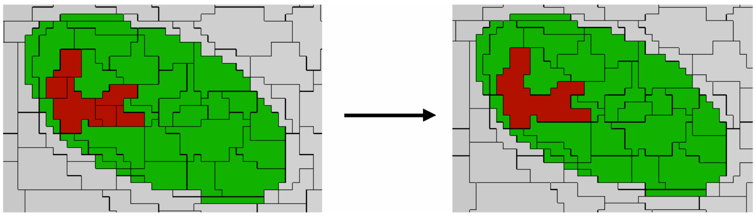


Figure 129: Red image objects are merged.

Classifications are not changed; only the number of image objects is reduced.

> **Note**
>
> The image object domain of a process using the **Merge region** algorithm should define one class only. If the target image objects belong to more than one class, the results are not likely to be predictable.

Grow Region

The **grow region** algorithm extends all image objects that are specified in the image object domain, and thus represent the seed image objects. They are extended by neighboring image objects of defined candidate classes. For each process execution, only those candidate image objects that neighbor the seed image object before the process execution are merged into the seed objects; the following sequence illustrates four **grow region** processes:

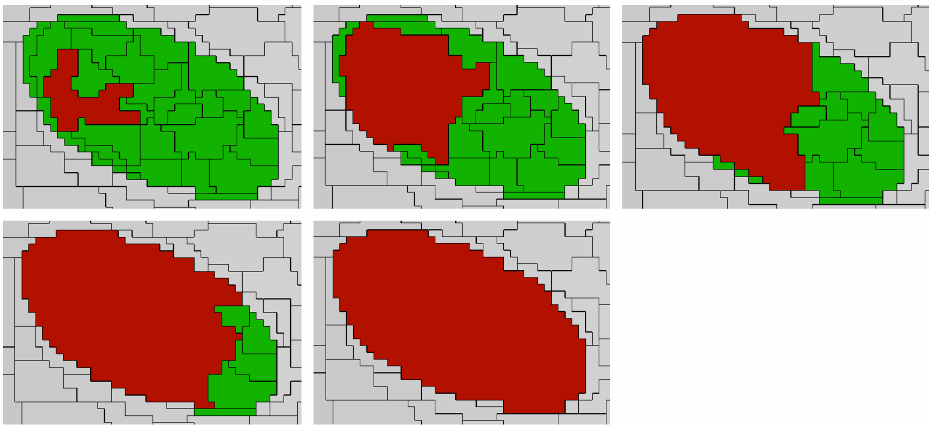


Figure 130: Red seed image objects grow stepwise into green candidate image objects.

> **Note**
>
> Grow region processes should begin the initial growth cycle with isolated seed image objects defined in the image object domain. Otherwise, if any candidate image objects border more than one seed image objects, ambiguity will result as to which seed image object each candidate image object will merge with.

7.5 Get Information on Image Objects

While developing a rule set for image analysis you need information about the attributes of image objects, such as their color or shape. Additionally, you need information on the environment of image objects, for example average brightness of all neighbors of the image object in question. There are different methods for access and display of that information.

The goal is to recognize the complex visual information which is contained within images and to translate this into processes for image analysis. Obtaining comprehensive knowledge about the features of an image and its image objects is usually good preparation for rule set development and this knowledge is also needed in most of the subsequent workflow steps. Thus, a thorough assessment and prior knowledge of the domain semantics of the image are essential preparations that will guide the detection of objects of interest.

In **Definiens** technology, an image object attribute is called a **feature**. There are simple, inherent image object features that inform about the shape or color of a single image object. They are calculated from the pixels that are a part of the image object. Other features return values based on how the image object relates to other image objects in its horizontal and vertical neighborhood. More features provide information on the classification or on the texture. Additionally, there exists functionality for assembling customized features using existing features.

→ About Features as a Source of Information on page 168

Thus, features represent a source of information concerning image objects. They help you to:

- **Investigate single image objects**: The **Image Object Information** window allows you to look at a selection features for one image object at a time.

→ Investigate Single Image Objects by Using the Image Object Information on page 174

- **Compare multiple image objects**: In the **Image Object Table** window you can list several features for many image objects, with one row for each image object, one column for each feature. You can sort the image objects by any feature. You can also navigate though the image using the table: the image object clicked in the table will be highlighted in the image.

→ Compare Multiple Image Objects by Using the Image Object Table on page 178

- **Compare feature values**: The **Feature View** display method and the **Feature View** window are functionalities that will color-code each image object based on the value returned from the feature selected.

→ Compare Feature Values by Using the Feature View on page 179

- **Analyze the correlation of two features**: The **2D-Feature Plot** window allows you to compare for all image objects two features in a separate graph, where to each axis a feature is assigned. Image objects are displayed as dots of the color representing their classification in that graph.

→ Analyze the Correlation of Two Features by Using the 2D Feature Space Plot on page 188

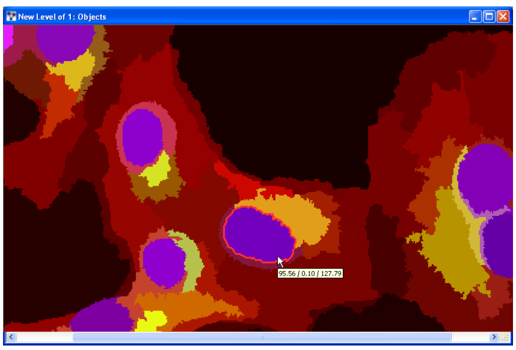


Figure 131: Project view of selected image object values. (Image data courtesy of EMBL Heidelberg).

7.5.1 About Features as a Source of Information

Image objects have spectral, shape, and hierarchical characteristics. These characteristic attributes are called **Features** in **Definiens** software. Features are used as source of information to define the inclusion-or-exclusion parameters used to classify image objects.

There are two major types of features:

- **Object features** are attributes of image objects, for example the area of an image object.
- **Global features** are not connected to an individual image object, for example the number of image objects of a certain class.

7.5.1.1 Features Overview

The available features are listed and briefly described here as an overview. For details about available features see the **Reference Handbook**.

→ Reference Handbook

Object Features

Object features are obtained by evaluating image objects themselves as well as their embedding in the image object hierarchy.

Object Features are grouped as follows:

- **Customized:** All features created in the **Edit Customized Feature** dialog box referring to object features.
- **Layer Values:** Layer values evaluate the first and second statistical moment (mean and standard deviation) of an image object's pixel value and the object's relations to other image object's pixel values. Use these to describe image objects with information derived from their spectral properties.
- **Shape:** Shape features evaluate the image object's shape in a variety of respects. The basic shape features are calculated based on the object's pixels. Another type of shape features, based on sub-object analysis, is available as a result of the hierarchical structure. If image objects of a certain class stand out because of their shape, you are likely to find a form feature that describes them.
- **Texture:** The image object's texture can be evaluated using different texture features. New types of texture features based on an analysis of sub-objects. These are especially helpful for evaluating highly textured data. Likewise, a large number of features based upon the co-occurrence matrix after Haralick can be utilized.
- **Variables:** Define variables to describe interim values related to variables.
- **Hierarchy:** This feature provides information about the embedding of the image object in the image object hierarchy. These features are best suited for structuring a class hierarchy when you are working with an image object hierarchy consisting of more than one image object level.
- **Thematic Attributes:** If your project contains a thematic layer, the object's thematic properties (taken from the thematic layer) may be evaluated. Depending on the attributes of the thematic layer, a large range of different features become available.

Class-Related Features

Class-related features are image object dependent features. They refer to the class of other image objects situated at any location in the image object hierarchy.

This location can be defined by a vertical distance in the image object hierarchy (superobjects and subobjects) or by a horizontal distance (neighbor objects). The distance is determined by the feature distance.

→ Edit the Feature Distance on page 171

- **Relations to neighbor objects:** Use these features to describe an image object by its mutual relationships to other image objects assigned to a certain class on the same level.
- **Relations to subobjects:** Use these features to describe an image object by its relationships to other image objects, assigned to a certain class on a lower level. Since the resolution increases the lower you move in the image object hierarchy, you can evaluate sub-scale information using these features.
- **Relations to superobjects:** Use these features to describe an image object by its relations to other image objects assigned to a certain class on a higher level in the image object hierarchy. Analogous to the relations to subobjects, it is possible to evaluate super-scale information here.
- **Relations to classification:** Use these features to find out about the current or potential classification of an image object.
- **Customized features:** Features created in the customized feature dialog referring to other classes are displayed here when they have first been created.

Scene Features

Scene features relate to the mean values of the entire image in the project view. Thus they are global features.

- **Variables:** Define variables to describe interim values related to scene.
- **Class-related:** Class-related features provide information on image objects of a certain class.
- **Scene-related:** Scene-related features provide information on the scene.
- **Customized feature:** Features created in the customized feature dialog referring to global features are displayed here when they have first been created.

Process-related Features

Process-related features are image object dependent features. They involve the relationship of a child process image object to the parent process. They are used in looping processes.

- **Create new 'Same super object as PPO':** Creates a new child process as the same super object of the Parent Process Object (PPO).
- **Create new 'Elliptic dist. from PPO':** Creates a new child process related to the elliptic distance from the Parent Process Object (PPO).
- **Create new 'Rel. border to PPO':** Creates a new child process relative to the border of the Parent Process Object (PPO).
- **Customized:** Features created in the customized feature dialog referring to a relation to a Parent Process Object (PPO).

Customized Features

Customized features can be arithmetic or relational features that depend on other existing features.

→ Use Customized Features on page 269

- **Create new 'Largest possible pixel value':** Creates a new feature related to the largest possible pixel value of the image layer selected.
- **Create new 'Smallest possible pixel value':** Creates a new feature related to the smallest possible pixel value of the image layer selected.

7.5.1.2 Create New Features

Some features must first be created before they may be used. They require the setting of parameter values before the feature may be created.

Prerequisite: Before a feature of image object can be displayed in the project view an image must be loaded and a (any) segmentation must be applied to the project.

→ Create Image Objects by Segmentation on page 157

1. To create a new feature, go to the **Image Object Information** window and right-click to open the context menu. Choose **Select Features to Display** to open the **Select Displayed Features** dialog box.
2. Select an item named **Create new '***feature***'**, e.g. **Create new 'Min. pixel value'.** The **[feature]** dialog box with the same name as the feature (e.g. **Min. pixel value**) will open.

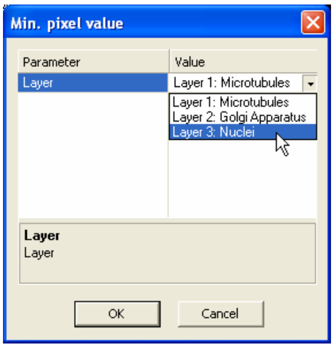


Figure 132: [Feature] dialog box for creating a new feature from the Select Displayed Features dialog box.
In this example for the feature **Layer Values** > **Pixel-based** > **Min. pixel value**, the parameter **Layer** will be set on the value **Nuclei**.

3. Depending on the feature and your project you have to set parameter values. Press **OK** and the new feature is listed in the feature tree. Additionally the newly created feature will be loaded into the **Image Object Information** window.

Find Out More

Create New Features

*New features may also be created from the **Feature View** and **Select Single Feature** dialog boxes.*

→ ***Feature View*** *on page 111*

→ ***Select Single Feature*** *on page 459*

7.5.1.3 Edit the Feature Distance

Some features may be edited to specify a distance relating two image objects. Selection of classes, scene features, or process related features may be required before the feature may be utilized.

→ About Classes on page 241
→ Scene Features on page 169
→ Process-related Features on page 169

There are two types of feature distance:

- The **level distance** between image objects on different image object levels in the image object hierarchy.
- The **spatial distance** between objects on the same image object level in the image object hierarchy.

Both kinds of feature distances maybe edited in the same way:

1. Go to the **Image Object Information** window or the **Feature View** window.
2. To change the feature distance of a feature, right-click it and choose **Edit** on the context menu. The **[***name of the feature***]** dialog box with the same name as the feature will open**.**

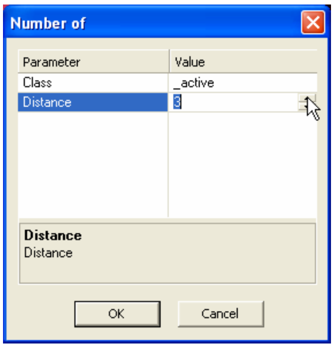


Figure 133: Editing feature distance (here the feature Number of).

3. Select the **Distance** parameter and click in the **Value** column to edit the **Distance** box. Use the arrows or directly enter the value.
4. Confirm with **OK**. The distance will be attached as a number in brackets to the feature in the feature tree.

Level Distance

The level distance represents the hierarchical distance between image objects on different levels in the image object hierarchy. Starting from the current image object level, the level distance indicates the hierarchical distance of image object levels containing the respective image objects (subobjects or superobjects).

Spatial Distance

The spatial distance represents the horizontal distance between image objects on the same level in the image object hierarchy.

Feature distance is used to analyze neighborhood relations between image objects on the same image object level in the image object hierarchy. It represents the spatial distance in the selected feature unit between the center of masses of image objects. The (default) value of 0 represents an exception as it is not related to the distance between the center of masses of image objects; only the neighbors that have a mutual border are counted.

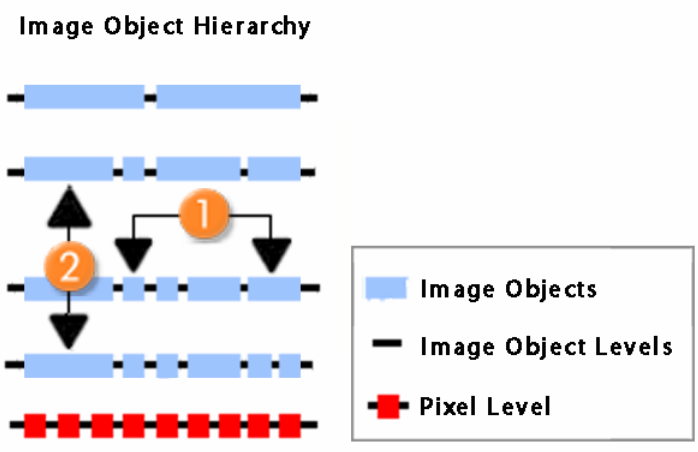


Figure 134: The image object hierarchy describes multiple image object levels.

The **Spatial Distance** 1 describes the horizontal distance between image objects.
The **Level Distance** 2 describes the vertical distance between object levels.

7.5.1.4 Thematic Attributes

Thematic attributes can only be used if a thematic layer has been imported into the project. If this is the case, all thematic attributes in numeric form that are contained in the attribute table of the thematic layer can be used as features in the same manner as you would use any other feature.

Thematic Layer Attribute Table

The **Thematic Layer Attribute Table** can be accessed from the main menu by choosing one of the following:

- **Tools** >**Thematic Layer Attribute Table**
- **View** > **Windows** >**Thematic Layer Attribute Table**

To view the thematic attributes, go to the **Manual Editing** toolbar and select **Thematic editing** under **Change Editing Mode**. From the **Select thematic layer** drop-down list, choose the name of the thematic layer whose attributes you want to display. The attributes are shown in the **Thematic Layer Attribute Table**.

Manual Editing

→ View Thematic Layer Attributes on page 286
→ Working with Thematic Layers on page 283

7.5.1.5 Legacy Features

A legacy feature is a feature introduced in earlier versions of **Definiens** software. Legacy features are listed in the feature tree, to enable usage of projects or rule sets that have been created by earlier versions of **Definiens**. The latest possible version number for usage of a legacy feature is noted after the feature name in the feature tree, e.g. **Elliptic fit (legacy feature, up to V3.5)**. If available, legacy features enable you to work consistently with rule sets created using earlier versions of **Definiens** software. However, we recommend adapting these rule sets on the current feature version to enable the most accurate results.

7.5.1.6 Object Oriented Texture Analysis

Object-oriented texture analysis allows you to describe image objects by their texture.

Precondition: An image level of subobjects has to exist.

By looking at the structure of a given image object's subobjects, an object's form and texture can be determined. An important aspect of this method is that the respective segmentation parameters of the subobject level can easily be adapted to come up with subobjects that represent the key structures of a texture.

A straightforward method is to use the predefined texture features provided by **Definiens Developer**. They enable you to characterize image objects by texture, determined by the spectral properties, contrasts and shape properties of their subobjects.

Another approach to object oriented texture analysis is to analyze the composition of classified subobjects. Class-related features (relations to subobjects) can be utilized to provide texture information about an image object, e.g., the relative area covered by subobjects of a certain classification.

Further texture features are provided by texture after Haralick. These features are calculated based upon the so-called co-occurrence matrix, which is created out of the pixels of an object.

> **Note**
>
> The calculation of Haralick texture features can require considerable processor power, since for every pixel of an object a 256 x 256 matrix has to be calculated.

7.5.2 About Metadata as a Source of Information

Many image data formats include metadata providing information about the related image, for example the acquisition time. Considering metadata might be beneficial for image analysis if you relate it to features.

The available metadata depends on the image reader or camera used, the industry-specific environment, and settings. Industry-specific examples are:

- Satellite image data may contain metadata providing cloudiness information.
- Microscopy image data may contain metadata providing information about the used magnification.

Definiens Developer can provide a selection of the available metadata. This selection is defined in a metadata definition which is part of the rule set.

The provided metadata can be displayed in the **Image Object Information** window. Further, it is listed together with features and variables in the feature tree of for example the **Feature View** window or the **Select Displayed Features** dialog box.

→ Investigate Single Image Objects by Using the Image Object Information on page 174
→ Select Features on page 457

Convert Metadata to Provide it to the Feature Tree

When importing data, you can provide a selection of available metadata. To do so, you have to convert external metadata to an internal metadata definition. This provides a selection of the available metadata to the feature tree and allows its usage in rule set development.

When developing rule sets, metadata definitions will be included in rule sets allowing the serialization of metadata usage.

Metadata conversion is available within the following import functions:

- Within the **Create Project** dialog box.

→ Create a New Project on page 44

- Within the **Customized Import** dialog box on the **Metadata** tab.

→ Metadata Import Options on page 72

7.5.3 Investigate Single Image Objects by Using the Image Object Information

When analyzing individual images or developing rule sets you need to investigate single image objects often. Use the **Features** tab of **Image Object Information** window for acquiring information on a selected image object. It makes the classification process transparent and amendable while allowing you to obtain specific information about an image object.

→ Select and Deselect Image Objects on page 115

Image Objects consist of spectral, shape, and hierarchical elements. These elements or characteristic attributes are called **Feature**s in **Definiens Developer.** In the **Feature** tab within the **Image Object Information** window, the values of the characteristic attributes of the selected features are displayed when an image object is selected from within the project view.

The **Image Object Information 2** window offers the same functionalities as the **Image Object Information** window. It can be used for comparison of image objects.

Prerequisite: Before a feature of image object can be displayed in the project view an image must be loaded and a (any) segmentation must be applied to the project.

→ Create Image Objects by Segmentation on page 157

1. To open the **Image Object Information** window do one of the following:

   Image Object Information

   - Choose **View** > **Image Object Information** on the main menu bar.
   - Choose **Image Objects** > **Image Object Information** on the main menu bar.
2. To acquire information on a specific image object click an image object in the project view. Some features are listed by default.
3. To add or remove features, right-click the **Image Object Information** window and choose **Select features to Display.** The **Select Displayed Features** dialog box opens Select the feature of your interest.

   → Select Features on page 457
4. The selected feature 4 values are now displayed in the project view. To compare single image objects, click another image object in the project view and the displayed feature values are updated.
5. To deselect a selected image object, click it in the project view a second time.

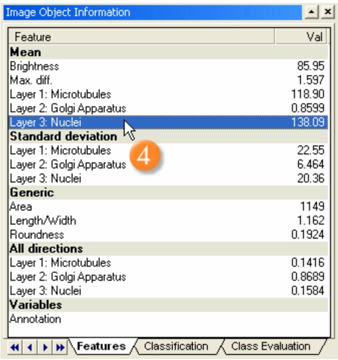


Figure 135: Image Object Information window.
It displays selected features of the image object that is selected in the project view.

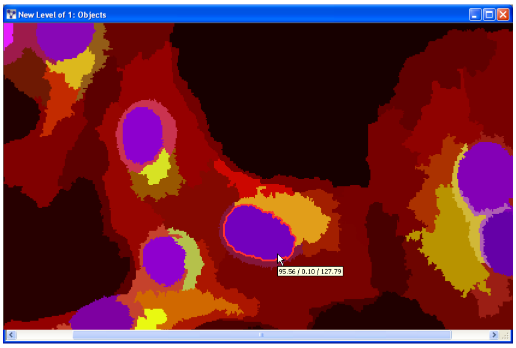


Figure 136: Project view of selected image object values. (Image data courtesy of EMBL Heidelberg).
Selected image object with the calculated means of the image layers displayed under the cursor.

6. If the processing for image object information takes too long, or if you want to cancel the processing for any reason, you can use the **Cancel** button in the status bar to cancel processing.

→ Cancel Image Analysis of an Open Project on page 368

Options

4 Double-click a feature 4 to display it in the project view in the same way as if it had been selected in the **Feature View** window.

5 The advanced functionalities of the **Classification** tab and **Class Evaluation** tab 5 are mentioned in the **About Classification** section.

→ Compare Feature Values by Using the Feature View on page 179

Context Menu Options

4 To display available context menu commands, right-click a feature 4 in the **Image Object Information** window.

> **Note**
>
> Only the available context menu options applicable to the selected feature will be displayed. They differ according to the selected feature and by different products.

- **Display in Image Object Information:** Deselect to remove the feature.
- **Select Features to Display:** Opens the **Select Displayed Features** dialog box.

→ Select Features on page 457

Find Out More

Display in Image Object Information

*Another way to add or remove features is to right-click on a feature in windows, where you have the option to select features. For instance, in the **Feature View** window in the context menu select **Display** in **Image Object Information** and the feature appears in the **Image Object Information** window. Likewise, when editing or creating a Class using the **Insert Expression** dialog box the newly inserted Membership Function will be added to the features selected within the **Image Object Information** window.*

> **Note**
>
> If you select **Edit** or **Create** from **Object Features > Customized**, you will activate the advanced **Edit Customized Feature** dialog box.

- **Edit:** Opens the **[***feature***]** dialog box with the same named as the feature. For example, if a feature from within the **Standard Deviation** feature tree grouping is selected, the **Standard Deviation** editing dialog box will open. It is used for editing parameter values of a feature.

→ Features Overview on page 168

- **Edit Object Value:** Opens **Edit Annotation** dialog box where you may change the selected image object value. The new value is displayed after its value in the feature tree as well as in the project view before the actual value.
- **Find:** Opens **Find and Replace** window. You may then substitute a created feature with another.

→ Find and Replace Rule Set Items on page 147

- **Delete:** Deletes the feature completely.
- **Create:** Opens a create dialog box, which is named after the *[Feature Group]*.
- **Create for All [***Feature Group***]:** Creates new features within the Feature View feature tree for all of the [*Feature Group*].
- **Manage Variables:** Opens the **Manage Variables** dialog box.
- **Manage Customized Features:** Opens the **Manage Customized Features** dialog box.
- **Save:** Saves feature as ***.duf** file.
- **Load:** Import saved customized features from ***.duf** files.
- **Mark as Shared:** Designates feature as **shared** in order to facilitate merging it within multiple rule sets.

> **Note**
>
> Once a feature is shared, it may not be edited.

- **Edit Unit:** Opens the **Select Unit** dialog box. You change the feature unit if available. The unit of a feature is displayed after its value in the feature tree.
- **Save to File:** Saves the dialog box contents as tabular ***.csv** or ***.tcsv** file.
- **Copy to Clipboard**: Saves the active dialog box to the clipboard.

7.5.3.1 Example: GetInfoOnImageObjects.dpr

For better understanding, we recommend opening the test project which is included within the installation. The test project is provided so that you might experiment with different methods and settings useful in analyzing information concerning image objects and features. We recommend saving your modified test project under a new name.

1. Open the test project by one of following ways:

    - Open the project named **GetInfoOnImageObjects.dpr**. The path is: **C:\Program Files\Definiens Developer** *Version number***\Examples \TestProjects\GetInfoOnImageObjects.dpr**.

    - Alternatively, open the workspace named **TestWorkspace.dpj**. The path is: **C:\Program Files\Definiens Developer** *Version number***\Examples \TestWorkspace\TestWorkspace.dpj**.
    Next, open the **Cell Assay** folder within the **Workspace** tree view of the **Workspace** window.
    Then view the project by double-clicking the **Cells** project within the **Name** column of the **Workspace** window.

→ Manage Projects in a Workspace on page 56

2. Make sure the **Image Object Information** window is open.

Image Object Information

3. You will notice that the image is present and a process tree is loaded and executed. After clicking an image object in the project view, some features will appear in the **Image Object Information** window.

4. Go to the **Image Object Information** window and add the following features by right-clicking and choosing **Select Features to Display** on the context menu:

    - **Object features > Layer Values > Mean > Brightness**
    - **Object features > Layer Values > Mean > Max. diff.**
    - **Object features > Layer Values > Standard deviation > (**all 3 image objects**)**
    - **Object features > Shape > Generic > Area**
    - **Object features > Shape > Generic > Length/Width**
    - **Object features > Shape > Generic > Roundness**
    - **Object features > Texture > Texture after Haralick > GLCM Homogeneity > All directions > (**all 3 image objects**)**

5. Go to the project view and click on single image objects to compare their feature values listed in the **Image Object Information** window.

6. In the **Image Object Information** window, double-click on **Layer3:Nuclei**. Now all image objects that are found within this layer are displayed in shades of gray. Each image object is displayed in a grayscale value according to the feature value that is selected for visualization. The low feature values are darker while the high values are brighter. Image objects displayed in red have not been defined for the evaluation of the chosen feature.

→ Get a Visual Overview of Feature Values on page 180

7. When the cursor is held above an image object for more than one second, the values of the image object will be displayed in the cursor tool tip.

8. In order to switch between the **Feature View** display method and the regular layer display in the project view, click the **View Layer** button or the **Feature View** button on the **View Settings** toolbar.

View Layer

Feature View

7.5.4 Compare Multiple Image Objects by Using the Image Object Table

Compare multiple classified image objects that are listed in table together with various feature values. Comparing multiple image objects will be done in different situations when you have already classified image objects that you want to investigate in detail. You may want to compare image objects of selected classes in cases of:

- Evaluation of classifications.
- Selection of image objects according to sorted feature values.
- Debugging and quality assurance.

The **Image Object Table** window allows you to sort the image objects according to feature values.

1. To open the **Image Object Table** window do one of the following:
   - Choose **Image Objects > Image Object Table** on the main menu bar
   - Choose **View > Windows > Image Object Table** on the main menu bar.
2. To configure the **Custom** tab of the **Image Object Table** window, right-click in the text area, choose **Configure Image Object Table** in the context menu. The **Configure Image Object Table** dialog box opens.

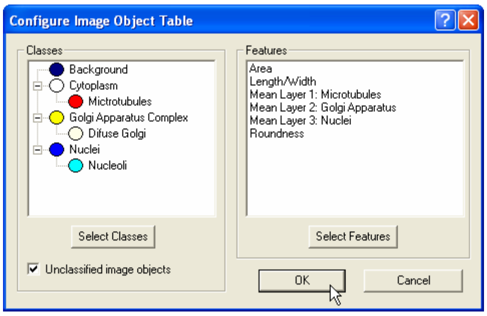


Figure 137: Configure Image Object Table dialog box.

3. **Select Classes** to display all its image objects. The **Select Classes for List** dialog box opens. Select the classes individually by clicking them or using the move **All** buttons.

→ Select Classes on page 456

4. Additionally, you may include **Unclassified Image Object** by selecting the check box.
5. **Select Features** to display the feature values and the **Select Features for Export** dialog box opens. Select the features individually by double-clicking them.

→ Select Features on page 457

6. Click **OK** to display all image objects and their feature values as configured in the **Image Object Table**. Each image object is assigned an **ID** number.

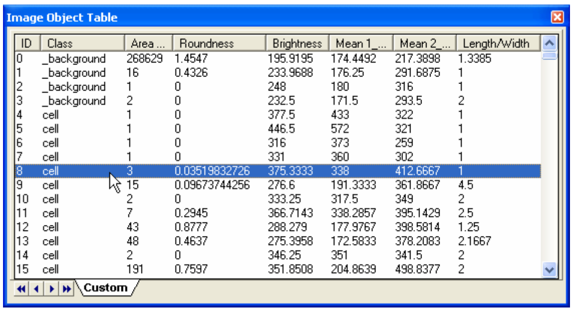


Figure 138: Image Object Table window.

7. Click on a column header to resort rows according to column values.

8. Depending on the export definition of the used analysis, there may be other tabs listing dedicated data.

Context Menu Options

To display available context menu commands, right-click on the text area of the **Image Object Table** window.

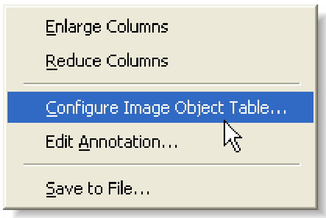


Figure 139: Image Object Table context menu.

- **Enlarge Columns**: Expand every column to fit the width of the corresponding text.
- **Reduce Columns:** Minimize columns to default width.
- **Configure Image Object Table:** Start point to work with the table (see step 2 above).
- **Edit Annotation:** Right-click on a table entry to add or modify annotations, which will be displayed in the second column (Class).
- **Save to File:** Export the **Image Object Table** or right-click a single entry as a **.csv** file or **.tcsv** file in the **Save Contents** dialog box.

7.5.5 Compare Feature Values by Using the Feature View

Compare image object attributes to evaluate which features are suitable for separating the relevant classes. A frequent task in image analysis is to reduce the number of features utilized for classification to the essential ones.

Feature View is a display method to visualize features over the entire project view. It helps the user to visualize the properties of image objects in a graphical way and therefore provides intuitive access to the peculiarity of a certain feature over all image objects in a project. Together with the **Feature View** window it allows you to perform the following tasks:

- Get visual overview of a feature by displaying an image object feature.

→ Get a Visual Overview of Feature Values on page 180

- Investigate threshold values of a feature by editing the range of the display of a feature.

→ Investigate Threshold Values of a Feature on page 181

The **Feature View** display method renders each image object in the project view according to its value for a selected feature. Each image object is displayed in a grayscale value according to the feature value that is selected for visualization. The low feature values are darker while the high values are brighter. Objects displayed in red have not been defined for the evaluation of the chosen feature. If you view, for example , the feature **Rel. border to brighter neighbors** all objects without a brighter neighbor will be displayed in red.

7.5.5.1 Get a Visual Overview of Feature Values

To get a visual overview of the properties of a specific feature, you can display an image object feature on the project view.

1. To display an image object feature, you have to activate the **Feature View** display method. Do one of the following:
   - Go to the **Feature View** window and double-click a feature you want to visualize.

     Feature View

   - Go to the **Image Object Information** window and double-click a feature you want to visualize.

     Image Object Information

   - From the context menu of either window, use **Select Features** or **Select Single Feature**, and select the feature you want to visualize by right-clicking it and choosing **Update Range** on the context menu.

   → Select Features on page 457

   → Select Single Feature on page 459

   Each image object is displayed in a grayscale value according to the feature value that is selected for visualization. The low feature values are darker while the high values are brighter. Image objects displayed in red have not been defined for the evaluation of the chosen feature.
2. Go to the project view and point (without clicking) to an image object to display its feature value in the cursor tool tip.

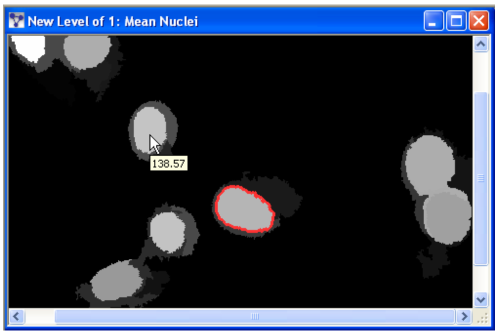


Figure 140: Using Feature View display method (image data courtesy of EMBL Heidelberg).

In this example from the test project, the **Feature View** display method visualizes the ratio that the image objects have in the blue image layer. In this case you can see, that the selected feature (**Object features > Layer values > Mean > Layer 3**) is suitable for the separation of these image objects Nuclei. Darker image objects have a lower value than lighter ones.

→ Example: CompareImageObjectAttributes.dpr on page 184

3. Display another feature by double-clicking to compare it with the previous feature.

4. Click the **View Layer** button to stop the **Feature View** display method. Click the **Feature View** buttons to come back to the **Feature View** display method.

View Layer

Feature View

Tip

Duplicate Image Object Level

For investigating the features related to superobjects or subobjects you may want to duplicate the image object level. Beside creating a dedicated process using the ***copy image object level*** *algorithm, you simply may choose* ***Image Objects*** *>* ***Copy Current Level*** *on the main menu bar.*

→ *Duplicate Image Object Level on page 195*

7.5.5.2 Investigate Threshold Values of a Feature

By editing the range of the display of a feature you can limit the **Feature View** display function to a specified range. All image objects whose values are within the range are colored according to the adjusted range in a smooth transition from blue (low values) to green (high values). Thus you can investigate suitable threshold values of a feature.

1. To open the **Feature View** window, do one of the following:
   - Choose **Tools** > **Feature View** on the main menu bar.
   - Choose **View** > **Feature View** on the main menu bar.

Feature View

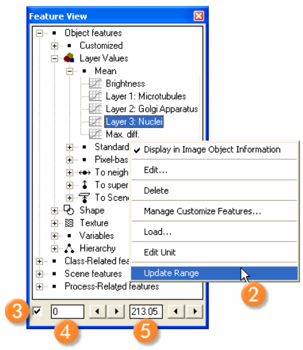


Figure 141: Feature View window with contextual menu active.

2. To display a feature from the **Feature View** window do one of the following:
   - Right-click an image object feature 2 and select **Update Range**.
   - Double-click the desired feature.

   The feature will be displayed in grayscale with darker image objects having a lower value than lighter ones. Also, the feature will be added to the feature tree in the **Image Object Information** window.

3. Select the checkbox 3 to activate the feature range display function. Nearly all image objects whose values are within the range are colored according to the adjusted range in a smooth transition from blue (low values) to green (high values).

Notes

If you do not select **Update Range** as noted in step 2, selecting the checkbox will automatically reuse the and range of the last displayed feature even if you selected another feature.

When updating, the feature range is calculated as approximated values. Thus you may recognize single image objects providing feature values that are outside the updated feature range.

If you show outlines using the **Show or Hide Outlines** button, the image objects will not be colored as described above.

When the **Feature View** display method is activated, the **Single Grayscale**, **Mix Three Layers RBG**, and **Show Previous/Next Image Layer** buttons do not function in the project view. However, their functionality is available to the **Pan Window**.

4. When editing the feature range the display colors will be adapted accordingly.
   - Edit the minimum feature value of the display in the text box 4 or use the arrows to the right.
   - Edit the maximum feature value of the display in the text box 5 or use the arrows to the right.
5. The blue and green colored areas indicate those image objects that belong to the indicated feature range 4 5. That means, classing image objects using these values will return you the image objects located in the blue and green colored areas.
6. Try out different features and ranges so that the blue and green colored areas match an image object that you want to describe. Use the arrow buttons for experimentation. The blue and green colored areas represent the image objects falling within the selected feature range. White, gray or black areas represent the image objects providing feature values outside the feature range.
7. Have a closer look to the areas within the selected range (blue and green) 1 which border uncolored areas (white, gray and black) 1. In these bordering areas, those image objects are located that provide suitable threshold values to classify them as separate or groups of image objects.

→ Assign Class Algorithm on page 190

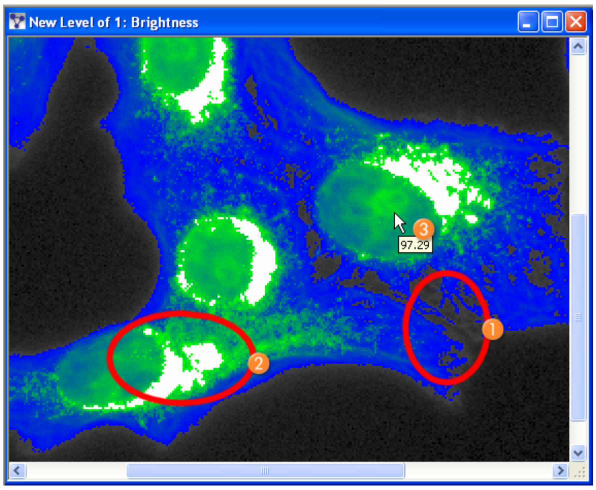


Figure 142: Visualizing image object threshold values using Feature View and Project View windows (Image data courtesy of EMBL Heidelberg).
In this example of a cellular assay, the feature range display method visualizes image objects that provide potential threshold values. Areas highlight useful threshold values: **Black** 1 regions fall below the selected range, while **White** 2 regions lie above the selected range. Variations in the

Blue-Green hue may give further information about potential image objects; here the individual nucleoli ③ are faintly visible within the nuclei.

Context Menu Options

To display available context menu commands, right-click a feature in the **Feature View** window.

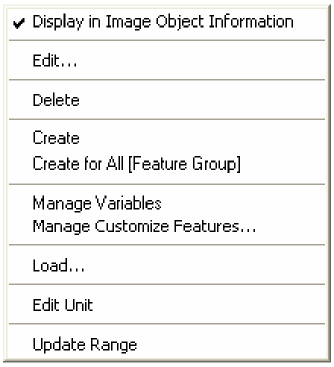


Figure 143: Contextual Menu from Feature View window.

> **Note**
>
> Only the available context menu options applicable to the selected feature will be displayed. They differ according to the selected feature and by different products.

- **Display in Image Object Information:** Deselect to remove the feature.

> **Note**
>
> If you select **Edit** or **Create** form **Object Features >Customized**, you will activate the advanced **Edit Customized Feature** dialog box.
>
> → **Create Customized Features** on page 269

- **Edit:** Opens the *[feature]* dialog box with the same named as the feature. For example, if a feature from within the **Standard Deviation** feature tree grouping is selected, the **Standard Deviation** editing dialog box will open. It is used for editing parameter values of a feature.
- **Find:** Opens **Find and Replace** window. You may then substitute a created feature with another.

→ **Find and Replace Rule Set Items** on page 147

- **Delete:** Deletes the feature completely.
- **Create:** Opens a create dialog box, which is named after the *[Feature Group]*.
- **Create for All** [*Feature Group*]**:** Creates new features within the Feature View feature tree for all of the [*Feature Group*].
- **Manage Variables:** Opens the **Manage Variables** dialog box.
- **Manage Customized Features:** Opens the **Manage Customized Features** dialog box.

- **Load:** Import saved customized features from **.duf** files.
- **Edit Unit:** Opens the **Select Unit** dialog box. You change the feature unit if available. The unit of a feature is displayed after its value in the feature tree.
  Not available for variables.
- **Update Range:** When checkbox is selected, the complete range of the feature variables are displayed within the range boxes 4 5.
  Not available for variables.

7.5.5.3 Example: CompareImageObjectAttributes.dpr

For better understanding, we recommend opening the test project which is included within the installation. The test project is provided so that you might experiment with different methods and settings useful in analyzing information concerning image objects and features. We recommend saving your modified test project under a new name.

1. Open the test project by one of following ways:
   - Open the project named **CompareImageObjectAttributes.dpr**. The path is: **C:\Program Files\Definiens Developer** *Version number***\Examples \TestProjects\CompareImageObjectAttributes.dpr**.
   - Alternatively, open the workspace named **TestWorkspace.dpj**. The path is: **C:\Program Files\Definiens Developer** *Version number***\Examples \TestWorkspace\TestWorkspace.dpj**.
     Next, open the **2nd Project** folder within the **Workspace** tree view on the left panel of the **Workspace** window. Then view the project by double-clicking the **Cells** project within the **Name** column of the **Workspace** window.

→ Manage Projects in a Workspace on page 56

2. Make sure the **Image Object Information** window and the **Feature View** windows are open.

Image Object Information

Feature View

3. For any selected feature, an image depicting the values of all objects may be displayed using the **Feature View** window. Double-click here on **Object features > Layer Values > Mean > Layer 1: Microtubules** to look at the values for the first layer (microtubules staining), on **Layer 2: Golgi Apparatus**, to observe the values for the second layer (Golgi staining), or **Layer 3: Nuclei** to see its corresponding values (Nuclei staining)**.**

Note

Unless you choose **Update Range** from the context menu, the program will reuse the same approximated range from the previously chosen feature.

Note

The approximated range does not necessarily include the whole range of values for the selected feature.

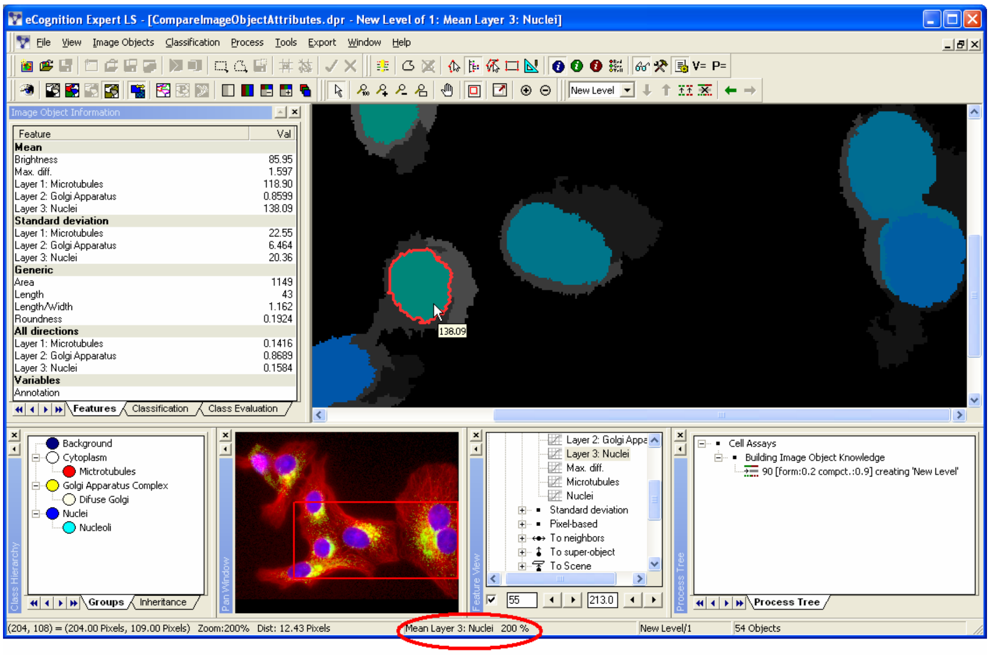


Figure 144: Main window with marked indication of the image layer (image data courtesy of EMBL Heidelberg).

4. **Feature View**: Activate the small check box in the bottom area of the **Feature View** dialog in order to be able to enter numbers as limits in the neighboring text fields. **Definiens Developer** will then display in green and blue those objects which fall within the specified values.

5. Try to identify the values which separate the nuclei from the rest of the image objects in the test project:

   - In the **Feature View** window, click on **Layer3: Nuclei**, the path is: **Object features > Layer Values > Mean > Layer3: Nuclei**.
   - Check the check box in the **Feature View** window, and right-click the feature to **Update Range** from the context menu.
   - Click on the right arrow of the lower value (left bottom) in the **Feature View** window until the **nuclei** stand out as blue and green object in a sea of black. This will occur at around 55. This is the threshold (for this feature) at which nuclei maybe separated from other object images in this image layer.

Note

Image objects that fall slightly bellow the selected range maybe dark gray in color. Likewise, image objects that are light gray in color lie slightly above the selected range. These image objects are near the threshold for the selected feature range.

6. Look for another image object you wish to exclude and identify its value using the cursor tool tip as described in step 2 of the **Get a Visual Overview of Feature Values** section.

→ Get a Visual Overview of Feature Values on page 180

7. Enter this restricting value into the minimum or maximum feature value box as described in step 4 the **Investigate Threshold Values of a Feature** section. As the range is < or > function, it will now fall outside the selected range. You may now use this information when classifying image objects.

8. In order to switch between the **Feature View** display method and the regular layer display in the project view, click the **View Layer** button or the **Feature View** button on the **View Settings** toolbar.

View Layer

Feature View

7.5.6 Compare Two Features by Using the 2D Scatter Plot

The **2D Scatter Plot** is a diagram which compares two features of a project. It provides information about image objects. You can select the features and classes to be compared.

Available only for Definiens eCognition **Life** Server

The information of the **2D Scatter Plot** always refer to the current project.

Prerequisites: A project must be open.

To open the **2D Scatter Plot** window, do one of the following:

- Choose **View** > **Windows** > **2D Scatter Plot** from the main menu.
- Click the preset layout **Result View**.

2D Scatter Plot

Result View

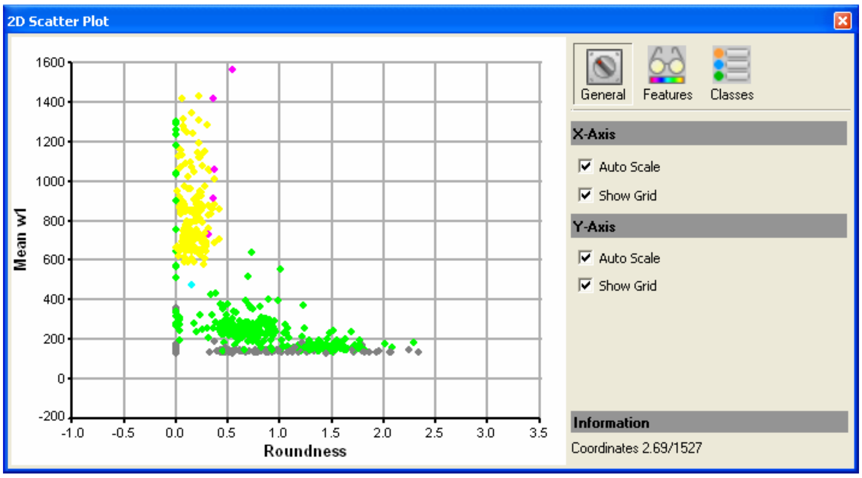


Figure 145: 2D Scatter Plot window.

The example displays the roundness of image objects (x-axis) compared to the mean value of this feature (y-axis).

→ **View Legend of Classes** on page 380

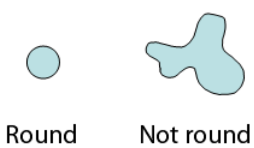


Figure 146: Samples for roundness.

The color coding refers to the colors of the classes.

Each dot in the **2D Scatter Plot** represents an image object of the project in the **Project View**.

→ **Manage Data in the Plate View** on page 90

Selecting an dot in the **2D Scatter Plot** selects the corresponding image object in the **Project View** and vice versa.

7.5.6.1 Configure the 2D Scatter Plot

Configure the display of the **2D Scatter Plot** window.

Available only for Definiens eCognition **Life** Server

1. Change to the **General** tab of the **2D Scatter Plot** window.

General **General tab**

2. To turn off the **Auto Scaling,** clear the **Auto Scale** check box for either the x- or the y-axis. Auto scaling adjusts either the axis to the value range of the selected feature. If you turn it off, the current scale stays. Further features that you select are displayed within this scale. This can lead to distorted results, when the range of value does not fit in the current scale.
3. By default, the **2D Scatter Plot** window displays a grid for both axes. To turn it off, clear the **Show Grid** check box for either or both axes.

7.5.6.2 Select Features for the Scatter Plot

You can select from a variety of features to be displayed in the **2D Scatter Plot** window.

Available only for Definiens eCognition **Life** Server

1. Open the **Features** tab of the **2D Scatter Plot** window.

Features **Features** tab

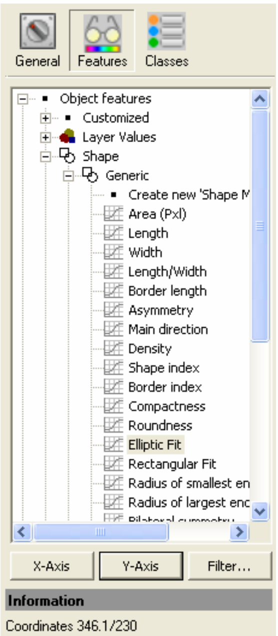


Figure 147: Features tab of the 2D Scatter Plot window.

2. In the tree view, select the feature you want to display on the x-axis.
3. Click **X-Axis**. The value of the image objects for the selected feature is displayed in the 2D Scatter Plot.
4. Select the feature for the **Y-Axis** and click the respective button.
5. You can change the selection any time by selecting a new feature and placing it on either axes.

6. To switch the axes, select the feature which is currently displayed on the x-axis and click the **Y-Axis** button. Do likewise for the feature on the other axis.

7.5.6.3 Filter 2D Scatter Plot by Classes

By default, the **2D Scatter Plot** displays all classes. You can select or deselect classes to reduce the number value dots of the **2D Scatter Plot**.

Available only for Definiens eCognition **Life** Server

1. Open the **Classes** tab of the **2D Scatter Plot** window.

Classes tab

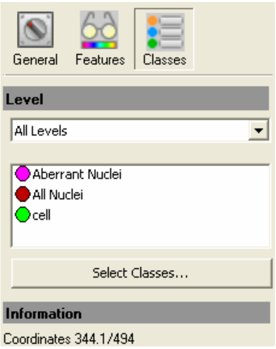


Figure 148: Classes Tab of the 2D Scatter Plot window.

2. From the **Level** list, select a specific level or leave the entry to **All Levels**.
3. Click **Select Classes** to open the **Edit Classification Filter** dialog box.
4. Select a class. To select more than one class, press the Ctrl key while clicking.
5. Click **OK** to apply the filter. The **2D Scatter Plot** displays only the selected classes.
6. To view all classes available in the project, select **Always use all classes.**
7. To show all classes again, click **Deselect All** button in the **Edit Classification Filter** dialog box.

7.5.7 Analyze the Correlation of Two Features by Using the 2D Feature Space Plot

Analyze the correlation of two features. If two features correlate highly, one of them may be deselected from the **Image Object Information** or **Feature View** windows. Acquire information about where an object or a group of objects is situated in a two-dimensional feature space. The **2D Feature Space Plot** dialog box facilitates the examination of the distribution of feature values plotted on x-y-axes of two different assigned features.

Similarly, as with **Feature View** window, not only spectral information maybe displayed, but all available features.

1. To open choose **Tools > 2D Feature Space Plot** on the main menu bar.

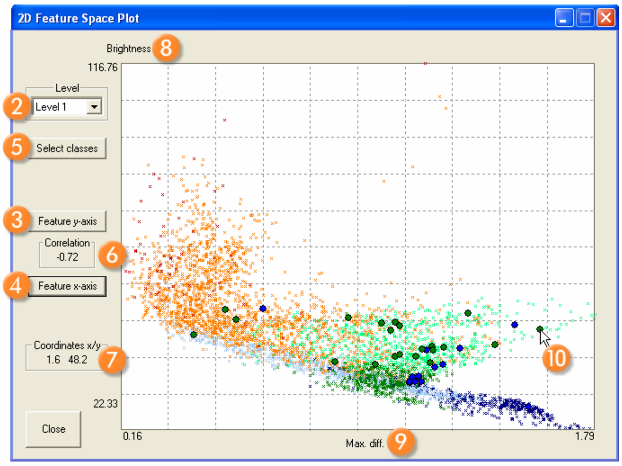


Figure 149: 2D (two-dimensional) Feature Space Plot dialog box.

2. Select the image object **Level** 2 for which feature values are to be plotted.
3. Click **Feature y-axis** 3 to select the feature to be plotted along the y-axis. In the **Select Feature for Y-axis** dialog box you may double-click on one feature.

→ Select Single Feature on page 459

4. Click **Feature x-axis** 4 to select the feature to be plotted along the x-axis. In the **Select Feature for X-axis** dialog box you may double-click on one feature.
5. **Select Classes** 5 for the feature space plot.

→ Select Classes on page 456

6. Group box display 6 of the Pearson's correlation coefficient between the values of the selected features and the selected image objects or classes.
7. x- & y-axes position 7 of the cursor in the feature space plot.
8. Name 8 of the feature plotted along the y-axis. Range is displayed vertically.
9. Name 9 of the feature plotted along the x-axis. Range is displayed horizontally.
10. Unclassified image objects are displayed as small black crosses 10. Image objects that are assigned a class () as small crosses colored according to the class color. **Sample** objects selected in the project view are displayed as circles colored in the class color. (For more information about selecting objects, see the section on **Manual Editing**).

→ Manual Editing on page 391

Note

In cases where you want to utilize **Sample** objects they must first be activated and edited.

→ Navigate Samples on page 231

7.6 Classify Image Objects

After image objects have been created in your scenes, you classify them to give them both a meaning and a label. Information contained in image objects is used as a filter for classification.

Based on a classification of image objects, you can analyze and interpret complete images. To perform a classification, appropriate classes need to be defined. During classification, the image objects are analyzed according defined criteria and assigned to classes that best meet the defined criteria.

7.6.1 Introduction to Classification

Definiens Developer offers two principal methods to classify image objects, all of which have their own strengths within their field of application. You can also combine them to improve your results.

- **Processes-based classification** allows you to combine several steps of image object classification into one process and execute it at once. For the classification steps you can choose from different types of classification algorithms and combine them to meet your needs.

→ About Process-Based Classification on page 190

- For experienced users, writing **class descriptions** and arranging classes hierarchically is a reliable method to fine-tune the classification. When you want to use fuzzy logic to classify your image object and when multiple conditions are necessary to assign an image object to a class, you need to work with class descriptions.

→ About Classification with Class Descriptions on page 212

→ About Classification on page 208

7.6.2 About Process-Based Classification

Process-based classification enables you to to classify image objects using algorithms and thresholds.

A classification process consists of the following components:

- **Algorithm** define how image objects are classified.

→ Algorithms for Process-Based Classification on page 208

- **Image object domain** define what image objects are considered for classification.

→ Define the Image Object Domain on page 135

- As a part of the image object domain, the **Condition** defines what feature an image object must hold to be classified.

7.6.2.1 Assign Class Algorithm

The **Assign Class** algorithm is the most simple classification algorithm. It determines by means of a threshold condition whether the image object is a member of the class or not.

This algorithm is used, when one threshold condition is sufficient to assign an image object to a class.

1. In the **Edit Process** dialog box, select **assign class** from the **Algorithm** list.

2. In the **Image Object Domain** group click the **condition** button; it is labeled **no condition** when no condition is yet selected.
3. Select a feature for the condition and define the operator and the reference value.
4. For the algorithm parameter **Use class**, select a class or create a class by selecting the entry **<create new class>**.

→ Use Thresholds for Classification with Class Descriptions on page 212

8 Develop Efficient Rule Sets

This chapter instructs you in using available tools to build efficient and reusable rule sets and other ruleware. It enables you to benefit from the tools of the development environment.

→ **Use Hierarchical Image Object Levels** on page 193
→ **Focus Processes by Using the Domain Concept** on page 197
→ **About Classification** on page 208
→ **Use Variables in Rule Sets** on page 257
→ **Use Customized Features** on page 269
→ **Reuse Process Sequences with Customized Algorithms** on page 276
→ **Working with Thematic Layers** on page 283
→ **Working with Polygons and Skeletons** on page 290
→ **Automate the Workspace Processing** on page 295
→ **Create Action Libraries** on page 306

We recommend that you simultaneously consult the **About Strategies** chapter, focusing on composing rule sets based on the **Definiens** Cognition Network Language (CNL).

→ **How to Approach Developing a Rule Set** on page 324

This chapters is the second of three chapters about rule set development.

→ **Three Chapters About Rule Set Development** on page 123
→ **How to Learn Developing Rule Sets** on page 121

8.1 Use Hierarchical Image Object Levels

Although you can perform some image analysis on a single image object level, the full power of the **Definiens** object oriented image analysis unfolds when using multiple image object levels. On each of these levels, a number of objects is defined by the image objects on the level below that are considered their sub-objects. In the same manner, the lowest level image objects are defined by the pixels of the image that belong to them.

In this hierarchical structure, the image object hierarchy, each image object provides access to information about its neighbors, sub- and super-objects at all times. By connecting image objects vertically, access to scale and advanced texture properties is possible. The image object hierarchy allows the representation of image information at different spatial resolutions simultaneously.

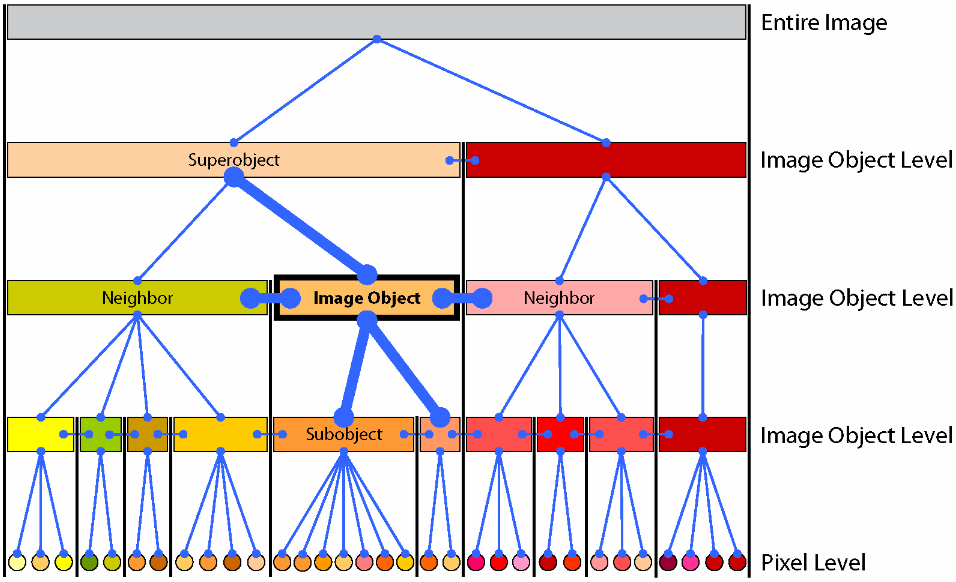


Figure 150: Within the image object hierarchy, each image object is linked to its neighbors, its superobject, and its subobjects.

The image object levels of an image object hierarchy range from fine resolution on the lowest image object level to coarse resolution on the highest image object level. On its superlevel, every image object has only one image object, the superobject. On the other hand an image object may have—but is not required to have—multiple subobjects.

→ Image Object Hierarchy on page 26
→ Navigate Within the Image Object Hierarchy on page 114

To better understand the concept of the image object hierarchy, imagine a meaningful hierarchy of image object levels, each representing a meaningful structure in an image. These meaningful image object levels are related to the various resolutions (coarse, medium, fine) of the image objects. The hierarchical positioning of the image object levels arranges subordinate image structures below generic image structures. See the following examples from biology and geography:

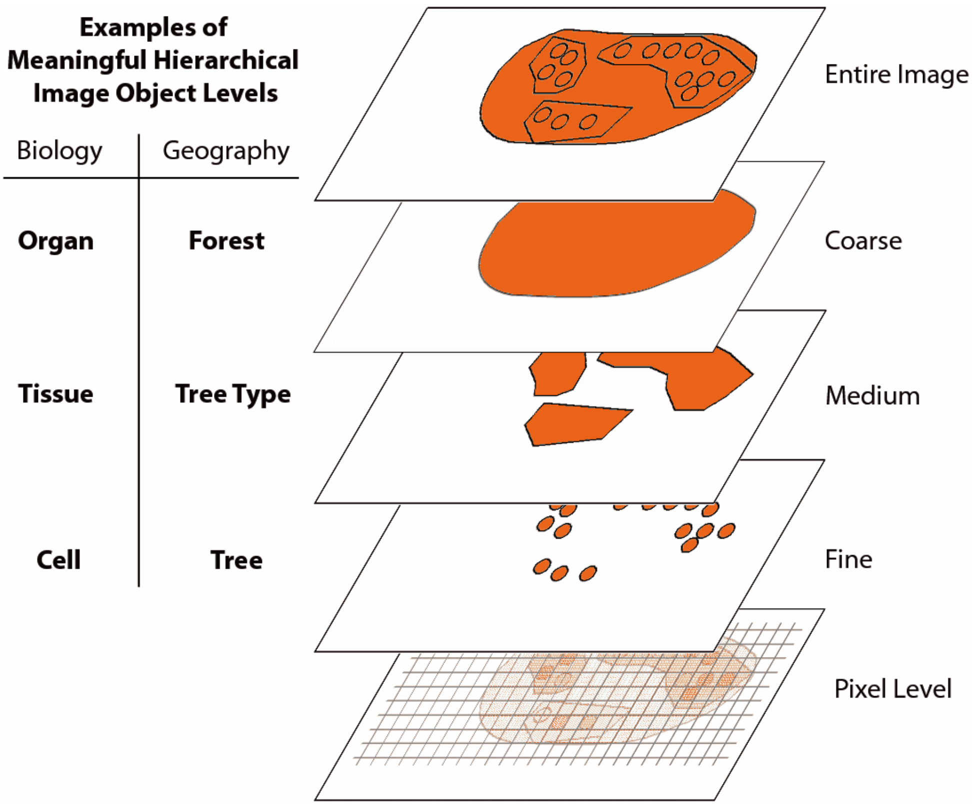


Figure 151: Meaningful image object levels within an image object hierarchy.

Two trivial image object levels are always implicitly given: the partition of the image into pixels, called the pixel level, and the level with only one object covering the entire image. Together they represent the boundaries of the image object hierarchy.

You can create an image object level by using either **segmentation algorithms** or the **copy image object level** algorithm. Commonly image object levels are added above the existing one. Some of the algorithms enable you to choose if the new image object level is inserted above or below the existing one.

→ Create Image Object Levels With Segmentation Algorithms on page 195
→ Duplicate Image Object Level on page 195

A new image object level can only be created between two already existing image object levels. The shapes of image objects on these super- and sublevels will constrain the shape of the objects in the new level.

The hierarchical network of an image object hierarchy is topologically definite. For example, the border of a superobject is consistent with the borders of its subobjects. The area represented by a specific image object is defined by the sum of its subobject's areas. The **Definiens** technology accomplishes this quite easily, because all segmentation techniques used in **Definiens Developer** use region-merging algorithms. For this reason, not all the algorithms used to analyze images allow a level to be created below an existing one. Each image object level is constructed based on its direct subobjects. For example, the subobjects one level are merged into larger image objects on the next higher level. This merge is limited by the borders of exiting superobjects; adjacent image objects cannot be merged if they have different superobjects.

8.1.1 Create Image Object Levels With Segmentation Algorithms

You can create an image object level by using different **segmentation algorithms**.

→ **Create Image Objects by Segmentation** on page 125
→ **Algorithms Overview** on page 155
→ **Define the Image Object Domain** on page 135

1. You can select relevant settings within the **Edit Process** dialog box.
2. Go to the drop-down list box within the **Image Object Domain** group box and select an available image object level. To switch to another image object level, select the currently active image object level and click the **Parameters** button to select another image object level in the **Select Level** dialog box.
3. For some segmentation algorithms, you choose to insert the new image object level either above or below the one selected in the image object level. Go to the **Algorithm Parameters** group box and look for a **Level Usage** parameter. If available, you can select from the options; if not available the new image object level is created above.

→ **Specify Algorithm Parameters** on page 134

Find Out More

Use Segmentation Algorithms to Create an Image Object Hierarchy

*Because **Definiens Developer** uses a pairwise merging algorithm, every segmentation uses the image objects of the next lower image object level as building blocks that are subsequently merged to segments of the new image object level. At the same time, the object borders of the next higher image object level are stringently obeyed. For this reason, creating new image object levels works with the following restrictions:*

••It is not possible to build an image object level that contains image objects larger than its superobjects.

••It is also not possible to build a level containing objects smaller than its subobjects.

••When creating the first image object level, the lower limit of the image objects size is represented by the pixels, the upper limit by the scene size.

This structure enables you to create an image object hierarchy by segmenting the image multiple times, resulting in different image object levels with image objects of different scales.

→ ***Segmentation Principles: Top-down and Bottom-up*** *on page 159*

8.1.2 Duplicate Image Object Level

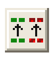
Duplicate an image object level to create a copy of an existing image object level within the image object hierarchy. Duplicating image object levels is an often-used step in image analysis. It commonly introduces both the specialization and the generalization of image objects. Thus duplicating contributes to the extension and refinement of the image object hierarchy. In addition, it is used for testing purposes while developing rule sets.

Precondition: The project already contains at least one image object level.

→ **Navigate Within the Image Object Hierarchy** on page 114

To duplicate an image object level do one of the following:

- Choose **Image Objects > Copy Current Level** on the main menu bar. The new



Copy Level

image object level is inserted above the currently active one.

- Create a process using the **copy image object level** algorithm. You can choose to insert the new image object level either above or below an existing one.

8.1.3 Edit an Image Object Level or Level Variable

By default, image object levels are named automatically (**Level 1**, **Level 2**, **Level 3**, ...) as they are created. They may also be assigned a name pre-defined by the user. You may want to rename an image object level name, for example, to prepare a rule set for further processing steps or to follow a naming convention.

You can also create or edit level variables and assign them to existing levels.

1. To edit an image object level or level variable choose **Image Objects** > **Edit Level Names** on the main menu bar. The **Edit Level Aliases** dialog box opens.

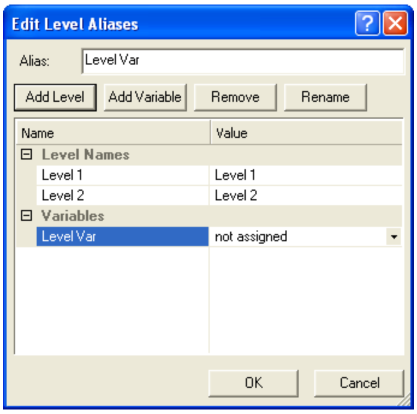


Figure 152: The Edit Level Aliases dialog box.

2. Select an image object **Level** or **Variable** and edit its **Alias**.

→ Create a Variable on page 258

3. To define a name of an fictive image object level, type a name in the **Alias** field and click the **Add Level** or **Add Variable** button adding a new **not assigned** item to the **Level/Variable** column. Select **not assigned** and edit its **Alias** or assign another **Level** from the drop-down box. After confirming the name with **OK**, the new level or variable is available to be assigned to a newly created image object level during process execution.
4. To assign a image object level or level variable to an existing value, select the item you want to assign and use the drop-down arrow to select a new value.

 (drop-down arrow button)

5. To remove a level variable alias, select it in the **Variable** area and click **Remove.**
6. To rename an image object level or level variable, select it in the **Variables** area, type a new alias in the **Alias** field, and click **Rename.**

Find Out More

Define Names of Image Object Levels in Advance

In some cases it might be helpful to define names of image object levels before they are assigned to newly created image object levels during process execution.

To do so, use the ***Add New*** *button within the* ***Edit Level Names*** *dialog box.*

*Alternatively, you can use the **Image Object Domain** group box of the **Edit Process** dialog box: In the drop-down list, choose the currently selected image object level (or **image object level** if none is selected). Then click the **Parameter** button to open the **Select Level** dialog box. Right-click in the **Select Level** dialog box and select **Create New Level Name**. In the opening dialog box you can enter the name of the new image object level. Repeat to define multiple names.*

→ *Define the Image Object Domain on page 135*

8.1.4 Delete Image Object Level

Delete an image object level. This enables you to work with image object levels that are temporary, or that might be required for testing processes while developing rule sets.

To delete an image object level do one of the following:

- Create a process using the **delete image object level** algorithm.
- Choose **Image Objects > Delete Levels** on the main menu bar.

Delete Level

1. The opening **Delete Level** dialog box displays a lists of all image object levels according to the image object hierarchy.

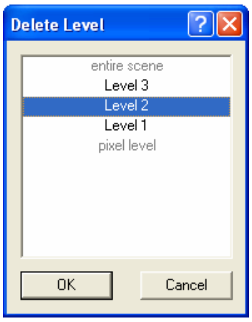


Figure 153: Delete Level dialog box

2. Select the image object level to be deleted. To select more than one level you may press Ctrl.
3. Confirm with **OK**. The selected image object levels will be removed from the image object hierarchy.

→ Options on page 452

8.2 Focus Processes by Using the Domain Concept

The domain concept of **Definiens Developer** is that any operation on image objects can be restricted to a selection of image objects within an image object level. Such a selection of image objects is called an **image object domain**. As a component of a **Definiens** process it allows you to focus the application of algorithms on a defined set of image objects.

You can define the image object domain by restricting the selection of image objects. To do so, you simply may apply a class filter or a feature filter within the **Edit Process** dialog box. That means that you choose one or more classes and/or select a condition on image object features, for example **area < 500** or **brightness > 190**.

Use Child Domains (Subdomains) for Local Processing

Child domain (or sometimes referred to as **subdomains**) are a more complex, but effective, way of restricting the image object domain: you define one image object domain first (you may call it the parent image object domain), then define a set of image objects by their relation to image objects in that parent image object domain. For example you have found Cell image objects on the second lowest image object level in a previous step; now you want to do an operation on the subobjects of those Cell image objects. You can define all those subobjects as your image object domain, or further restrict them by adding another class or feature filter. Since the image object domain is now defined by their relationship to an image object in a super-ordinate image object domain, it is referred to as child domain (subdomain). From the perspective of each image object in that child domain, the image object defined in the super-ordinate image object domain is referred to as the **Parent Process Object (PPO)**. Doing operations on a child domain (subdomain) is also called **local processing**.

Process-related features are a more powerful tool that comes with local processing. When a child process line is executed, the image objects in the image object domain 'know" their parent process object. It can be very practicable to be able to directly compare properties of those image objects to the properties of the PPO. A new group of features, the process-related features, do exactly this job.

Process-related features are features where one image object is analyzed with respect or in comparison to precisely one other image object, the PPO:

- **In comparison** means the difference or ratio of any object feature between the two objects.
- **With respect to** can for example be the common border length between the two objects.

8.2.1 Implement Child Domains via the Execute Child Process Algorithm

Learn how to implement child domains (subdomains) into rule sets. The focused processing of child domains is realized by the Execute child process algorithm. This algorithm allows a hierarchical structure of the rule set expressed by the process tree.

Precondition: Before defining a child domain, you need to have a hierarchical process tree, that is with at least one parent and one child process, as a prerequisite.

To create child processes within the **Process Tree** window, just select the **Insert Child** command on the right-click context menu. In the drop-down list box of the **Image Object Domain** group box of the **Edit Process** dialog box do not define an image object level. Instead, use one of the four local processing options. Depending on your selection, the child domain consists of image objects related to the parent image object in the manner described:

- **Current image object**: The parent image object itself.
- **Neighbor obj.**: Neighbors of the parent image object.
  If the distance setting is **dist=0**, this refers to image objects that have a common border with the parent image object and lie on the same image object level.
  If the distance defined is non-zero (**dist=n≠0**), it refers to all image objects whose center of mass has a distance to the parent's center of mass is below the threshold given.

→ Spatial Distance on page 171

- **Sub objects**: Subobjects of the parent image object:
  This refers to image objects whose image area covers all or part of the parent's image area and that lie the specified number of image object levels below the

→ Level Distance on page 171

parent's image object level.

- **Super object**: The super image object of the parent image objects:
  This refers the image object whose image area covers at least the parent's image area and that that is located the specified number of image object levels above the parent's image object level. Note, that the child image object is on top here.

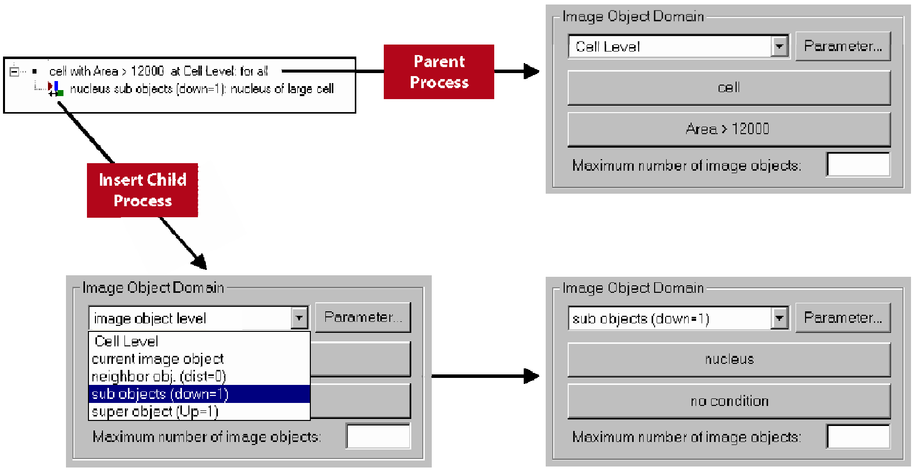


Figure 154: Example how to create a parent process and a child process with a subdomain. Here all subobjects of the Cell image objects specified in the parent process that are of class Nucleus are in the child domain (subdomain).

For a general demonstration of the relation of a child domain to the image object in the parent process, see the following graph:

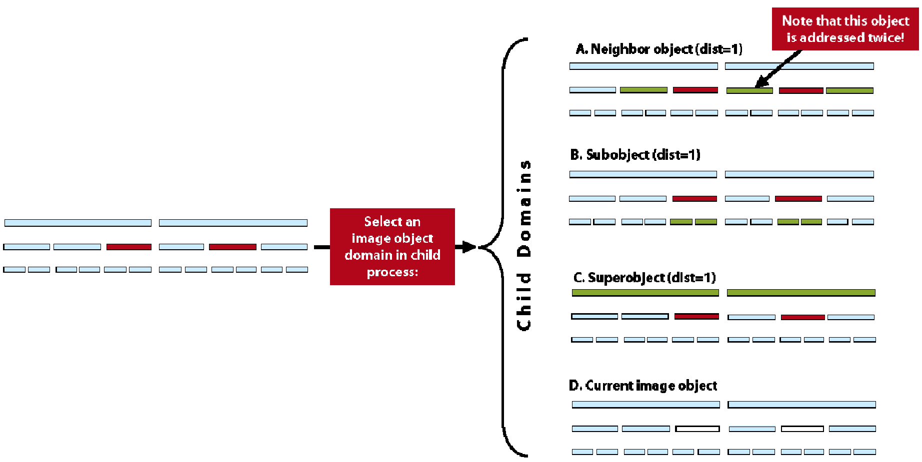


Figure 155: Schematic description of the four different types of subdomains.
Left: objects in the image object domain of the parent process in red; right: objects in the child domains (subdomains) of the two red objects in green. Note that in the case of neighbor object (dist=0), the center green object is in the subdomain of both the left and of the right red objects.

8.2.2 About Child Domains and Parent Processes

Find Out More

About Subdomains (Child Domains) and Subobjects

*A common source of confusion is the usage of the prefix **sub-** in the context of **subdomains** or better called **child domains**.*

*Remember that the prefix **sub-** in subdomain refers to the relative location of a process in the process tree.*

On the other hand, in the word subobject, the same prefix is used for the relative location of an image object in the image object hierarchy.

A mixed example: The subdomain (child domain) can consist of subobjects, but it can also consist of neighbor objects, the superobject, or the parent object itself.

*We recommend that you describe the relative location of processes in the hierarchical process tree by using the terms **child process** and **parent process**.*

Terminology

Terms used in the context of process hierarchy:	
Parent process	A parent process is used for grouping child processes together on a hierarchy level beneath the parent process within a process hierarchy.
Child process	A child process is grouped on a level beneath a parent process within a process hierarchy. .
Child domain (Subdomain)	An image object domain defined by using one of the four local processing options.
Parent process object (PPO)	A Parent process object (PPO) is the object defined in the parent process.
Terms used in the context of image object hierarchy:	
Superobject	An image object that has a superobject within an image object hierarchy.
Subobject	An image object that has subobjects within an image object hierarchy.

8.2.3 Use Parent Process Objects (PPO)

A parent process object (PPO) is an image object to that a child process refers to. The PPO has to be defined in the parent process before.

An image object can be called through the respective selection in the **Edit Process** dialog box: Go to the **Image Object Domain** group box and select one of the four local processing options from the drop-down list box; for example, **Current image object**.

→ Implement Child Domains via the Execute Child Process Algorithm on page 198

When you do local processing, the routine goes to the first random image object described in the parent domain and processes all child processes defined under the parent process, where the parent process object (PPO) is always that same image object.

Then the routine moves to the next image object in the parent domain and so on. The routine does not update the parent domain after each processing of one parent process object; it will continue to process those image objects found to fit the parent process's image object domain criteria, no matter if they still fit them when they are to be executed.

A special case of parent process object (PPO) is the 0-th order parent process object (PPO), also referred to as **PPO(0)**. Here the parent process object (PPO) is the image object defined in the image object domain in the same line (0 lines above).

8.2.3.1 Example: ParentProcessObjects.dpr

For better understanding of child domains (subdomains) and the usage parent process objects (PPO), we recommend opening an example project which is included within the installation.

Different Types of Child Domains

Understand the different types of child domains (subdomains) demonstrated in a sample project.

1. Open the test project **ParentProcessObjects.dpr** which is included within the installation. The path is:
   C:\Program Files\Definiens *Version* **number\Examples \ParentProcessObjectPPO\ParentProcessObjects.dpr**.

   Image data courtesy of European Molecular Biology Labaratory (EMBL) Heidelberg.
2. Run the Preparation part of it. This finds the complete cell in the image on the upper (Cell-) level and Nucleus, Cytoplasm and Golgi on the lower (Organelle-) level. In addition, two backup levels are created that reverse each demonstration step after running it.
3. Before executing each step, take a look at the different object levels and the objects therein, and make a prediction on which object will be assigned the class **MyClass**.
4. Run the subsequent **Refresh** process to get back to the state before running the exercise (step 3).

Use Parent Process Objects (PPO) for Local Processing

The next example demonstrates how local processing is used in order to change the order in which class or feature filters are applied. During execution of each process line, **Definiens** software first creates a list of image objects that are defined in the image object domain. Then the desired routine is executed for all image objects on the list.

1. If not yet done, open the test project **ParentProcessObjects.dpr** which is included within the installation. The path is:
   C:\Program Files\Definiens Developer *Version number***\Examples \ParentProcessObjectPPO\ParentProcessObjects.dpr**.

   Image data courtesy of European Molecular Biology Labaratory (EMBL) Heidelberg.

2. In the first parent process named **simple use** under **Try out PPO**, you can compare the results of the **Assign class** algorithm with and without using the parent process object (PPO).

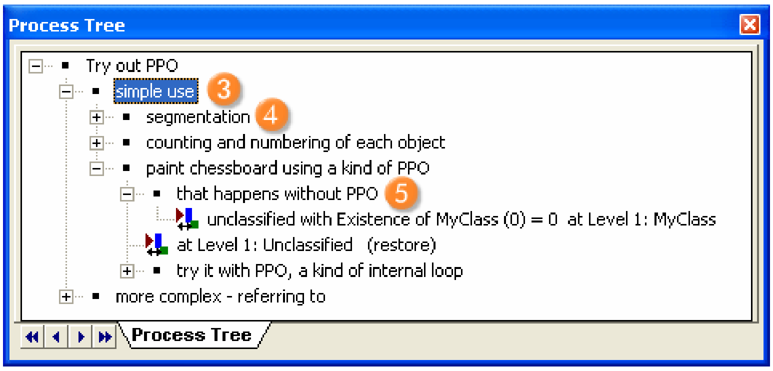


Figure 156: Process Tree window of the example project ParentProcessObjects.dpr.

3. Execute **segmentation** 3 process.

4. Execute the process **counting and numbering of each object** 4.

5. Execute **that happens without PPO** 5 process using the **Assign class** algorithm. The result without a parent process object (PPO) will be that the whole image has been classified:

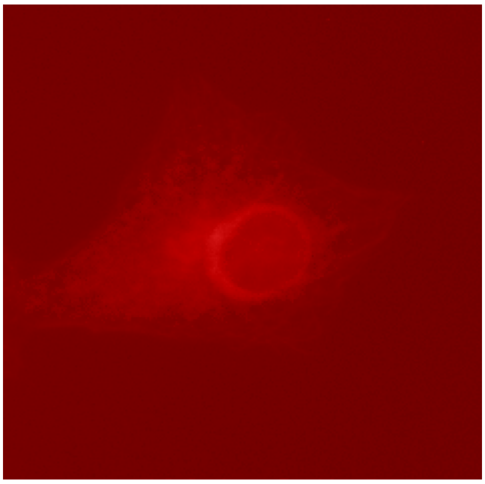


Figure 157: Result without parent process object (PPO). (Image data: Courtesy of EMBL Heidelberg)

6. That is due to the fact that before processing the line, no objects of class **MyClass** exists, so all objects in **Level 1** return true for the condition that no **MyClass** objects exist in the neighborhood. In the next block, the two pieces of information defining the image object domain, **objects on Level 1** and **no MyClass objects exist in the neighborhood** are split into two different lines. Still, you will not get the same result.

7. Execute the process **at Level1: Unclassified (restore)** 7 to remove the classification to come back to the situation after step 3.

8. Execute the process **try it with PPO, a kind of internal loop** 8.

The process **if with Existence of MyClass (0) = 0 : MyClass** applies the algorithm **Assign class** to the image object that has been set in the parent process **unclassified at Level 1: for all.** This has been invoked by selecting **Current image object** 8 as image object domain. Thus, all unclassified image objects will be called and each unclassified image object will be treated separately.

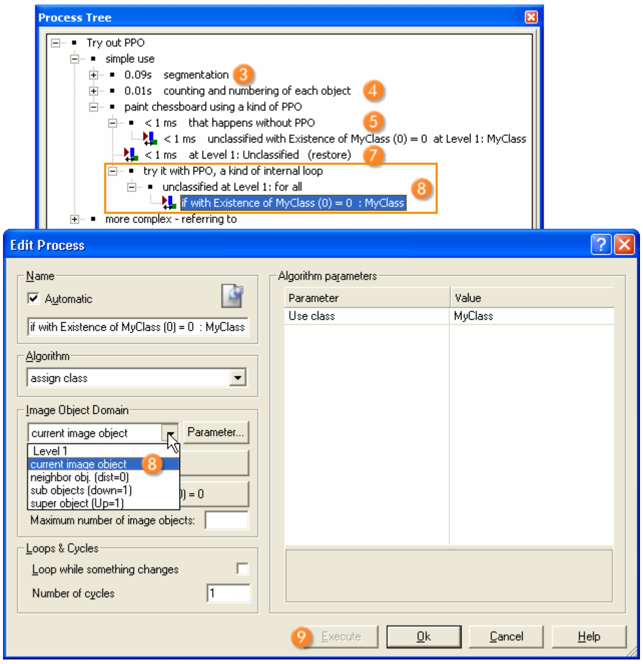


Figure 158: Setting with parent process object (PPO), a kind of internal loop.

Note

Algorithms that are referring to a parent process object (PPO), must be executed from the parent process. That means, you have to execute the parent process itself or in between a superordinated parent process.

9. Execute 9 and the result is a painted chessboard:

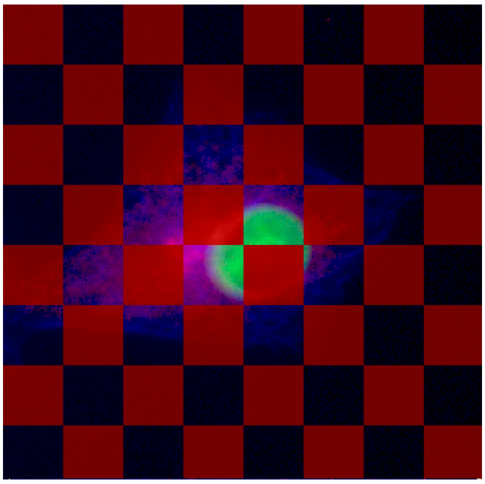


Figure 159: Result with parent process object (PPO), a kind of internal loop.

10. In the first line, all objects on image object **Level 1** are put in a list. The process does nothing but pass on the identities of each of those image objects down to the next line, one by one. That second line - the child process - has only one object in the

image object domain, the current image object passed down from the parent process. It then checks the feature condition, which returns true for the first object tested. But the next time this process is run with the next image object, that image object is tested again and returns false for the same feature, because now the object has the first object as a **MyClass** neighbor.

11. To wrap up the difference in one sentence: in the first example, one list with 64 objects that fit the condition is created at once; in the second example, one list of 64 is created in the upper process line, but then 64 lists of 1 or 0 image objects each are created in the 64 run-times of the child process.

12. In other words: The result with the usage of the parent process object (PPO) is totally different than without using it. The case is that using the parent process object (PPO) will process each image object in the image in succession. That means: the algorithm checks for the first unclassified image object complying with the set condition which is **Existence of MyClass (0) = 0**). The image object identifies that there is no **MyClass** neighbor, so it classifies itself to **MyClass**. Then the algorithm goes to the second unclassified image object and finds a neighbor, which means the condition does not fit. Then it goes to the third, there is no neighbor, so it classifies itself, and so on.

13. For verification we generated an object variable to give each object a number, with the parent process called **counting and numbering of each image object**. Select the variables to be loaded to the **Image Object Information** window, and select an image object in your image.

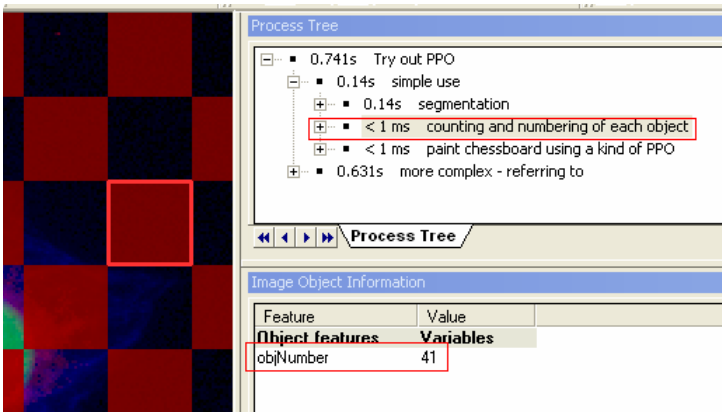


Figure 160: The number of the selected object here is 41.

Use Process-Related Features for Advanced Local Processing

One more powerful tool comes with local processing. When a child process line is executed, the image objects in the image object domain "know" their parent process object. It can be very practicable to be able to directly compare properties of those image objects to the properties of the PPO. A new group of features, the **Process-Related** features, do exactly this job. To learn about them, continue with the example project which is included within the installation.

1. If not yet done, open the test project **ParentProcessObjects.dpr** which is included within the installation. The path is:
C:\Program Files\Definiens Developer *Version* **number\Examples \ParentProcessObjectPPO\ParentProcessObjects.dpr**.

2. Execute each child process from the next rule set called **more complex - referring to**. You should switch to the outline view after doing the basic segmentation.

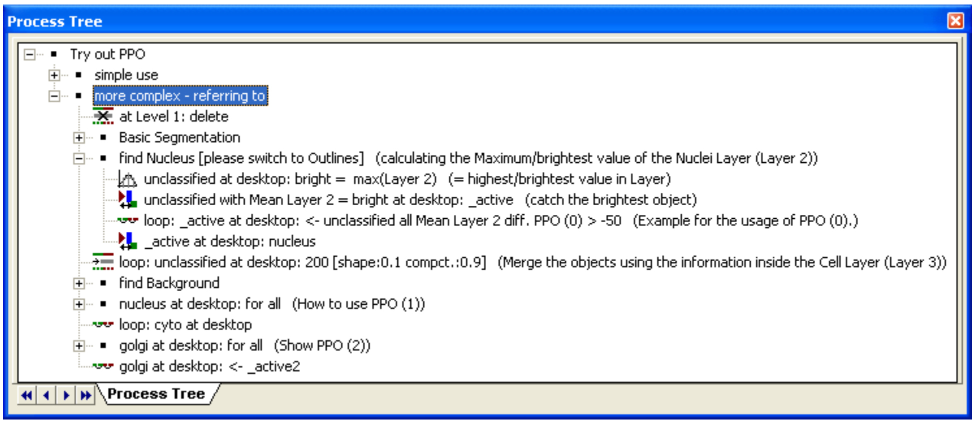


Figure 161: A more complex usage of parent process object (PPO) creating customized parent process objects.

In this rule set the PPO(0) procedure is used to merge the image objects with the brightest image object (classified as: **_active**) in the **nucleus** image layer (**Layer 2**). For this purpose a difference range (**> -50**) to the **_active** image object is used.

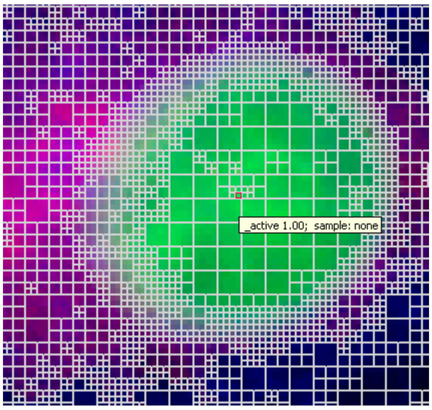


Figure 162: The brightest image object in this image layer. (Image data: Courtesy of EMBL Heidelberg)

3. The red bordered image object (**_active**) in the image is the brightest image object in this image layer. To find out the difference to the similar image objects you might want to merge with, please select it using the Ctrl key. Doing that, you have set manually your parent process object (PPO). The parent process object (PPO) will be highlighted in green.

4. Hide the outlines and go to the **Feature View** window in the **Mean Layer 2 diff. PPO (0)** feature. Check the values in the **Image Object Information** window, to find the best fitting range for the difference to the brightest object (**_active**).

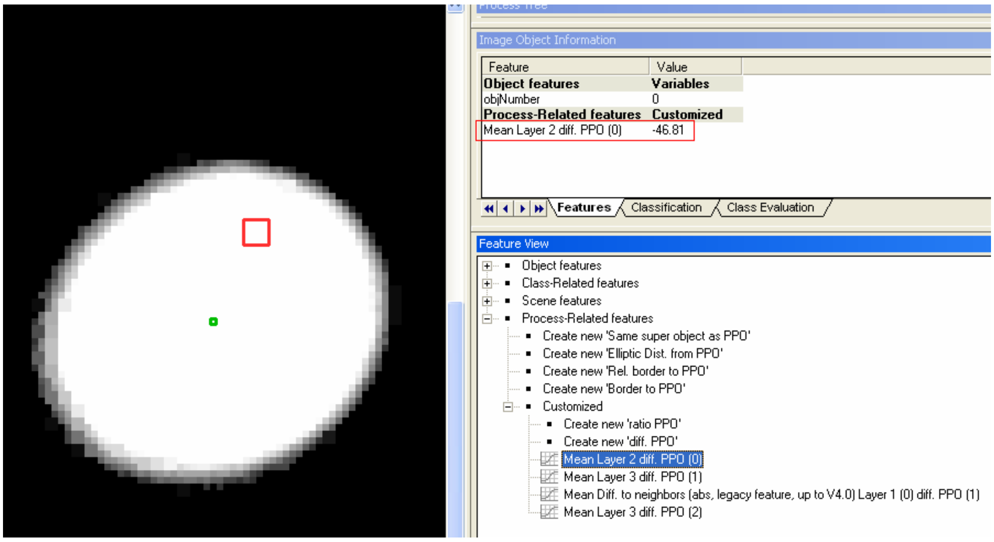


Figure 163: Compare the difference between the red highlighted image object and the green highlighted parent process object (PPO).

The green highlighted image object displays the parent process object (PPO). All other image objects that are selected will be highlighted in red and you can view the difference to the green highlighted image object in the **Image Object Information** window.

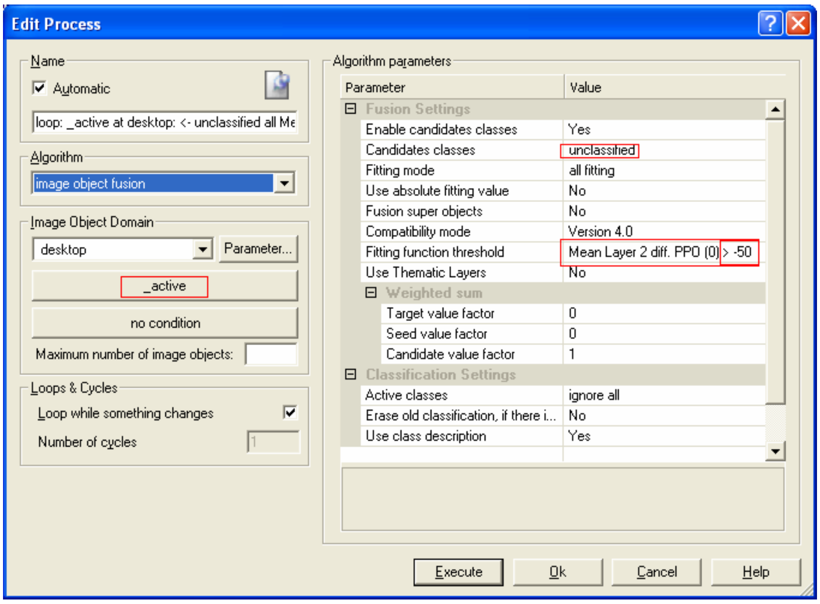


Figure 164: Edit Process dialog box with settings for using the difference to the parent process object (PPO), doing an image object fusion.

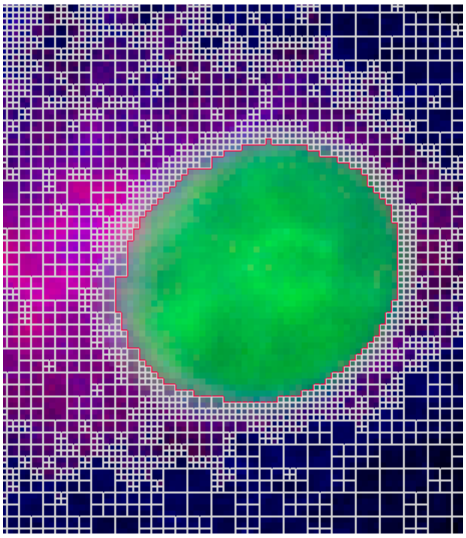


Figure 165: Result after doing an image object fusion using the difference to the PPO(0).

5. All process-related features have to be created in order to be used. For features that set an image object in relation to the parent object only an integer number has to be specified, the distance **(Dist.)** It refers to the distance in the process hierarchy, that is that number of lines above the present line, in which you find the definition of the parent object. This is true for the following features:

   - **Same super object as PPO**
   - **Elliptic Distance from PPO**
   - **Rel. border to PPO**
   - **Border to PPO**

   For the following process-related features that compare an image object to the parent object the distance **(Dist.)** has to be specified as well:

   - **Ratio PPO**
   - **Diff PPO**

   In addition, you have to select the feature that you want to be compared. For example, if you create a **new ratio PPO**, select **Distance=2** and the feature **Area**, the created feature will be **Area ratio PPO (2)**. The number it returns will be the area of the object in question divided by the area of the parent process object of order 2, that is the object whose identity was handed down from two lines above in the process tree.

   A special case are process-related features with **Distance=0**, called PPO(0) features. They only make sense in processes that need more than one image object as an input, for example image object fusion. You may have a PPO(0) feature evaluated for the candidate or for the target image object, where that feature is then compared/put to relation to the image object in the image object domain of the same line, that is the seed image object of the image object fusion.

Create a **process-related feature** sometimes referred to as PPO feature. Go to the **Feature View** window and expand the **Process-Related features** group.

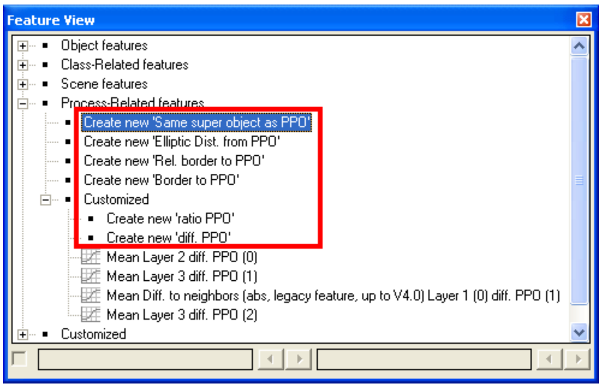


Figure 166: Process-Related features can be used for parent process objects (PPO).

To create a process-related feature (PPO feature), double-click on the feature you want to create and add a distance to the parent process object. The distance is a hierarchical distance in the process tree, for example:

- PPO(0), has the distance 0, this refers to the image object in the current process, that is mostly used in the algorithm **image object fusion**.
- PPO(1), has the distance 1, this refers to the image object in the parent process one level above.
- PPO(2), has the distance 2, this refers to the parent process two levels above in the process hierarchy.

If you want to create a **Customized** PPO, you also have to choose a feature.

6. The next processes in the rule set are using different parent process object hierarchies. Applying them is the same procedure as shown before with the PPO(0).

8.3 About Classification

This chapter gives you an overview of the various concepts and terms of classification.

The production of image objects in **Definiens** software is not a goal in itself. Image objects are rather basic information carriers for subsequent classification steps. Compared to single pixels, image objects provide additional features aplenty and a significantly increased signal to noise ratio. Based on the generation and classification of image objects, you can analyze and interpret complex images.

→ Classify Image Objects on page 190

To perform a classification, classes need to be defined and depending on the applied approach, algorithms, thresholds, or class descriptions need to be selected. During the classification process, image objects are analyzed according to the defined criteria and assigned to the class which meets the defined criteria.

→ Introduction to Classification on page 190
→ Algorithms for Process-Based Classification on page 208
→ About Classification with Class Descriptions on page 212

8.3.1 Algorithms for Process-Based Classification

Classification algorithms analyze image objects according defined criteria and assign them each to a class that best meets the defined criteria.

When editing processes, you can choose from the following classification algorithms:

→ About Process-Based Classification on page 190

- **Assign class**: assigns a class to image object with certain features.
- **Classification**: uses the class description to assign a class
- **Hierarchical classification**: uses the class description as well as the hierarchical structure of classes.
- **Advanced Classification Algorithms**: are designed to perform a specific classification task like finding extrema or identifying connections between objects.

8.3.1.1 Assign Class Algorithm

The **Assign Class** algorithm is the most simple classification algorithm. It determines by means of a threshold condition whether the image object is a member of the class or not.

This algorithm is used, when one threshold condition is sufficient to assign an image object to a class.

1. In the **Edit Process** dialog box, select **assign class** from the **Algorithm** list.
2. In the **Image Object Domain** group click the **condition** button; it is labeled **no condition** when no condition is yet selected.
3. Select a feature for the condition and define the operator and the reference value.
4. For the algorithm parameter **Use class**, select a class or create a class by selecting the entry **<create new class>**.

→ Use Thresholds for Classification with Class Descriptions on page 212

8.3.1.2 Classification Algorithm

The **Classification** algorithm uses class descriptions to classify image objects. It evaluates the class description and determines on this basis whether an image object can be a member of this class.

Classes without a class description are assumed to have a membership value of 1.

This algorithm is used if you want to use the fuzzy logic of membership functions or if you have combined conditions in a class description.

Based on the calculated membership value, information about the three best fitting classes are stored in the image object classification result. Thus, you can see what other classes this image object would fit and possibly fine-tune your settings.

→ Review Results of a Classification on page 255

1. In the **Edit Process** dialog box, select **classification** from the **Algorithm** list.
2. Define the image object domain.
3. For the algorithm parameters, select **actives classes** which can be assigned to the image objects.
4. Select **Erase old classification, if there is no new classification** to define that the existing classification of the selected domains' objects are removed if the image objects do not match the class description.
5. Select **Use Class Description** if you want to use the class description for classification. Class descriptions are evaluated for all classes. An image object is assigned to the class with the highest membership value.

→ About Classification with Class Descriptions on page 212

8.3.1.3 Hierarchical Classification Algorithm

The **hierarchical classification** algorithm is used to apply complex class hierarchies to entire image object levels.

It is specifically tailored to allow applying **eCognition** version 4 and older class hierarchies without major changes. This algorithm

This algorithm can be applied to an entire set of hierarchically arranged classes. It thereby applies a predefined logic activating respectively deactivating classes based on the following rules:

A. Classes are not applied to the classification of image objects whenever they contain applicable child classes within the inheritance hierarchy.

   Parent classes pass on their class descriptions to child classes. These child classes then add additional feature descriptions and—if they are not parent classes themselves—are meaningfully applied to the classification of image objects. The above logic is following the concept that child classes are used to further divide a more general class. Therefore when defining subclasses for one class always keep in mind that not all image objects defined by the parent class are automatically defined by the subclasses. If there are objects which would be assigned to the parent class but none of the descriptions of the subclasses fit those image objects, they will be assigned neither to the parent nor to the child classes.

B. Classes are only applied to a classification of image objects, if all contained classifiers are applicable.

   The second rule applies mainly to classes containing class related features. The reason for this is that you might generate a class which describes objects of a certain spectral value in addition to certain contextual information given by a class related feature. The spectral value taken by itself without considering the context would cover far too many image objects, so that only a combination of the two would lead to satisfying results. As a consequence, when classifying without class related features, not only the expression referring to another class but the whole class is not used in this classification process.

Contained and inherited expressions in the class description produce membership values for each object and according to the highest membership value, each object is then classified.

→ **About Inheritance** on page 252

If the membership value of an image object is lower than the pre-defined minimum membership value, the image object remains unclassified. If two or more class descriptions share the highest membership value, the assignment of an object to one of these classes is random.

→ **Edit Minimum Membership Value** on page 218

The three best classes are stored as the image object classification result.

> **Note**
>
> Unlike the **Classification** algorithm, classes without a class description are assumed to have a membership value of **0**.

Class-related features are considered only if explicitly enabled by the corresponding parameter.

How to Use With a Process

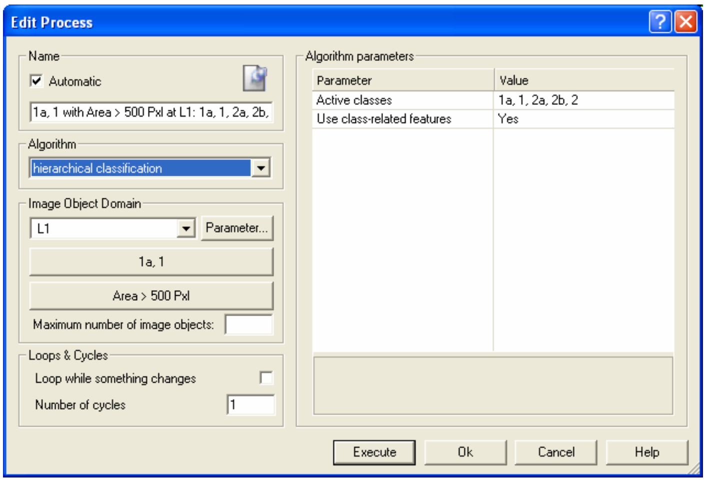


Figure 167: Settings for the Hierarchical Classification algorithm.

1. In the **Edit Process** dialog box, select **hierarchical classification** from the **Algorithm** drop-down list box.
2. Define the **Image object domain**, if necessary.
3. For the **Algorithm parameters**, select the **Actives classes** that can be assigned to the image objects.
4. Select if you want to **Use class-related features**.

8.3.1.4 Advanced Classification Algorithms

Advanced classification algorithms are designed to perform specific classification tasks.

All advanced classification settings allow defining the same classification settings as the classification algorithm. In addition algorithm specific settings need to be set. The following algorithms are available:

- **Find domain extrema** allows identifying areas which fulfill an extrema condition within the defined image object domain.
- **Find local extrema** allows identifying areas which fulfill a local extrema condition within the defined image object domain and within a defined search range around the object.
- **Find enclosed by class** finds objects which are completely enclosed by a certain class.
- **Find enclosed by object** finds objects which are completely enclosed by an image object.
- **Connector** classifies image objects which represent the shortest connection between objects of a defined class.

8.3.2 About Classification with Class Descriptions

There are different types of expressions and methods to use them in class descriptions:

- Thresholds
- Membership functions
- Similarities
- Use samples for Nearest Neighbor Classification

→ Use Thresholds for Classification with Class Descriptions on page 212
→ Use Membership Functions for Classification on page 213
→ Use Similarities for Classification on page 218
→ Use Samples for Nearest Neighbor Classification on page 219

Depending on the goal you want to achieve, you can mix the different types of expressions, combine them with logical operators, and put them in a logical order.

Review and Adapt

Because classification needs a lot of experience, many cycles of "trial and error" might accompany your work. We recommend that you start with a simple class description, for example, a single threshold, then classify or execute the process and review the results. Adapt your settings, classify again and review again. Do so until you are satisfied with the result.

Only then should you start adding additional thresholds, membership functions or similarity expressions to the class description. Combine them with logical operators. Test and review again and again.

In this way, you can build up complex classification scenarios with many - hierarchically ordered - classes and complex class descriptions.

8.3.3 Use Thresholds for Classification with Class Descriptions

You can use thresholds within the class description. A threshold condition determines whether an image object matches the condition or not. There is no fuzzy logic used. Typically, you use thresholds in class descriptions if classes can be separated clearly by a feature.

→ Insert an Expression on page 246

Theoretically, if there are no other expressions within the class description a threshold-based classification allows you to assign image objects to a class, based on only one condition. The actual advantage of using class descriptions however lies in the combination of several conditions.

The concept of threshold conditions is available for process-based classification too. There the threshold condition is part of the image object domain and thus can be added to most algorithms. This limits the execution of the respective algorithm to only those objects that fulfill this condition.

→ Define the Image Object Domain on page 135

An alternative to the threshold condition can be the **Larger than (Boolean)** or **Smaller than (Boolean)** membership functions.

→ Use Membership Functions for Classification on page 213

1. Go to the **Class Hierarchy** dialog box and double-click a class. Open the **Contained** tab of the **Class Description** dialog box.

→ Edit the Class Description on page 246

2. In the **Contained** area, right-click the initial operator **and(min)** and choose **Insert new Expression** on the context menu.

→ Insert an Expression on page 246

3. From the **Insert Expression** dialog box, select the desired feature. Right-click it and choose **Insert Threshold** from the context menu.

   The **Edit Threshold Condition** dialog box opens where you can define the threshold expression.

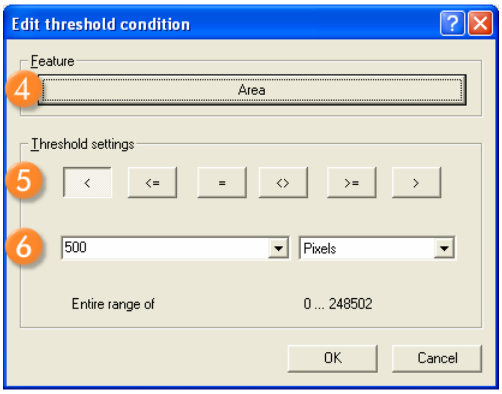


Figure 168: Edit Threshold Condition dialog box.

   In the **Feature** group box 4, the feature that has been selected to define the threshold is displayed on a broad button. To select a different feature, click this button.

4. Select a logical operator 5.

→ About Operators on page 245

5. Enter the number defining the threshold 6. With **Definiens Developer**, you can alternatively select a process variable, if one exists. For constants, you can define the unit to be used. The feature range is displayed below.

6. Click **OK** to apply your settings.

7. The resulting logical expression is displayed in the class description, for example with the values of this example **Area < 500 pxl**.

Tip

Defining a threshold conditions is available for process based classification too. There it is part of the image object domain.

→ ***Define the Image Object Domain*** *on page 135*

8.3.4 Use Membership Functions for Classification

Membership functions offer a transparent relationship between feature values and the degree of membership to a class.

A membership function is defined by its **Left** and **Right border** values in combination with the function slope.

1. To open the **Membership Function**, do one of the following:

- Insert a new expression into a class description.

→ Insert an Expression on page 246

- Right-click an existing expression in a class description and choose **Edit Expression** from the context menu.

→ Edit an Expression on page 249

The **Membership Function** dialog box opens.

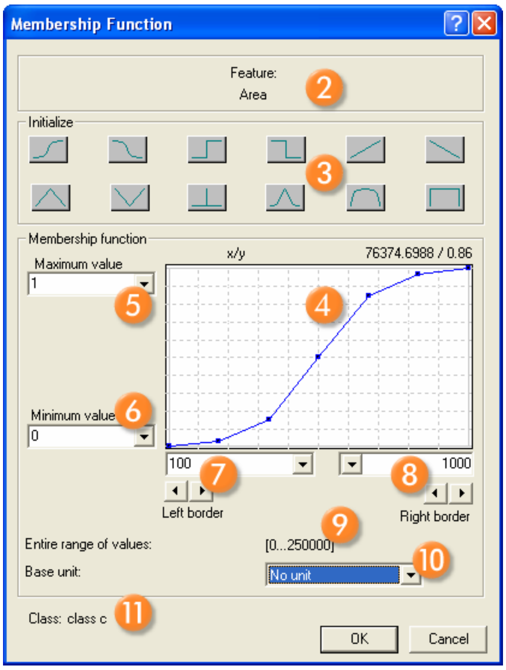


Figure 169: The Membership Function dialog box.

The **Feature** group box 2 displays the name of the selected feature.

2. In the **Initialize** group box 3, select the pre-defined membership function type you want to use.

→ Membership Function Type on page 215

3. The function curve is displayed in the **Membership function** group box 4. You can edit the function by dragging the vertices thus integrating special knowledge. However, this is usually not necessary. It is recommended that you use membership functions which are as general and broad as possible while still ensuring satisfying results on a given data set. This method will result in more robust rule bases and will make it easier to transfer your rule base to other data sets.

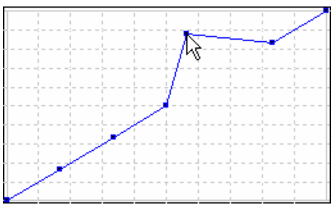


Figure 170: Drag-and-drop a nodal point to modify a membership function.

4. Edit the **Maximum value** 5 of the resulting membership function. The membership value on the x-axis for the sample function slope is **1**.

5. Edit the **Minimum value** 6 of the resulting membership function. The membership value on the y-axis for the sample function slope is **0**.

6. Edit the **Left border** ⑦ of the transition zone of the membership function. The membership value for the left border will be assigned to all feature values equal to or smaller than the left border value.
7. Edit the **Right border** ⑧ of the transition zone of the membership function. The membership value for the right border will be assigned to all feature values equal to or larger than the right border.
   In the above example, the part of the range which is fuzzy (i.e., not definitely **yes** or **no**) lies between 100 and 1000, as defined by the **Left** and **Right border** values. This results in the following membership values for image objects:
   - 0 for a feature value of 100 and below,
   - 0.5 for a feature value of 450 (i.e., half-way between 100 and 1000),
   - 1 for a feature value of 1000 and above.
8. The **Entire range of values** ⑨ displays the possible value range for the selected feature. In the example, the range is from 0 to 250000.
9. For certain features, you can edit the **Base unit** ⑩.

   The bottom of the dialog box contains the name of the class you are currently editing ⑪.
10. Click **OK** to save your settings.

8.3.4.1 About Fuzzy Logic

Fuzzy logic is a mathematical approach to quantify uncertain statements.

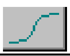
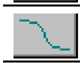
The basic idea is to replace the two strictly logical statements **Yes** and **No** by the continuous range of **[0…1]**, where 0 means **Exactly no** and 1 means **Exactly yes**. All values between 0 and 1 represent a more or less certain state of **Yes** and **No**.

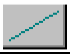
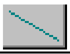
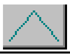
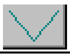
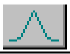
Thus, fuzzy logic is able to emulate human thinking and take into account even linguistic rules. Fuzzy classification systems are well suited to handling most vagueness in information extraction from images.

Instead of the binary **True** and **False** multivalued fuzzy logic allows transitions between **True** and **False**. Additionally, there are more or less strict realizations of the logical operations **And** or **Or**. The output of a fuzzy classification system is a fuzzy classification, where the membership degree to each class is given for each object. This enables detailed performance analysis and gives insight into the class mixture for each image object, which is a major advantage. The maximum membership degree determines the final classification to build an interface to crisp (boolean) systems.

8.3.4.2 Membership Function Type

For your membership function, you can select from a list of pre-defined functions.

Button	Function Form
	Larger than
	Smaller than
	Larger than (Boolean, crisp)
	Smaller than (Boolean, crisp)

	Larger than (linear)
	Smaller than (linear)
	Linear range (triangle)
	Linear range(triangle inverted)
	Singleton (exactly one value)
	Approximate Gaussian
	About range
	Full range

8.3.4.3 Generate Membership Functions Automatically

In some cases, especially when classes can be clearly distinguished, it is convenient to automatically generate membership functions. This can be done within the **Sample Editor** window.

→ Work with the Sample Editor on page 226

Precondition: The **Sample Editor** window has to be open and samples for the active class have to be selected.

1. To generate a membership function, right-click the respective feature in the **Sample Editor** window and select **Membership Functions > Compute**.

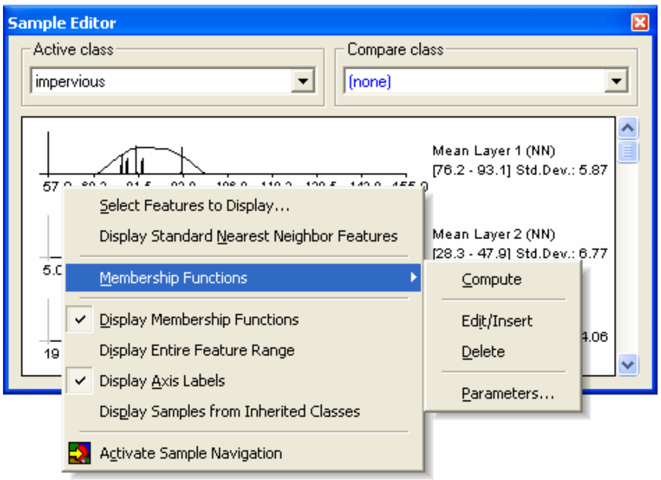


Figure 171: Sample Editor with generated membership functions and context menu.

Membership functions can also be inserted and defined manually in the **Sample Editor** window. To do this, right-click a feature and select **Membership Functions > Edit/Insert** which opens the **Membership Function** dialog box. This also allows you to edit an automatically generated function.

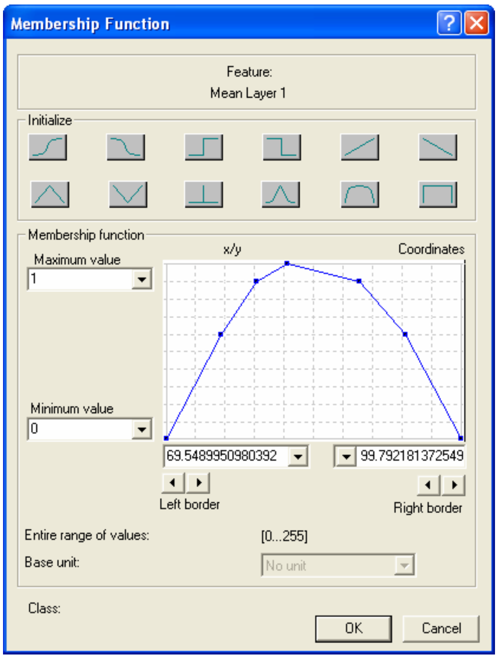


Figure 172: Automatically generated membership function.

2. To delete a generated membership function, select **Membership Functions > Delete**.
3. Switch the display of generated membership functions on or off by right-clicking in the **Sample Editor** window and activating or deactivating **Display Membership Functions**.

Edit Membership Function Parameters

You can edit parameters of a membership function computed from sample objects.

1. In the **Sample Editor**, select **Membership Functions > Parameters** from the context menu.
   The **Membership Function Parameters** dialog box opens.

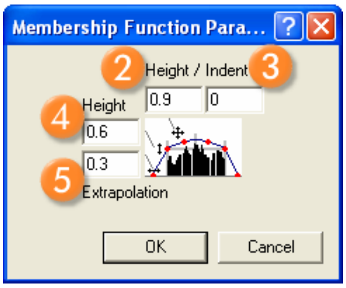


Figure 173: The Membership Function Parameters dialog box.

2. Edit the absolute **Height** 2 of the membership function.
3. Modify the **Indent** 3 of membership function.
4. Choose the **Height** 4 of the linear part of the membership function.
5. Edit the **Extrapolation** 5 width of the membership function.

8.3.4.4 Edit Minimum Membership Value

The minimum membership value defines the value an image object must reach to be considered a member of the class.

If the membership value of an image object is lower than the pre-defined minimum membership value, the image object remains unclassified. If two or more class descriptions share the highest membership value, the assignment of an object to one of these classes is random.

1. To change the default value of **0.1**, open the **Edit Minimum Membership Value** dialog box. Choose **Classification > Advanced Settings > Edit Minimum Membership Value** on the main menu bar.

Figure 174: The Edit Minimum Membership Value dialog box.

2. Edit the **Minimum membership value** for the classification.
3. Confirm with **OK**.

8.3.4.5 Use Similarities for Classification

→ About Inheritance on page 252

Similarities work like the inheritance of class descriptions. Basically, adding a similarity to a class description is equivalent to inheriting from this class. However, since similarities are part of the class description, they can be used with much more flexibility than an inherited feature. This is particularly obvious when they are combined by logical terms.

A very useful method is the application of inverted similarities as a sort of negative inheritance: consider a class **bright** if it is defined by high layer mean values. You can define a class **dark** by inserting a similarity feature to bright and inverting it, thus yielding the meaning **dark** is not **bright.**

It is important to notice that this formulation of **dark** is not **bright** refers to similarities and not to classification. An object with a membership value of 0.25 to the class **bright** would be correctly classified as **bright.** If in the next cycle a new class **dark** is added containing an inverted similarity to **bright** the same object would be classified as **dark** since the inverted similarity produces a membership value of 0.75. If you want to specify that **dark** is everything which is not classified as **bright** you should use the feature **Classified as**.

Similarities are inserted into the class description like any other expression.

8.3.5 Use Samples for Nearest Neighbor Classification

The use of the Nearest Neighbor classifier is recommended when you either need to make use of a complex combination of object features or your image analysis approach has to follow a set of defined sample areas.

The principle is simple: first, the software needs samples, typical representatives for each class. After a representative set of sample image objects has been declared for each class, the algorithm searches for the closest sample image object in the feature space for each image object. If an image object's closest sample object belongs to a certain class, the image object will be assigned to this class.

The nearest neighbor classifier utilizes a selection of features and is trained by sample image objects. In comparison to pixel-based training, the object-based approach of the nearest neighbor requires fewer training samples: one sample image object already covers many typical pixel samples and their variations. The nearest neighbor classifies image objects in a given feature space and with given samples for the classes of concern.

For advanced users, the **Feature Space Optimization** function offers a method to mathematically calculate the best combination of features in the feature space.

→ Optimize the Feature Space on page 237

To classify image objects using the Nearest Neighbor classifier, follow the recommended workflow:

1. Load or create classes.

→ About Classes on page 241

2. Define the feature space.

→ Define the Feature Space with Nearest Neighbor Expressions on page 221

3. Define sample image objects.

→ Define Sample Image Objects on page 226

4. Classify, review the results and optimize your classification.

→ Assess the Quality of Samples on page 230

8.3.5.1 About Nearest Neighbor Classification

The Nearest Neighbor classification procedure is another method to classify image objects using the class description.

Classification with membership functions are based on user-defined functions of object features. In contrast, Nearest Neighbor classification uses a set of samples for different classes in order to assign membership values. The procedure therefore consists of two major steps:

1. Teach the system by giving it certain image objects as samples.
2. Classify image objects in the image object domain due to their nearest sample neighbors.

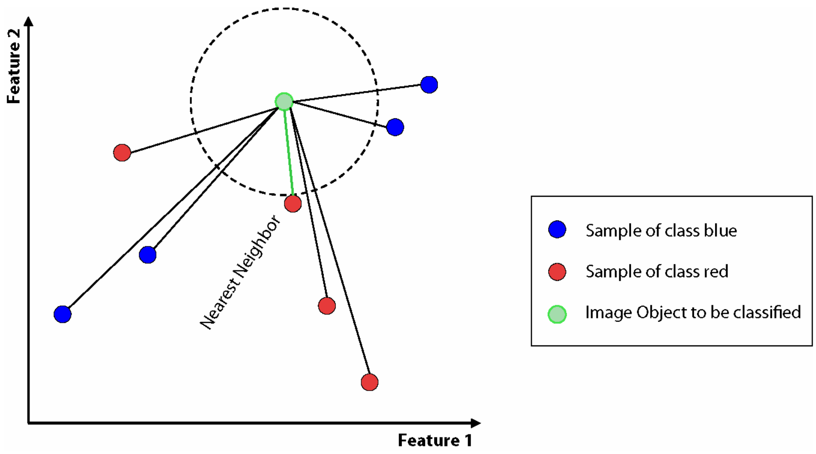


Figure 175: Principle of the Nearest Neighbor classification.

The Nearest Neighbor classifier returns a membership value between 0 and 1 based on the image object's feature space distance to its nearest neighbor. The membership value is 1 if the image object is identical to a sample. If the image object differs from the sample, the feature space distance has a fuzzy dependency on the feature space distance to the nearest sample of a class. You can select the features to be considered for the feature space.

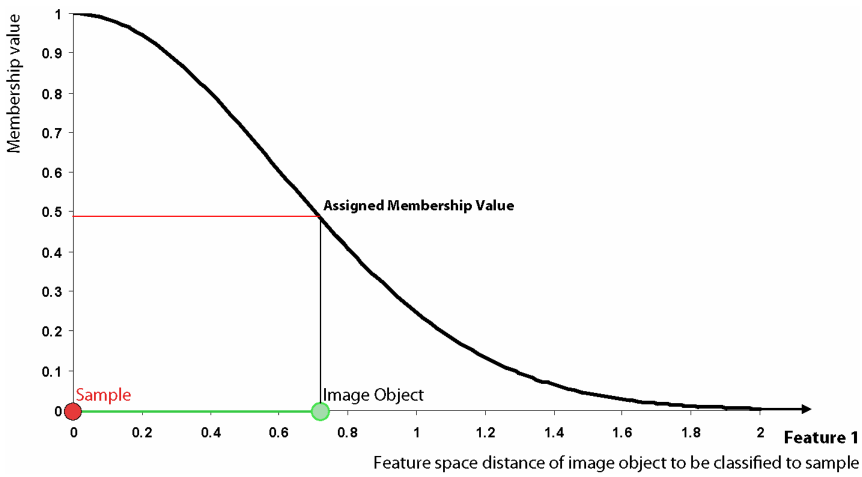


Figure 176: Membership function created by Nearest Neighbor classifier.

Considering that for an image object to be classified, only the nearest sample is used to evaluate its membership value. The effective membership function at each point in the feature space is a combination of fuzzy function over all samples of that class. See the membership functions in one dimension that means related to one feature:

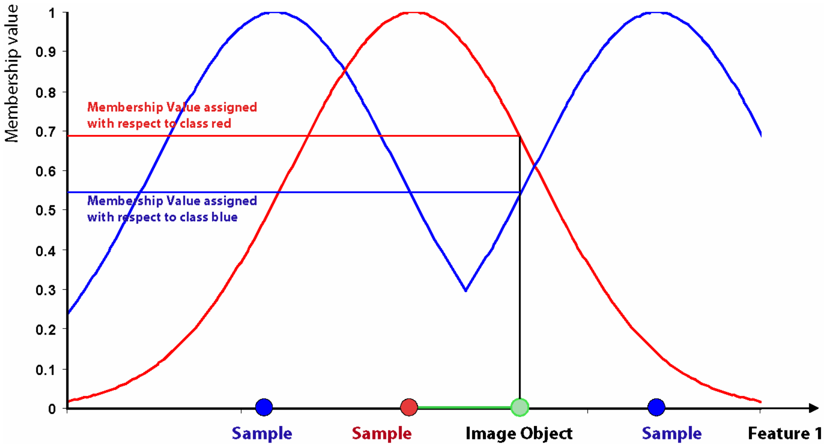


Figure 177: Membership function showing Class Assignment in one dimension.

In higher dimensions, depending on the number of features considered, it is harder to depict the membership functions. However, if you consider two features and two classes only, it might look like the following graph:

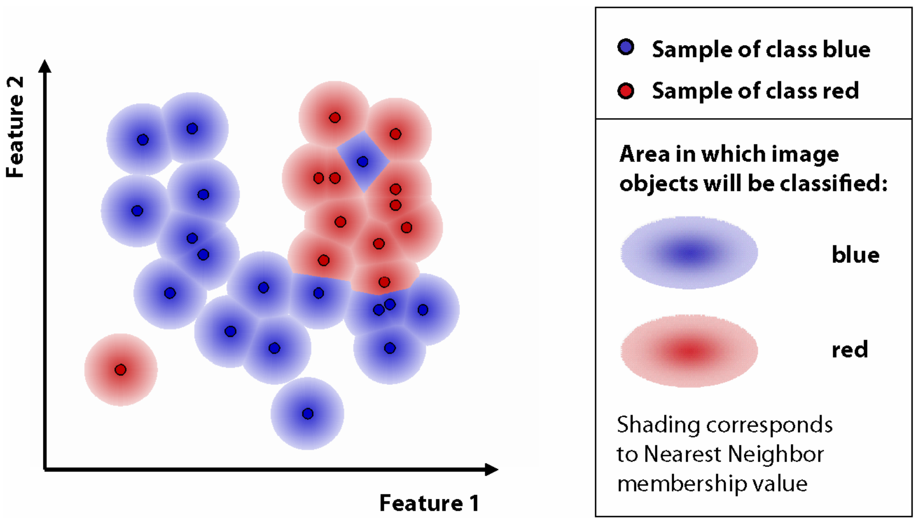


Figure 178: Membership function showing Class Assignment in two dimensions. Samples are represented by small circles. Membership values to red and blue classes correspond to shading in the respective color, whereby in areas in which object will be classified red, the blue membership value is ignored, and vice versa. Note that in areas where all membership values are below a defined threshold (0.1 by default), image objects get no classification; those areas are depicted white in the graph.

8.3.5.2 Define the Feature Space with Nearest Neighbor Expressions

To define feature spaces, **Nearest Neighbor (NN)** expressions are used and later applied to classes.

Definiens Developer distinguishes between two types of nearest neighbor expressions:

- **Standard Nearest Neighbor** (Standard NN): the feature space is valid for all classes it is assigned to within the project.

→ Apply the Standard Nearest Neighbor Classifier on page 222

- **Nearest Neighbor** (NN): the feature space can be defined separately for each class by editing the class description.

→ Edit the Class Description on page 246

1. From the main menu bar, choose **Classification** > **Nearest Neighbor** > **Edit Standard NN Feature Space**. The **Edit Standard Nearest Neighbor Feature Space** dialog box opens. By default, the some object features are contained in the **Selected** area 3 below **Layer Values** > **Mean**.

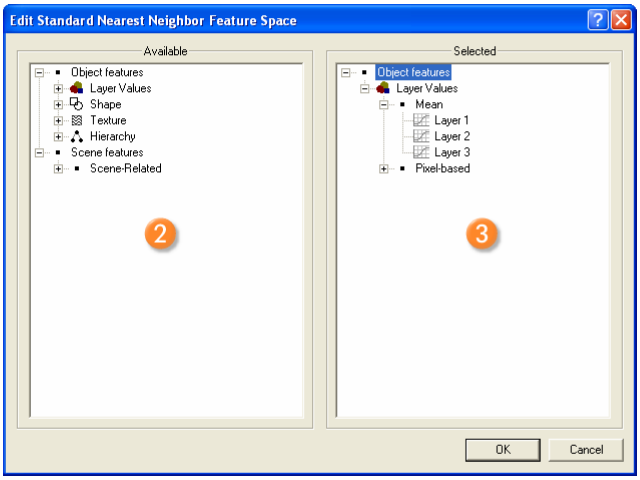


Figure 179: The Edit Standard Nearest Neighbor Feature Space dialog box.

2. To add a feature to the feature space, navigate to the feature in the tree in the **Available** area 2 on the left side and double-click it. The selected feature is displayed in the **Selected** area on the right.

   At this stage you cannot use class-related features because they are available only after an initial classification.

→ About Using Class-Related Features in a Nearest Neighbor Feature Space on page 241

3. To remove a feature from the feature space, double-click it in the **Selected** area 3.
4. Select features that are useful for distinguishing different classes in the image.

→ Get a Visual Overview of Feature Values on page 180

5. Use the feature space optimization to combine the best features.

→ Optimize the Feature Space on page 237

Apply the Standard Nearest Neighbor Classifier

After defining the Standard Nearest Neighbor, you apply this function to selected or all classes in your project.

1. From the main menu bar, select **Classification** > **Nearest Neighbor** > **Apply Standard NN to Classes**. The **Apply Standard NN to Classes** dialog box opens.

→ Select Classes on page 456

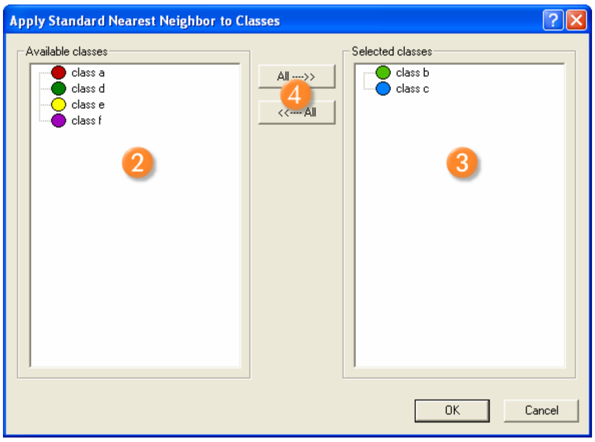


Figure 180: The Apply Standard Nearest Neighbor to Classes dialog box.

2. From the **Available classes** list (2) on the left, select the appropriate classes by clicking on them.

3. To remove a selected class, click it in the **Selected classes** list (3). The class is moved to the **Available classes** list.

4. Click the **All --->>** (4) button to transfer all classes from **Available classes** to **Selected classes**.
   To remove all classes from the **Selected classes** list, click the **<<--- All** button.

5. Click **OK** to confirm your selection.

6. In the **Class Hierarchy** window, double-click one class after the other to open the **Class Description** dialog box and to confirm that the class contains the **Standard Nearest Neighbor** expression.

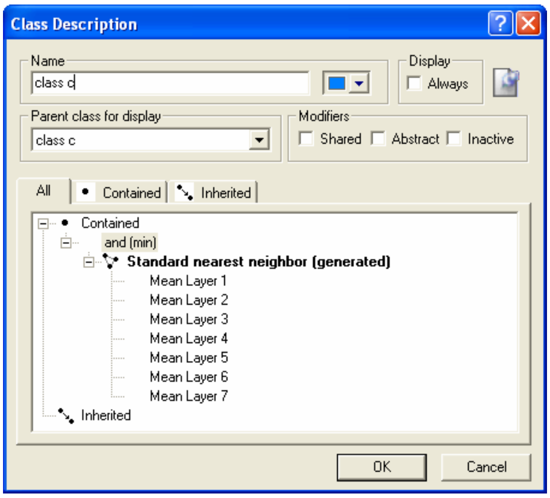


Figure 181: The Class Description dialog box.

Note

The Standard Nearest Neighbor feature space is now defined for the entire project. If you change the **Standard Nearest Neighbor** feature space in one class description, all classes that contain the **Standard Nearest Neighbor** expression are affected.

The feature space for both the **Nearest Neighbor** and the **Standard Nearest Neighbor** classifier can be edited by double-clicking them in the **Class Description** dialog box.

Now that you have assigned the **Nearest Neighbor** classifier to your classes, you need to collect samples representing each of the classes.

→ **Define Sample Image Objects** on page 226

8.3.5.3 Interactive Workflow for Nearest Neighbor Classification

Successful Nearest Neighbor classification usually requires several iterations of sample selection and classification.

It is most effective start with a small number of samples, then classify before again selecting more samples among the wrongly classified. The trick about the iteration is this: Within the feature space the misclassified image objects are located most likely near the border of the general area of this class. Those image objects are the ones you need most in order to accurately describe the feature space region covered by the class.

Consequently, you are more efficient when following an iterative workflow instead of selecting many samples from the beginning:

1. Insert Standard Nearest Neighbor into the class descriptions of classes to be considered.

→ **Apply the Standard Nearest Neighbor Classifier** on page 222

2. Select samples for each class; initially only one or two per class.

→ **Select Samples** on page 228

3. Run classification process.

→ **Algorithms for Process-Based Classification** on page 208

4. If image objects are misclassified, select more samples out of those and go back to step 2. You are finished if the classification is accurate.

Why Is Iterative Sample Selection More Efficient?

The Nearest Neighbor classifier classifies a good part of the scene correctly even when selecting only a small number of samples for each class. Image objects which are still classified incorrectly are likely to be at the border between the feature spaces of the classes:

For example: two classes, **red** and **blue**, are to be separated by nearest neighbor classification in two dimensions, this means by using two features. Some samples have been selected for each class. The distribution of the classes **red** and **blue** within the feature space might look like this:

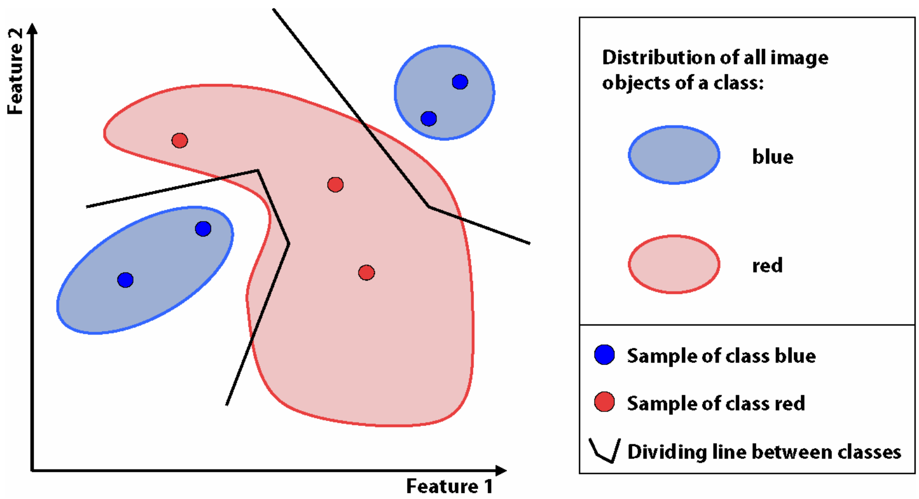


Figure 182: Feature space based on a few numbers of samples.

Nearest neighbor classification effectively separates the feature space into distinct parts, for example it will classify objects in the lower left and upper right in **blue**, image objects in-between in **red**. The dividing line between **blue** and **red** areas is well defined by chains of straight lines, each representing equidistance between their **red** and their **blue** nearest sample.

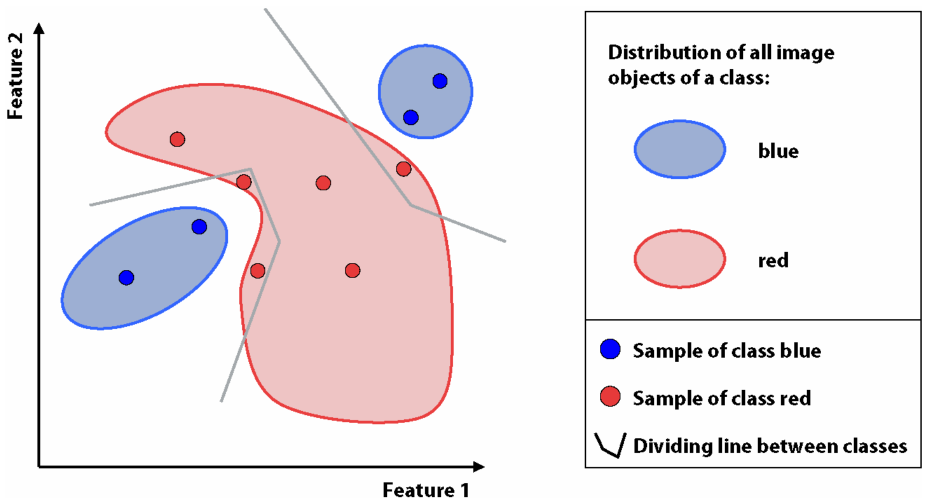


Figure 183: Samples are added for the border areas in the feature space..

When selecting additional samples in the area that was originally misclassified, the dividing lines are corrected. By adding samples along the border between the classes the borders are defined more accurately. Adding samples in the already well defined areas, does not add information.

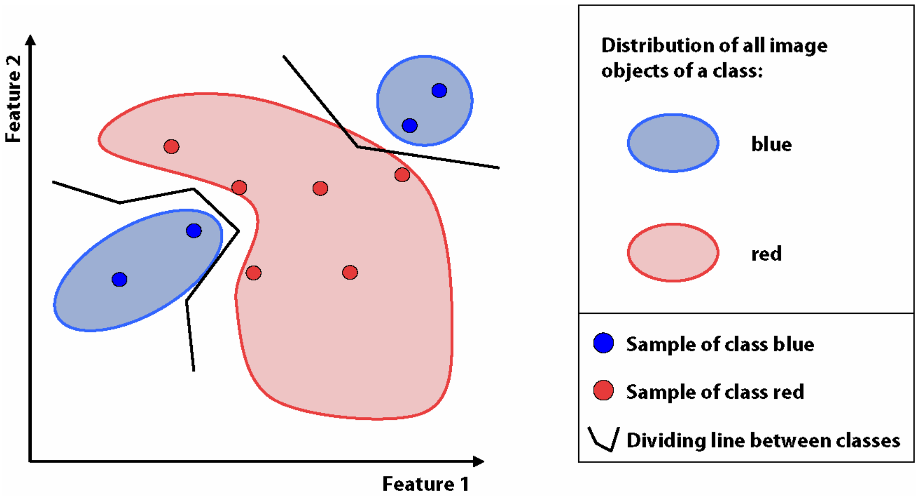


Figure 184: Feature space displaying the improved separation of classes.

8.3.5.4 Define Sample Image Objects

For the Nearest Neighbor classification, you need sample image objects. Sample image objects are image objects that you consider a significant representative of a certain class and feature.

By doing this, you train the Nearest Neighbor classification algorithm to differentiate between the classes. The more samples you select, the less errant the classification is.

You can define sample image objects manually by clicking the image object in the scene.

→ Work with the Sample Editor on page 226

You can load a TTA mask which contains previously manually selected sample image objects.

→ Load Samples from Training and Test Area (TTA) Mask on page 232

You can load a Shape file which contains information about image objects.

→ Create Samples Based on Shape File on page 235

Work with the Sample Editor

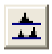
The **Sample Editor** window is the main tool used in the sample input process. It shows, for a selected class, the histograms of the feature values of its samples for a selection of chosen features. The same can be done for all image objects of a certain level or of all levels in the image object hierarchy.

The **Sample Editor** window is mainly used for comparing the attributes or histograms of image objects and samples for different classes. It is helpful to get an overview on the feature distribution of image objects or of samples of specific classes. The features of an image object can be compared to the total distribution of this feature over one or all image object levels. If you assign samples, features can also be compared to the samples of other classes.

Use this tool to assign samples using a Nearest Neighbor classification or to compare an object to already existing samples in order to determine to which class an object belongs. It is also very useful for obtaining a quick overview of the different feature values of an object.

1. To open the **Sample Editor** window, do one of the following:

- Select **View** > **Sample Editor** from the main menu bar.
- Choose **Classification** > **Samples** > **Sample Editor** from the main menu bar.


Sample Editor

2. By default, the **Sample Editor** window shows diagrams for only a selection of features. To select the features to be displayed in the **Sample Editor,** right-click in the **Sample Editor** window and select **Select Features to Display** from the context menu**.**

3. In the **Select Displayed Features** dialog box, double-click a feature from the left box to select it. To remove a feature, click it in the right box.

4. To add the features used for the Standard Nearest Neighbor expression, select **Display Standard Nearest Neighbor Features** from the context menu.

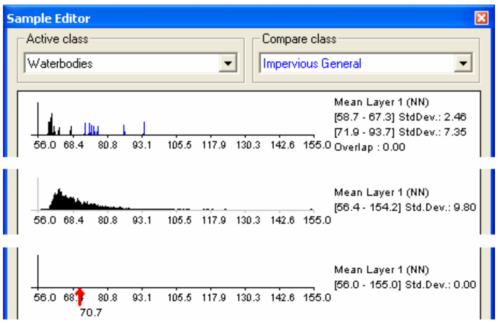


Figure 185: Sample Editor window.
Top: Active class and compare class histograms.
Middle: Histogram for all levels.
Bottom: Arrow indicating the feature value of a selected object.

Compare Features

1. To compare samples or layer histograms of two classes, select the classes or the levels that you want to compare in the **Active class** and **Compare class** lists.

2. Values of the active class are displayed in black in the diagram, the values of the compared class in blue. The value range and standard deviation of the samples are displayed to the right of the diagram.

View the Value of an Image Object

When you select an image object, the feature value is highlighted with a red pointer. This enables you to compare different objects with regard to their feature values.

The following functions help you to work with the **Sample Editor**:

- The feature range displayed for each feature is limited to the currently detected feature range. To display the whole feature range, select **Display Entire Feature Range** from the context menu.
- To hide the display of the axis labels, deselect **Display Axis Labels** from the context menu.
- To display the feature value of samples from inherited classes, select **Display Samples from Inherited Classes**.
- To navigate to a sample image object in the **Project View**, click on a slot in the **Sample Editor**.

In addition, the **Sample Editor** window allows generating membership functions. To do this, you have the following options:

- To insert a membership function to a class description, select **Display Membership Function** > **Compute** from the context menu.

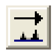
→ Generate Membership Functions Automatically on page 216

- To display membership functions graphs in the histogram of a class, select **Display Membership Functions** from the context menu.
- To insert a membership function or to edit an existing one for a feature, select the feature histogram and then select **Membership Function** > **Insert/Edit** from the context menu.
- To delete a membership function for a feature, select the feature histogram and select **Membership Function** > **Delete** from the context menu.
- To edit the parameters of a membership function, select the feature histogram and select **Membership Function** > **Parameters** from the context menu.

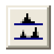
→ Edit Membership Function Parameters on page 217

Select Samples

A nearest neighbor classification needs training areas. Therefore, a representative selection of image objects, the **sample objects**, needs to be collected.


Select Samples

1. To assign sample objects, activate the input mode. Choose **Classification** > **Samples** > **Select Samples** from the main menu bar. The project view changes to the **View Samples** mode.


Sample Editor

2. To open the **Sample Editor** window, which helps to gather adequate sample objects do one of the following:
    - Choose **Classification** > **Samples** > **Sample Editor** from the main menu bar.
    - Choose **View** > **Sample Editor** from the main menu bar.

    The **Sample Editor** window opens.
3. To select a class for which you want to collect samples, do one of the following:
    - Select the class in the **Class Hierarchy** window.
    - Select the class from the **Active class** drop-down list in the **Sample Editor** window.

    This makes the selected class your active class so any samples you collect will be assigned to that class.
4. To define an object as sample for a selected class, double-click or Shift-click the image object in the project view. To undo the declaration of an object as sample, double-click or Shift-click the sample object again.
    As long as the sample input mode is activated, the view will always change back to the **Sample View** when an object is selected. The **Sample View** mode displays sample objects in the class color. This way the accidental input of samples can be avoided.
5. To view the feature values of the sample object, see the **Sample Editor** window. This enables you to compare different objects with regard to their feature values.
6. Click another potential sample object for the selected class.
    Analyze its membership value and its distance to the selected class and to all other classes within the feature space. You have to decide if the sample includes new information to describe the selected class (low membership value to selected class, low membership value to other classes), if it is really a sample of another class (low membership value to selected class, high membership value to other classes) or if it is a sample needed to distinguish the selected class from other classes (high

membership value to selected class, high membership value to other classes). In the first iteration of selecting samples, start with only a few samples for each class, covering the typical range of the class in the feature space. Otherwise, its heterogeneous character will not be fully considered.

7. Repeat the same for your remaining classes.

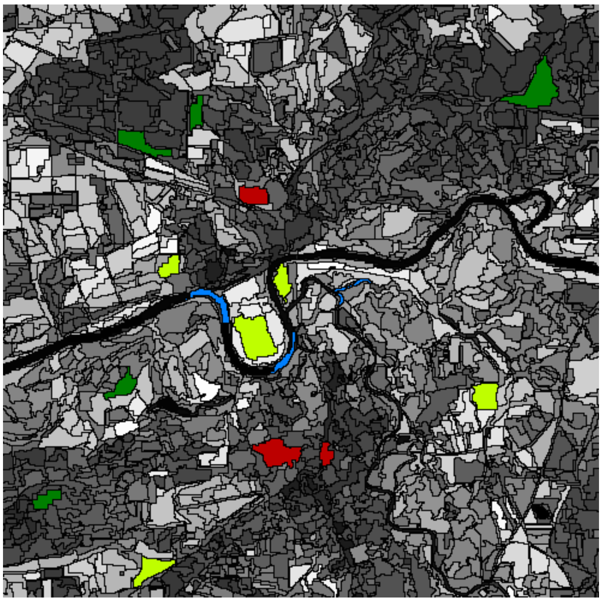


Figure 186: Project View with selected samples in View Samples mode. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

8. Classify the image.
9. The results of the classification are now displayed. In the **View Settings** dialog box, the mode has changed from **Samples** to **Classification**.
10. Notice that some objects may have been classified incorrectly or not at all. All objects that are classified are displayed in the appropriate class color. If you hover your mouse over a classified object, a tool tip pops up indicating the class to which the object belongs, its membership value, and whether or not it is a sample object. Objects that are unclassified appear transparent. If you hover your mouse over an unclassified object, a tool tip pops up indicating that no classification has been applied to this object. This information is also available in the **Classification** tab of the **Image Object Information** window.
11. The refinement of the classification result is an iterative process:
    - First, assess the quality of your selected samples.
    - Then, remove samples that do not represent the selected class well and add samples that are a better match or have previously been misclassified.
    - Classify the image again.
    - Repeat this step until you are satisfied with your classification result.

→ Algorithms for Process-Based Classification on page 208

Note

When you are finished collecting samples, remember to turn off the **Select Samples** input mode. As long as the sample input mode is turned on, the view mode will automatically switch back to the sample view mode, whenever an image object is selected. This is done to prevent the user from accidentally adding samples without noticing.

Select Samples with the Sample Brush

Use the **Sample Brush** to select samples.

The sample brush is an interactive tool that allows you to use your cursor like a brush that creates samples as you sweep it across the project view. You use it without clicking on image objects to use them as samples.

1. To use the **Sample Brush**, active it by doing one of the following:

   Select Samples with Brush

   - Click the **Select Sample** button in the **Sample** toolbar and right-click in the project view.
   - Click the **Sample Brush** button and click in the project view.
2. Drag the cursor across the image to select samples.
3. Select more precisely, simply zoom in. The brush diameter will not change.
4. By default, samples are not re-selected if the image objects are already classified but existing samples are replaced if drag over them again. These settings can be changed in the **Options** dialog box.

   → Options on page 452
5. To de-select samples, press the shift key on the keyboard while you drag.

> **Note**
>
> The **Sample Brush** will select up to one hundred image objects at a time, so you may need to increase magnification to use it if you have a large number of image objects.

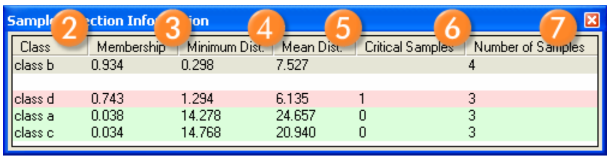
Assess the Quality of Samples

Once a class has at least one sample, the quality of a new sample can be assessed in the **Sample Selection Information** window. It can help you to decide if an object contains new information for a class, or if it should belong to another class.

1. To open the **Sample Selection Information** window do one of the following:

   Sample Selection Information

   - Choose **Classification** > **Samples** > **Sample Selection Information** from the main menu bar.
   - Choose **View** > **Sample Selection Information** from the main menu bar.



Class	Membership	Minimum Dist.	Mean Dist.	Critical Samples	Number of Samples
class b	0.934	0.298	7.527		4
class d	0.743	1.294	6.135	1	3
class a	0.038	14.278	24.657	0	3
class c	0.034	14.768	20.940	0	3

Figure 187: The Sample Selection Information window.

2. **Class** 2 displays the name of the class to which the values in the row belong.
3. **Membership** 3 shows the potential degree of membership according to the adjusted function slope of the nearest neighbor classifier. In other words, it indicates the membership value of the nearest neighbor classifier for the selected object.
4. **Minimum Dist.** 4 displays the distance in feature space to the closest sample of the respective class.

5. **Mean Dist.** 5 indicates the average distance to all samples of the corresponding class.

6. **Critical Samples** 6 displays the number of samples that are in a critical distance to the selected class in the feature space.

7. **Number of Samples** 7 indicates the number of samples selected for the corresponding class.

The following highlight colors are used for a better visual overview:

- **Gray:** Used for the selected class (first row).
- **Red:** Used if a selected sample is critically close to samples of other classes in the feature space.
- **Green:** Used for all other classes that are not in a critical relation to the selected class.

The critical sample membership value can be changed by right-clicking inside the window. Select **Modify Critical Sample Membership Overlap** from the context menu. The default value is **0.7**, which means all membership values higher than 0.7 are critical.

Figure 188: The Modify Threshold dialog box.

To select the classes to be displayed, right-click inside the dialog box and choose **Select Classes to Display**. Select the desired classes. For details on this dialog box, see **Select Classes**.

→ Select Classes on page 456

Navigate Samples

Use **Sample Navigation** to navigate to samples selected in the **Sample Editor**.

1. To activate **Sample Navigation**, do one of the following:

   - Choose **Classification** > **Samples** > **Sample Editor Options** > **Activate Sample Navigation** from the main menu bar.

     Sample Navigation

   - Right-click inside the **Sample Editor** and choose **Activate Sample Navigation** from the context menu.

   The **Sample Navigation** is activated.

2. Before navigating to samples you have to select a class in the **Select Sample Information** dialog box. Then switch to the **Sample Navigation** mode to investigate either the critical, nearest or current samples. To navigate use the drop-down list or the **Navigate to Sample** buttons to select your samples. A selected sample is highlighted in the **Project View** and as well as in the **Sample Editor**.

   Navigate to Next Sample

   Navigate to Previous Sample

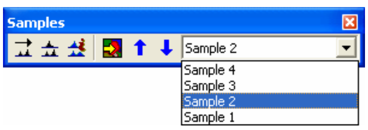


Figure 189: Sample Navigation: choose from a list of similar samples.

3. If in the **Sample Editor** there are two or more samples so close together that it is not possible to select them separately, switch between the samples using the blue arrows or the drop-down list.

Alternatively, the **2D Feature Space Plot** can be used to navigate through your selected samples.

Go to the **2D Feature Space Plot** dialog box and click on the desired sample to highlight it in the **Project View**. Samples are displayed as circles colored in the class color.

→ Analyze the Correlation of Two Features by Using the 2D Feature Space Plot on page 188

As with the **Sample Editor**, if two or more samples are so close together that they cannot be selected separately through the dialog box, use the blue arrows or the drop-down list to switch between the samples.

Delete Samples

Delete single sample objects, samples of specific classes, or all samples.

Delete Single Sample Objects

To delete a single sample object, simply double-click or Shift-click it.

Delete Samples of Classes

To delete samples of specific classes, choose one of the following from the main menu bar:

- **Classification** > **Class Hierarchy** > **Edit Classes** > **Delete Samples** which deletes all samples from the currently selected class.
- **Classification** > **Samples** > **Delete Samples of Classes** which opens the **Delete Samples of Selected Classes** dialog box.
  Move the desired classes from the **Available Classes** to the **Selected Classes** list or vice versa and click **OK** to confirm.

Delete All Samples

To delete all samples you have assigned, select **Classification** > **Samples** > **Delete All Samples**.

Alternatively you can delete samples by using the **delete all samples** algorithm or the **delete samples of class** algorithm.

→ Reference Handbook

Load Samples from Training and Test Area (TTA) Mask

Existing samples can be stored in a file called Training and Test Area (TTA) mask. That way, samples can be transferred to other datasets.

To allow mapping samples to image objects, you can define the degree of overlap that a sample image object must show to be considered within in the training area.

The TTA mask also contains information about classes for the project. You can use these classes or add them to your existing class hierarchy.

Create and Save a TTA Mask

You can create a Training and Test Area (TTA) mask from every project with sample image objects selected.

Precondition: Sample image objects must be selected.

→ Select Samples on page 228

1. From the main menu bar, select **Classification** > **Samples** > **Create TTA Mask from Samples**.

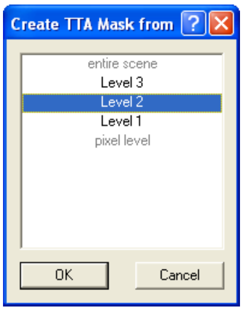


Figure 190: The Create TTA Mask from Samples dialog box.

2. In the opening **Create TTA Mask from Samples** dialog box, select the image object level that contains the samples that you want to use for the TTA mask. If your samples are all in one layer, this layer is selected automatically and cannot be changed.
3. Click **OK** to save your changes

   Your selection of sample image objects is now converted to a TTA mask.
4. To save the mask to a file, select **Classification** > **Samples** > **Save TTA Mask**.
5. In the **Save TTA Mask** dialog box, enter a file name and select the preferred file format.

Load and Apply TTA Mask

You can load and apply a previously saved Training and Test Area (TTA) mask.

1. From the main menu bar, select **Classification** > **Samples** > **Load TTA Mask**.
2. In the **Load TTA Mask** dialog box, select the desired TTA Mask file and click **Open**.
3. In the **Load Conversion Table** dialog box, open the corresponding conversion table file. The conversion table enables mapping of TTA mask classes to existing classes in your project. You can edit the conversion table.

→ Edit Conversion Table on page 234

4. Click **Yes** to create classes from the conversion table.
   If your project already contains classes, you can select if you want to replace them by the classes from the conversion file or add them. If you choose to replace them, your existing class hierarchy will be deleted.

   If you want to retain the class hierarchy, you can save it to a file.
5. Click **Yes** to replace the class hierarchy by the classes stored in the conversion table.

→ Load and Save Class Hierarchies on page 254

6. To convert the TTA Mask information into sample image objects, select **Classification > Samples > Create Samples from TTA Mask**. The **Apply TTA Mask to Level** dialog box opens.

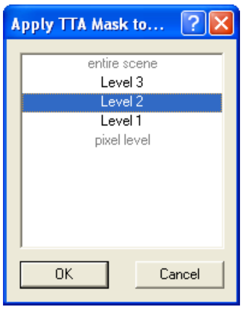


Figure 191: Apply TTA Mask to Level dialog box.

7. Select the level to which you want to apply the TTA mask information. If the project contains only one image object level, this level is preselected and cannot be changed.

8. In the **Create Samples** dialog box, enter the **Minimum Overlap for Sample Objects** and click **OK**.

   The default value is 0.75, i. e. 75 %. Since a single training area of the TTA mask does not necessarily have to match an image object, the minimum overlap decides whether an image object that is not 100 % within a training area in the TTA mask should be declared a sample object. The value 0.75 indicates that 75 % of an image object has to be covered by the sample area for a certain class given by the TTA mask in order for a sample object for this class to be generated.

   The **Image View** window displays the original project with sample image objects selected where the test area of the TTA mask have been.

Edit Conversion Table

The **Conversion Table** dialog box contains the linkage between the classes of the project and the classes of a Training and Test Area (TTA) mask.

You must edit the Conversion table only if you chose to keep your existing class hierarchy and used different names for the classes.

Precondition: A TTA mask has to be loaded. The project must contain classes.

1. To edit the conversion table, choose **Classification > Samples > Edit Conversion Table** from the main menu bar.

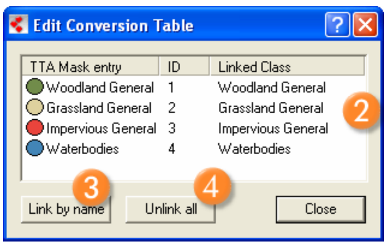


Figure 192: Edit Conversion Table dialog box.

2. The list 2 displays how classes of the project are linked to classes of the training and test area mask. To edit the linkage between the TTA mask classes and the classes of your project, right-click a TTA mask entry and select the appropriate class from the drop-down list.

3. Choose **Link by name** 3 to link all identical class names automatically.
4. Choose **Unlink all** 4 to remove the class links.

Create Samples Based on Shape File

You can use shapefiles to create sample image objects.

A shapefile, also called an ESRI shapefile, is a standardized vector file format used to visualize geographic data. You can obtain shapefiles either from other geo applications or by exporting them from **Definiens Developer** projects.

A shapefile consists of several individual files such as **.shx**, **.shp** and **.dbf**.

In short, using a shapefile for sample creation comprises of the following steps:

- Open a project and load the shapefile as a thematic layer.
- Segment the project using the thematic layer.
- Classify the image objects using the shapefile information.

Prerequisites: You must have a project with a corresponding shapefile.

1. Open the project.
2. Select **File > Modify Open Project** from the main menu. The **Modify Project** dialog box opens. **Insert** the shapefile as a new thematic layer.

→ Modify a Project on page 54

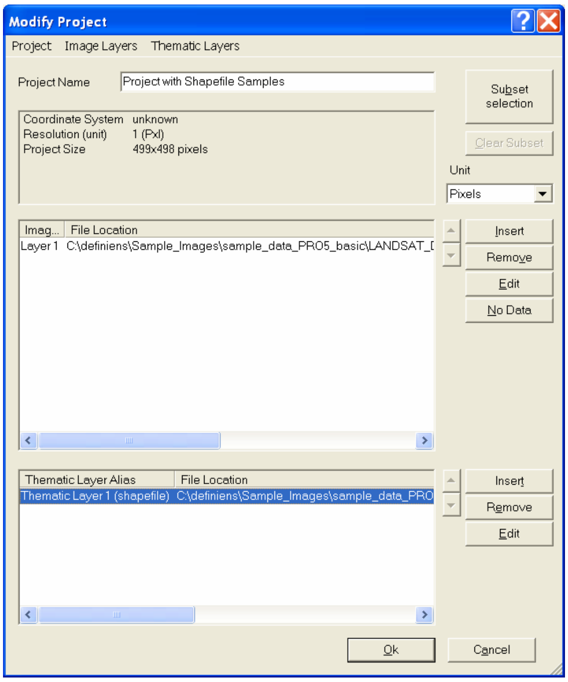


Figure 193: Insert a shapefile as a thematic layer.

3. For the segmentation and the following classification, create a new process.
4. In the **Process Tree** window, right-click and select **Insert Child** to create a new process:

→ Use Processes to Build Rule Sets on page 130

- Enter a process name.
- From the **Algorithm** list, select **execute child processes**.
- From the **Image Object Domain** list, select **no image object**.

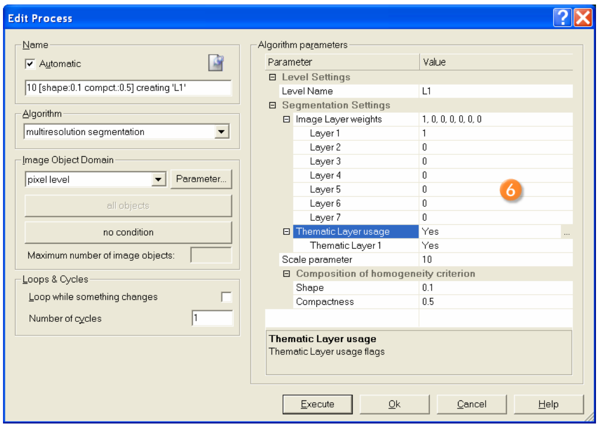


Figure 194: Edit process to use a shapefile.

5. In the **Process Tree** window, right-click and choose **Insert Child** on the context menu.
6. From the **Algorithm** drop-down list, select **multiresolution segmentation**. Under **Segmentation** settings, select **Yes** in the **Thematic Layer** entry 6. The segmentation finds all objects of the shape file and converts them to image objects in the thematic layer.
7. For the classification, create a new class.

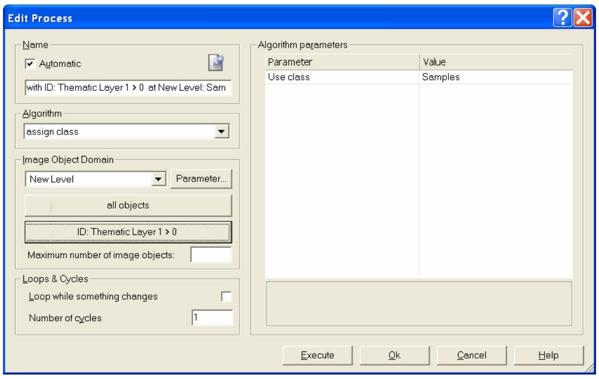


Figure 195: Process to import samples from shape file.

8. In the **Process Tree** window, add another child process. The child process identifies image objects by means of their ID and classifies them. Therefore, you use the threshold classifier **Image object ID**.
    - Apply the classification to **all objects**.
    - Select the following feature: **Object features** > **Thematic attributes** > **Thematic object attribute** > **[Thematic Layer 1]** > **ID**.
    - Set the threshold to > **0**.
    - For the parameter **Use class**, select the class for assignment.
9. To convert the classified image objects into sample image objects, add another child process. Use the **classified image objects to samples** algorithm. From the **Image Object Domain** list, select **New Level**. No further conditions are required.

10. Execute the process.

→ Execute Processes for Testing on page 151

8.3.5.5 Set Nearest Neighbor Function Slope

The Nearest Neighbor function slope defines the distance an object may have from the nearest sample in the feature space while still being classified. Enter values between 0 and 1. Higher values result in a larger number of classified objects

1. To set the function slope, choose **Classification** > **Nearest Neighbor** > **Set NN Function Slope** from the main menu bar.

Figure 196: The Set Nearest Neighbor Function Slope dialog box.

2. Enter a value and click **OK**.

8.3.5.6 Optimize the Feature Space

The Feature Space Optimization is an instrument to help you find the combination of features that is particularly suitable for separating classes in conjunction with a nearest neighbor classifier.

The Feature Space Optimization compares the samples for selected classes with respect to the features. As a result it finds the combination of features that produces the largest average minimum distance between the samples of the different classes. Given a maximum number of features f_{max} to be used for the combination it will systematically analyze all combinations with a dimension from 1 to f_{max}.

In order to compensate for the enlargement of the Euclidian distance between samples that automatically appears with each expansion of the feature space (even with highly correlated or identical descriptors) the Euclidian distance is normalized by $\sqrt{d}$, where **d** is the number of the features in the combination.

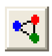
Use the Feature Space Optimization

The **Feature Space Optimization** dialog box helps you to optimize the feature space of a nearest neighbor expression.

1. To open the **Feature Space Optimization** dialog box, do one of the following:

    - Choose **Tools** > **Feature Space Optimization** from the main menu bar.
    - Choose **Classification** > **Nearest Neighbor** > **Feature Space Optimization** from the main menu bar.

    The **Feature Space Optimization** dialog box opens.



Feature Space Optimization

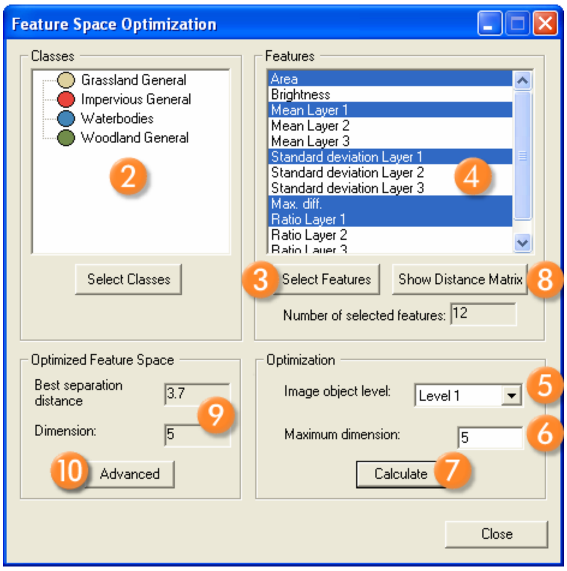


Figure 197: The Feature Space Optimization dialog box.

2. Click the **Select Classes** button 2 to select the classes for which you want to calculate the optimal feature space. Only classes for which you selected sample image objects, are available for selection.

3. Click the **Select Features** button 3 and select an initial set of features, which will later be reduced to the optimal number of features. You cannot use class-related features in the feature space optimization.

4. Highlight single features 4 to select a subset of the initial feature space.

5. Select the image object level for the optimization 5.

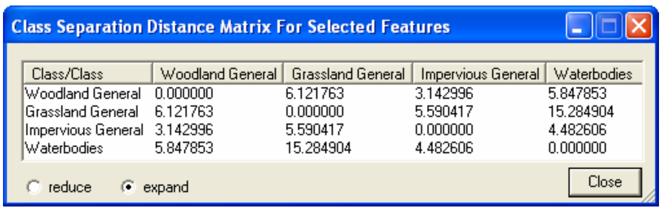
6. Enter the maximum number of features within each combination 6. A high number reduces the speed of calculation.

7. Click **Calculate** 7 to generate feature combinations and their distance matrices.

8. Click **Show Distance Matrix** 8 to display the Class Separation Distance Matrix for Selected Features dialog box. The matrix is only available after a calculation.

Note

When you change any setting of features or classes, you must first click **Calculate** before the matrix reflects these changes.



Class/Class	Woodland General	Grassland General	Impervious General	Waterbodies
Woodland General	0.000000	6.121763	3.142996	5.847853
Grassland General	6.121763	0.000000	5.590417	15.284904
Impervious General	3.142996	5.590417	0.000000	4.482606
Waterbodies	5.847853	15.284904	4.482606	0.000000

Figure 198: The Class Separation Distance Matrix for Selected Features dialog box.
The matrix displays the distances between samples of the selected classes within a selected feature space. Notice that the matrix is symmetric, which is due to the fact that each class is compared to each other.

9. After calculation, the **Optimized Feature Space** group box displays the following results:

- The **Best separation distance** 9 between the samples. This value is the **minimum** overall class combinations, because the overall separation is only as good as the separation of the closest pair of classes.
- The **Dimension** 9 indicates the number of features of the best feature combination.

10. Click **Advanced** 10 to open the **Feature Space Optimization - Advanced Information** dialog box and see more details about the results.

> **Note**
>
> The distance calculation is only based upon samples. Thus, adding or deleting samples also affects the separability of classes.

View Feature Space Optimization Advanced Information

The **Feature Space Optimization - Advanced Information** dialog box provides further information about all feature combinations and the separability of the class samples.

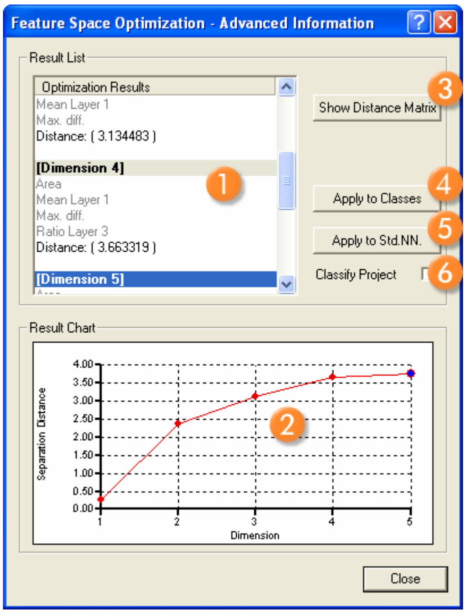


Figure 199: The Feature Space Optimization - Advanced Information dialog box.

1. The **Result List** 1 displays all feature combinations and their corresponding distance values for the closest samples of the classes.
   Select a feature space. The feature space with the highest result is highlighted by default.
2. The **Result Chart** 2 shows the calculated maximum distances of the closest samples along the dimensions of the feature spaces. The blue dot marks the currently selected feature space. Click another dot to show other feature combinations.

3. Click the **Show Distance Matrix** button 3 to display the **Class Separation Distance Matrix** window.
   This matrix shows the distances between samples of the selected classes within a selected feature space. Select a feature combination and re-calculate the corresponding distance matrix.

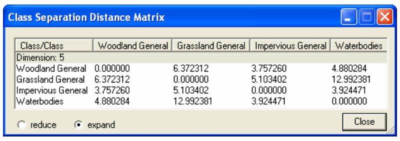


Figure 200: The Class Separation Distance Matrix dialog box.

Apply the Feature Space Optimization Results

You can automatically apply the results of your Feature Space Optimization efforts to the project.

4. In the **Feature Space Optimization Advanced Information** dialog box, click **Apply to Classes** 4 to generate a nearest neighbor classifier using the current feature space for selected classes.
5. Click **Apply to Std. NN.** 5 to use the currently selected feature space for the Standard Nearest Neighbor classifier.
6. Check the **Classify Project** 6 checkbox to automatically classify the project when choosing **Apply to Std. NN.** or **Apply to Classes**.

→ Define the Feature Space with Nearest Neighbor Expressions on page 221

Background Information on Feature Space Optimization

This section describes how the algorithm for calculating the separation distance works.

Calculation of the Separation Distance d_{ij} of Class i to Class j

Take each sample of class i and find this sample of class j with the smallest Euclidean distance to it. Take each sample of class j and find this sample of class i with the smallest Euclidean distance to it. Average these Euclidean distances over all (i + j) samples (i.e. $d_{ij} = d_{ji}$).

Calculation of the General Separation Distance (Between All Classes)

Take the minimum of all separation distances d_{ij} in the separating matrix with $i \neq j$.

Finding the Best Feature Combination

This algorithm computes the separation distances for all combinatorial combinations (constrained by f_{max}) and selects the biggest.

8.3.5.7 About Using Class-Related Features in a Nearest Neighbor Feature Space

To prevent non-deterministic classification results when using class-related features in a nearest neighbor feature space, several constraints have to be mentioned:

- It is not possible to use the feature **similarity to** of a class that is described by a nearest neighbor with class-related features.
- Classes cannot inherit from classes which use a nearest neighbor that contains class-related features. Only leaf classes in the inheritance class hierarchy can use class-related features in a nearest neighbor.
- It is impossible to use class-related features that refer to classes in the same group including the group class itself.

8.3.6 About Classes

Classes describe the semantic meaning of image objects in the cognition network. Classes can be put in a hierarchical structure so that class descriptions can be passed on from parent to child classes. You can also group classes to semantic groups.

Available classes are listed in the **Class Hierarchy** window. To open the **Class Hierarchy** window, do one of the following:

→ Organize Classes Within the Class Hierarchy on page 250

- Choose **Classification** > **Class Hierarchy** from the main menu bar.
- Choose **View** > **Class Hierarchy** from the main menu bar.


Class Hierarchy

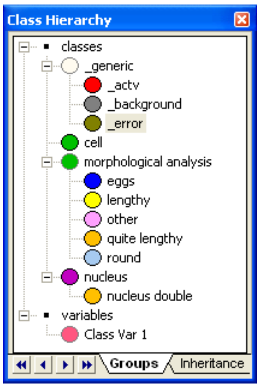


Figure 201: Sample Class Hierarchy window.

8.3.6.1 Basic Handling of Classes

This section explains the basic handling of classes in **Definiens Developer**: how to create, edit and delete them.

Beside some basic settings the class description is used for the definition of a class.

→ Edit the Class Description on page 246

Create and Edit a Class

When starting a new project, the class hierarchy is empty. You can insert new classes.

1. To insert a new class, do one of the following:

- Select **Classification > Class Hierarchy > Edit Classes > Insert Class** from the main menu bar.
- Right-click in the **Class Hierarchy** window and choose **Insert Class** from the context menu.

The **Class Description** dialog box opens.

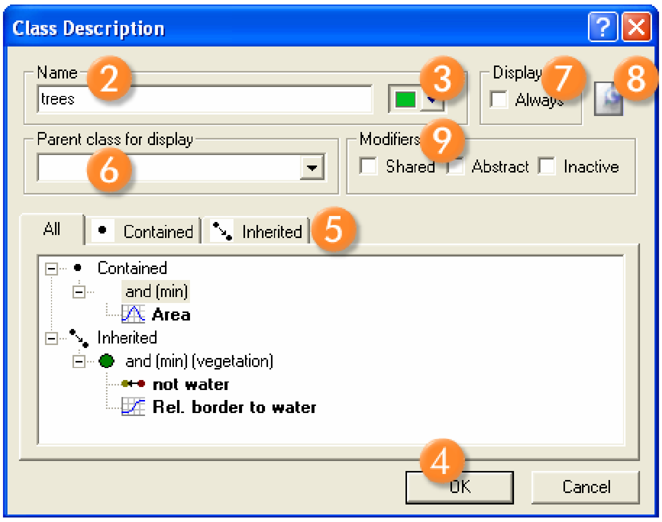


Figure 202: Class Description dialog box.

2. Enter a **Name** 2 for the class. This is set to **new class** by default.
3. Select a color 3 for the class. From this point forward, this color is used in diagrams or to mark classified image objects. You can change the color of the class at any time with no effect on the class description.
4. Click **OK** 4 to add the new class into the **Class Hierarchy** window.

> ## Note
>
> Naming and assigning a color to a class generates the basic class description but does not define the class. This is done by editing the class description.
>
> → Edit the Class Description on page 246

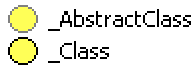
Options

5. By default, the class description field 5 displays all expressions, those inherited from the parent class as well as its own (contained) expressions. Change to the corresponding tab to view only the **Contained** or **Inherited** expressions.
6. Select the **Parent class for display** 6 to navigate up the semantic groups hierarchy. This function is necessary only when a class belongs to multiple parent classes in the group hierarchy.
7. Check **Display Always** 7 to enable the display of the class (for example, after exporting) even if it has not been used to classify objects.
8. Click the **Comment** button to open the **Comment** 8 dialog box and enter a comment.


Comment

9. Select modifiers 9 to change the scope of a class:

- **Shared**: Lock a class to prevent it from being changed. Shared image objects can be shared among several rule sets. Therefore, they must not be modified later. (Additionally a **Mark as Shared** command is available on the right-click context menu of a selected class in the **Class Hierarchy** window.)
- **Abstract**: Abstract classes do not apply themselves to image objects directly; they only inherit or pass on their class descriptions to child classes. In the **Class Hierarchy** window, an abstract class is assigned by a gray (instead of black) ring around the colored circular area.



- **Inactive**: Inactive classes are ignored in the classification process. In the **Class Hierarchy** window, an inactive class is noted in square brackets.

[InactiveClass]

10. To modify a class you can edit it as described above. Go to the **Class Hierarchy** window and double-click a class . The **Class Description** dialog box opens for editing. Additionally, class modification are available on the right-click context menu of a selected class in the **Class Hierarchy** window.

→ Manage Classes in the Class Hierarchy on page 250

Duplicate a Class

You can create a new class by duplicating an existing one.

1. In the **Class Hierarchy** window, select the class to be duplicated.
2. Do one of the following:
   - Select **Classification** > **Class Hierarchy** > **Edit Classes** > **Duplicate Class**.
   - Right-click and choose **Duplicate Class**.

   A new class is inserted in the class hierarchy. It is named **Copy of [...]** and has the same color as the original class. It also contains the same class description.
3. Modify the class as you wish.
4. Modify or extend the class description by renaming the class, choosing a different color or changing the class description.

Delete a Class

To delete a class, do one of the following:

- Select a class and press DEL.
- Choose **Classification** > **Class Hierarchy** > **Edit Classes** > **Delete Class** from the main menu bar.

In addition to deleting a class, you can do the following:

- Delete the classification of image objects

→ Delete Classification on page 442

- Delete the entire class hierarchy

→ Delete a Class Hierarchy on page 255

Document a Class

Insert comments in classes to use in rule set documentation.

→ Document a Rule Set on page 153

You can use the **Rule Set Documentation** window to view and store all the classes in your rule set including your comments.

1. To document a rule set item (such as a process, class or expression) select it in the window where the rule set item displays, which is one of the following:
    - **Process Tree** window
    - **Class Hierarchy** window
    - **Class Description** window.
2. Move your mouse over the lower right corner of the window to display the **Comment Window** button.

   Comment Window
3. Click the button to open the comment editing field and add a comment. The editing field is not available unless you have selected a rule set item. You can resize the window if necessary by dragging the borders.
   Your comment is automatically added to the rule set item as soon as you select another rule set item or another window.
4. You can also paste text into the editing area from the clipboard by using the context menu. Right-click at the insertion point and select **Paste** from the context menu.
5. To move up or down in the list of rule set items, click the up and down arrow buttons next to the editing field. This enables you to add comments without opening the rule set items for editing.
6. To remove a comment, use the **Undo** button or delete the comment from the comment editing field.

→ Undo Process Editing Operations on page 139

7. To view, store and copy your comments along with classes, processes and customized features, use the **Rule Set Documentation** window.

→ Document a Rule Set on page 153

Tip

*You can also open the **Class Description** window and use the **Comment** dialog box to add a comment. These comments will also display in the **Rule Set Documentation** window.*

→ *Create and Edit a Class on page 241*

8.3.6.2 About the Class Description

The class description contains class definitions like name and color and various settings. In addition it can hold expressions that describe the requirements an image object must meet to be a member of this class when class description based classification is used. There are the following families of expressions:

→ Edit the Class Description on page 246

- **Threshold expressions** define whether a feature is true or not, for example the value is 1 or 0. For instance the expression **Area < 500 pxl** classifies all image objects with an area of 499 pixels or fewer to be a member of the class. All other image objects are not in this class.

→ Use Thresholds for Classification with Class Descriptions on page 212

- **Membership functions** apply fuzzy logic to a class description. You can define the degree of membership, i.e. any value between 1 (true) and 0 (not true). The highest membership value defines the class to which the image object is assigned. If the membership value of an image object is lower than the pre-defined minimum membership value, the image object remains unclassified. If two or more class descriptions share the highest membership value, the assignment of an object to one of these classes is arbitrary.

  There are several pre-defined types of membership functions that you can adapt to meet your needs.

→ Use Membership Functions for Classification on page 213

- Use samples for **Nearest Neighbor** classification. For this method you declare image objects to be significant members of a certain class. The Nearest Neighbor algorithm then finds image objects that resemble the samples.

→ Use Samples for Nearest Neighbor Classification on page 219

- **Similarities** allow you to use class descriptions of other classes to define a class. Similarities are most often used as inverted expressions.

→ Use Similarities for Classification on page 218

You use **logical operators** to combine the expressions. These expressions can be nested to produce complex logical expressions.

About Operators

In the class description, you use operators to combine two or more expressions to a complex logical statement.

Expressions and logical operators can be combined to form well-structured class descriptions. An operator can combine one of the following:

- Expressions only
- Expressions and additional operators which again link expressions.

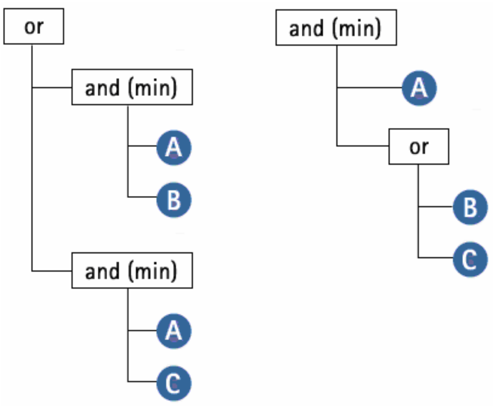


Figure 203: Example of using operators to describe classes.
Both constellations represent the same conditions to be met in order to classify an object.

Two common operators are **or (max)** and **and (min)** which are used in the following way:

- **or (max)**:
  The membership of the output equals the maximum fulfillment of the single statements. This means that out of a number of conditions combined by the maximum operator, the highest membership value is returned.

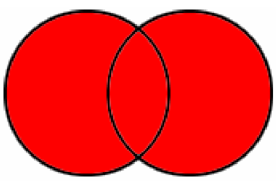


Figure 204: The or (max) operator combines conditions.

- **and (min)**:
  The membership of the output equals the minimum fulfillment of the single statements. This means that out of a number of conditions combined by the minimum operator, the lowest membership value is returned.

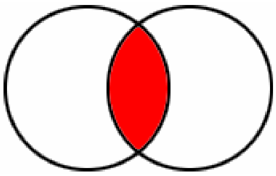


Figure 205: The and (min) operator uses the intersection of conditions.

For the minimum and maximum operators, only one statement determines the output, while for all other operators the values of all contained conditions contribute to the output.

When creating a new class, the minimum operator **and (min)** is used by default. Choose one of the following operators:

Operator	Description
or (max)	**or**-operator returning the maximum of the fuzzy values, the strongest **or**
mean (arithmetic)	arithmetic mean of the fuzzy values
and (min)	**and**-operator returning the minimum of the fuzzy values (used by default, the most reluctant **and**)
mean (geo.)	geometric mean of the fuzzy values
and (*)	**and**-operator returning the product of the fuzzy values
For given input values the membership degree of the condition and therefore of the output will decrease with the above sequence. Applicable to all of the above:	
not	inversion of a fuzzy value: returns 1 – fuzzy value.

→ Invert an Expression or Operator on page 248

8.3.6.3 Edit the Class Description

Edit the class description to handle the features describing a certain class and the logic by which these features are combined.

1. Open a class by double-clicking it in the **Class Hierarchy** window.
2. To edit the class description, open either the **All** or the **Contained** tab.
3. Insert or edit the expression to describe the requirements an image object must meet to be member of this class.

Insert an Expression

A new or an empty class description contains the **and (min)** operator by default.

1. To insert an expression, do one of the following:

- Right-click the operator in the **Class Description** dialog and choose **Insert new Expression**.
- Double-click the operator.

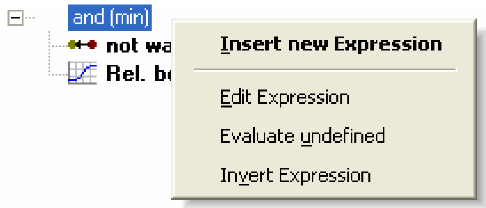


Figure 206: Context menu of the Class Description dialog box.

The **Insert Expression** dialog box opens. It shows all features plus extra possibilities like **Standard Nearest Neighbor** and other logical terms.

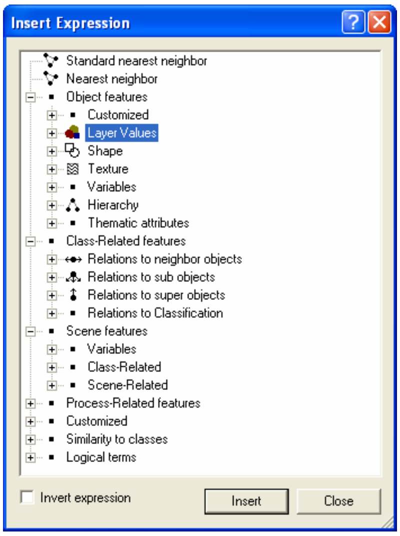


Figure 207: Insert Expression dialog box.

2. Select a feature and right-click it to open the context menu. Choose one of the following:

→ Select Single Feature on page 459

- **Insert Threshold**
  In the **Edit Threshold Condition** dialog box, set a condition for the selected feature, for example **Area <= 100**. Click **OK** to add the condition to the class description. Click **Close** on the **Insert Expression** dialog box.

→ Use Thresholds for Classification with Class Descriptions on page 212

- **Insert Membership Function**
  In the **Membership Function** dialog box, edit the settings for the selected feature.
  Click **OK** to add the function and close the **Insert Expression** dialog box.
  Alternatively, the **Membership Function** dialog box can be opened by double-clicking a feature.

→ Use Membership Functions for Classification on page 213

Note

While operators and similarities can be inserted into a class as they are, the nearest neighbor and the membership functions require further definition.

→ Define the Feature Space with Nearest Neighbor Expressions on page 221

Move an Expression

To move an expression, drag it to the desired location.

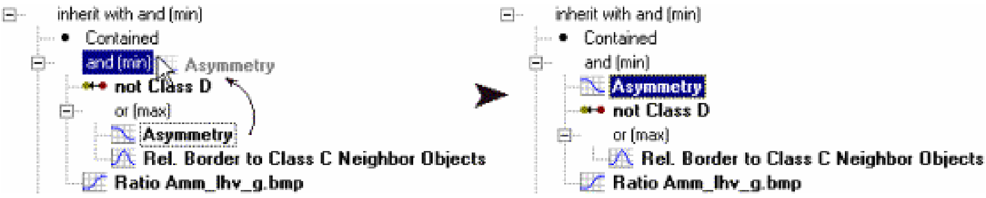


Figure 208: Moving expressions using drag-and-drop operations.

Invert an Expression or Operator

All expressions can be inverted into their opposite meaning. Mathematically speaking, an inverted expression has the value 1 minus the membership value.

Since logical operators are expressions that combine the membership values of a number of expressions, they yield a membership degree themselves and can consequently be inverted as well.

There are two possibilities to invert an expression or operator:

- Check the **Invert Expression** checkbox in the **Insert Expression** dialog box before inserting it into the class description.

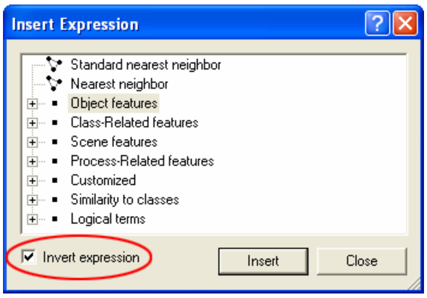


Figure 209: Invert an expression in the Insert Expression dialog box.

- If the expression or operator has already been inserted into the class description, right-click it and choose **Invert Expression** from the context menu.

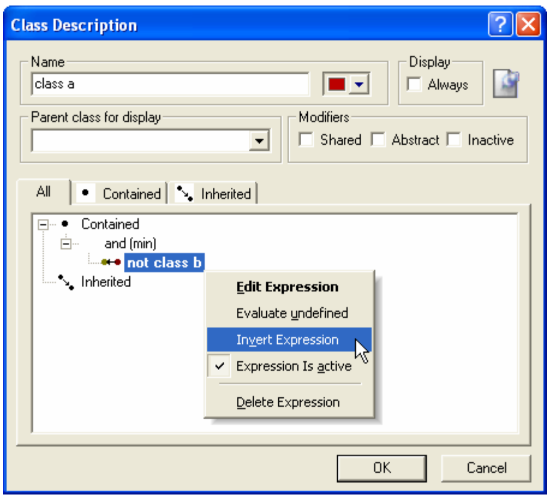


Figure 210: Invert an expression in the Class Description dialog box.

An inverted expression or operator appears in the class description with the prefix **not**. A common use is its application to similarities. This makes it possible to define that **Class A is not Class B**.

Edit an Expression

To edit an expression, double-click the expression or right-click it and choose **Edit Expression** from the context menu.

Depending on the type of expression, one of the following dialog boxes opens:

- **Edit Threshold Condition**
  Modify the threshold condition for a feature.
- **Membership Function**
  Modify the membership function for a feature.
- **Select Operator for Expression**
  Choose a logical operator from the list.
- **Edit Standard Nearest Neighbor Feature Space**
  Select or deselect features for the standard nearest neighbor feature space.
- **Edit Nearest Neighbor Feature Space**
  Select or deselect features for the nearest neighbor feature space.

Evaluate Undefined Image Objects

Image objects retain the status **undefined** when they do not meet the criteria of a feature. If you want to use these image objects anyway, e.g. for further processing, you must put them in a defined state. The function **Evaluate Undefined** assigns the value **0** for the specified feature.

1. In the **Class Description** dialog box right-click an operator.
2. From the context menu, select **Evaluate Undefined**. The expression below this operator is now marked with **(evaluate undefined)**.

Delete an Expression

To delete an expression, do one of the following:

- Select the expression and press DEL.
- Right-click the expression and choose **Delete Expression** from the context menu.

Document an Expression

Insert comments in an expression to use in rule set documentation.

→ Document a Rule Set on page 153

You can use the **Rule Set Documentation** window to view and store all the expressions in your rule set including your comments.

1. To document a rule set item (such as a process, class or expression) select it in the window where the rule set item displays, which is one of the following:

- **Process Tree** window
- **Class Hierarchy** window
- **Class Description** window.

2. Move your mouse over the lower right corner of the window to display the **Comment Window** button.

Comment Window

3. Click the button to open the comment editing field and add a comment. The editing field is not available unless you have selected a rule set item. You can resize the window if necessary by dragging the borders.
   Your comment is automatically added to the rule set item as soon as you select another rule set item or another window.
4. You can also paste text into the editing area from the clipboard by using the context menu. Right-click at the insertion point and select **Paste** from the context menu.
5. To move up or down in the list of rule set items, click the up and down arrow buttons next to the editing field. This enables you to add comments without opening the rule set items for editing.
6. To remove a comment, use the **Undo** button or delete the comment from the comment editing field.

→ Undo Process Editing Operations on page 139

7. To view, store and copy your comments along with classes, processes and customized features, use the **Rule Set Documentation** window.

→ Document a Rule Set on page 153

8.3.6.4 Organize Classes Within the Class Hierarchy

The class hierarchy is the framework to create the knowledge base for a given classification task. It contains all classes and is organized in a hierarchical structure.

→ About Classes on page 241

The class hierarchy distinguishes between passing down class descriptions (from parent classes to their child classes) in the inheritance hierarchy and the meaningful semantic grouping of classes in the groups hierarchy. The purpose is to reduce redundancy and complexity in the class descriptions, on the one hand, and to create a meaningful grouping of classes, on the other.

Manage Classes in the Class Hierarchy

The class hierarchy is the framework for the knowledge base in **Definiens Developer**. It contains all classes of a classification scheme.

It is organized in a hierarchical structure which distinguishes:

- The passing down of class descriptions from parent to child classes **(inheritance)**
- Meaningful semantic grouping **(groups)**

The class hierarchy is used to define class semantics and helps to reduce complexity when creating a knowledge base.

1. To open the **Class Hierarchy** window, do one of the following:
   - Choose **Classification** > **Class Hierarchy** from the main menu bar.
   - Choose **View** > **Class Hierarchy** from the main menu bar.

Class Hierarchy

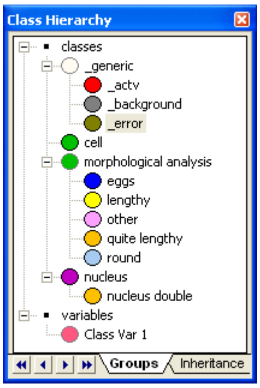


Figure 211: Sample Class Hierarchy window.

2. Select a class by clicking on it in the **Class Hierarchy** window.

→ Basic Handling of Classes on page 241

Each class can be found in both the **Groups** and the **Inheritance** tabs. This helps to give additional meaning to the classes.

→ About Inheritance on page 252
→ About Groups on page 252

Context Menu Options

Right click on a class in the **Class Hierarchy** window to perform one of the following:

Commands	Usage
Edit	Open the **Class Description** dialog box. Edit the currently selected class.
Insert Class	Open the **Edit Class Description** dialog box. Add a new class to the class hierarchy
Insert Class Variable	Open **Create Class Variable** dialog box. Add a new class variable to the hierarchy.
Duplicate	Create a copy of the currently selected class.
Find	Open the **Find and Replace** dialog box. Locate the currently selected class within the current rule set.
Delete Classification	Delete the currently selected class. Not available for class variables.
Select Color	Open the **Colors** dialog box. Change the color of the selected class.
Mark as Shared	Mark class as shared to use it in several rule sets.
Delete	Delete the currently selected class.
Delete Samples	Delete samples of the currently selected class. Not available for class variables.
Load Class Hierarchy	Open the **Load Class Hierarchy** dialog box. Load an existing class hierarchy.
Save Class Hierarchy	Open the **Save Class Hierarchy** dialog box. Save your class hierarchy as **.dkb** file.
Export Class Hierarchy	Open the **Export Class Hierarchy** dialog box. Export your class hierarchy as **.xml** file.
Delete Class Hierarchy	Delete the existing class hierarchy and processes.

→ Edit the Class Description on page 246
→ Create and Edit a Class on page 241
→ Create a Class Variable on page 259
→ Duplicate a Class on page 243
→ Find and Replace Rule Set Items on page 147
→ Delete Classification on page 442
→ Create and Edit a Class on page 241
→ Create and Edit a Class on page 241
→ Delete a Class on page 243
→ Delete Samples on page 232
→ Load and Save Class Hierarchies on page 254
→ Load and Save Class Hierarchies on page 254
→ Export a Class Hierarchy on page 255
→ Delete a Class Hierarchy on page 255

About Inheritance

The inheritance hierarchy of classes refers to the physical relations between classes. Here, the class descriptions defined in parent classes are passed down to their child classes. This aspect helps to differentiate a parent class, for example representing woodland in general, into two child classes representing woodland in rural or urban surroundings. It helps considerably to reduce the necessary number of input in the class descriptions.

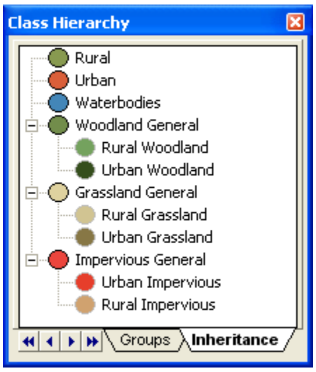


Figure 212: Sample Class Hierarchy using Inheritance.
This class hierarchy is an example of how to apply the inheritance hierarchy. The land cover classes **Woodland General**, **Grassland General** and **Impervious General** (parent classes) are further divided into land use classes (child classes). The two classes **Urban** and **Rural** do not contain any expressions. Their meaning is defined in the **Groups** hierarchy.

If you want to formulate the class description of a general class, you do this once in the relevant parent class in the inheritance hierarchy; this change is passed down to all child classes of the respective parent class. A child class has to fulfill the rules of the parent class as well as additional rules of its own.

Find Out More

Implicit Logic of the Inheritance in Class Descriptions

The inheritance of class descriptions contains an implicit logic concerning the application of inheritance and nearest neighbor classification:

Samples of child classes are simultaneously samples of their parent classes.

Because child classes are similar to their parent classes by inheritance, all parent classes must have the same samples as their child classes. This is especially important for classifications with class-related nearest neighbor. Otherwise, leaf classes in the inheritance class hierarchy, which are only described by class-related nearest neighbor, could not be applied. When you take a sample for a child class, this sample is automatically copied to all its parent classes.

*When using the **hierarchical classification** algorithm, additional logical rules are applied.*

→ ***Hierarchical Classification Algorithm*** *on page 210*

About Groups

The groups hierarchy refers to the semantic relation of classes. The child classes of parent classes in the groups hierarchy enable you to group classes, which can even contain very different feature descriptions, to an identical superior semantic meaning.

Functionally, this grouping has two consequences:

- Class-related features refer only to classes (and child classes) in the groups hierarchy. So, when using relationships to a parent class in the groups hierarchy, this also applies to the respective child classes.
- You can navigate the **Classification View** through the levels of the groups hierarchy by collapsing and expanding groups within the class hierarchy. This allows visualizing the semantic grouping of the classes.

→ Navigate Within the Image Object Hierarchy on page 114

Often the physical features of objects do not represent their semantic meaning. Therefore, the groups hierarchy allows you to group classes semantically. So, when classifying with class-related features, you can refer to groups of classes that belong to one semantic entity by referring to their parent class in the groups hierarchy.

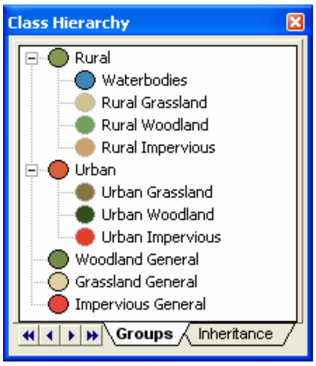


Figure 213: Sample Class Hierarchy using groups.
The example shows the same class hierarchy as in the **Inheritance** section. Here, the **Groups** hierarchy is displayed. The two classes **Urban** and **Rural** have an empty class description, i.e., they only act as a general semantic group. They do not pass on any information to their child classes but only merge all child classes semantically.

As in the given example, objects classified as woodland after an initial land cover classification might be divided into woodland areas in a rural and in an urban environment. The approach would be to subdivide this class into two child classes, **Urban** and **Rural Woodland** within the **Inheritance** hierarchy. But to be able to represent the semantic meaning at the same time, two classes (**Urban** and **Rural**) could be generated to function as groups for all **Urban** and **Rural** classes within the **Groups** hierarchy.

→ About Inheritance on page 252

Multiple Inheritance and Multiple Membership

A class can be the child class of more than one parent classes in both the inheritance and groups hierarchy. This leads to the inheritance of class descriptions from more than one parent class to a child class in the inheritance hierarchy.

In the groups hierarchy it means that the child class is a member of more than one semantic group.

To assign the same child class to a second parent class, use the right mouse button to drag and drop it to the new additional parent class. The same class will now appear twice, indicating the child class's membership to each respective parent class. However, double-clicking one of each of the symbols always opens the same class description.

> **Note**
>
> If a class is defined as a child class of more than one parent class in the groups hierarchy, you have to decide which of these parent classes to display when navigating upwards through the levels of the groups hierarchy in the classification view. To do so, select the correct parent class to be displayed under **Parent class for display** in the class description dialog of the specific child class.

To remove a class from a certain parent class, left-click the specific symbol and drag it below the list of classes.

Replace Classes

You can replace a class within the class hierarchy with another class. You can use this function to reduce duplicate classes to one, e.g. after load and append a class hierarchy. The redundant class will be replaced in all references to it.

1. In the **Class Hierarchy** window, right-click the class you want to replace.
2. Select **Find Class** from the context menu. The **Find and Replace** window opens.

→ Find and Replace Rule Set Items on page 147

Additionally you can search for classes if you have extended class hierarchies.

Editing the Hierarchical Structure

Within the class hierarchy, a hierarchical structure can easily be created by dragging classes to the desired location.

To define a class as a child class, click it and, while holding down the mouse button, drag it over the class to be defined as the parent class and drop it.

Figure 214: Defining a class as a child class.

To move a child class from one parent class to another, left-click it and drag it to the new parent class.

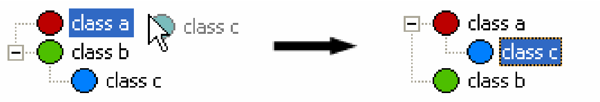


Figure 215: Moving a child class to another parent class.

To redefine a child class as a class without a parent class, left-click and drag it below the list of classes and drop it there.

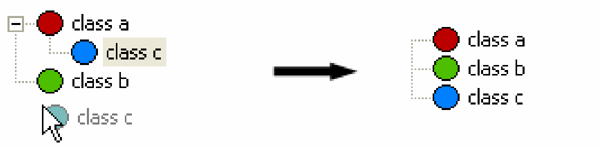


Figure 216: Defining a child class as a class without a parent class.

8.3.6.5 Load and Save Class Hierarchies

To be able to use different classification schemes on one data set or the same classification scheme on different data sets, you can save a class hierarchy to a **.dkb** file.

If you want to be able to use sample objects of a saved class hierarchy in a different project, you first have to save the sample objects in a TTA mask.

→ Create and Save a TTA Mask on page 233

1. To save a class hierarchy, do one of the following:

- Choose **Classification > Save Class Hierarchy** from the main menu bar.
- In the **Class Hierarchy** dialog box, select **Save Class Hierarchy** from the context menu.

2. To load a class hierarchy, do one of the following:
   - Choose **Classification > Load Class Hierarchy** from the main menu bar.
   - In the **Class Hierarchy** dialog box, select **Load Class Hierarchy** from the context menu.
3. Select if you want to replace any existing class hierarchy or add the new one to the existing hierarchy.

8.3.6.6 Export a Class Hierarchy

1. To export a class hierarchy to an **.xml** file, select **Classification > Class Hierarchy > Export Class Hierarchy** from the main menu bar.
2. In the **Save Class Hierarchy** dialog box, select a location and name for your file and click **Save**.

8.3.6.7 Delete a Class Hierarchy

Delete the entire class hierarchy including all classes.

To delete an entire class hierarchy, choose **Classification > Class Hierarchy > Delete Class Hierarchy** from the main menu bar. .

8.3.7 Review Results of a Classification

Use the various methods to evaluate the result of the classification and improve it if necessary.

Change to the classification view to get an overall impression of the classification results.

→ View Classification on page 109


View Classification


Image Object Information

Information about classification of image objects is provided in the **Image Object Information** windows. These contain information about features, classification, and class evaluation of image objects.

You can use this tool to obtain information about image objects that are classified incorrectly. Find out why the classification was incorrect and modify the respective class description. Another application is to verify the classification. In order to see how stable objects are in their classification, select the respective objects and compare the membership values of the first and second best classes. Low values or very similar values mean that the class descriptions are not very well suited to separate the classes.

The **Classification** tab of the **Image Object Information** window provides information about the current classification of an image object as well as alternative class assignments, sorted by descending membership values if fuzzy classification is used. Under **Current Classification**, the classification and membership values for the best as well as the second and third best class assignments resulting from the current classification are displayed. In contrast to the **Alternative Assignments** information, the **Current Classification** information is not calculated on demand when opening the dialog box. The values of the **Current Classification** will be kept until the next classification is performed or one of the class descriptions is changed.

The **Alternative Assignments** section shows a list of the evaluations of all classes in the current class hierarchy concerning the selected image object. The displayed membership values are calculated on demand when you select an object.

The **Class Evaluation** tab of the **Image Object Information** window provides detailed information about the classification of the selected image object. When you select an object, the values displayed are calculated on demand, using the current class descriptions of all applicable classes. This enables easy access to the evaluation of even complex class descriptions. At the same time, feature space borders distinguishing this class from another can be quantitatively ascertained.

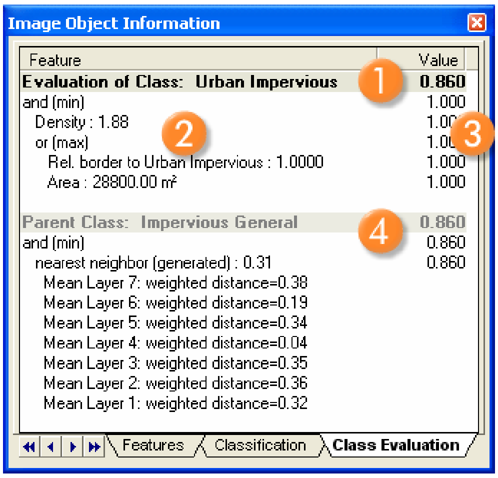


Figure 217: Class Evaluation tab of the Image Object Information window.

1. At the top of the **Class Evaluation** dialog box, the membership value of the selected class is displayed in bold letters 1.
2. The membership values for the expressions themselves as well as the membership values resulting from a combination of expressions by operators are displayed 2. Each expression of the class is listed beneath its operator. To the right of the expression, the object's feature value for this expression is given.
3. In the **Value** column 3, the resulting membership value for this expression is displayed.
4. If the selected class has one or more parent classes, the membership values for these classes are given as well 4.

No values in the **Value** column indicate that this feature is not applicable.

Note

The resulting membership values can differ from those given in the **Current Classification** section of the **Classification** tab of the **Image Object Information** window, as the calculation is performed on demand when you select the specific image object. Therefore, the **Alternative Assignments** section of the **Classification** tab and also the **Class Evaluation** tab show membership values, even if no classification has been performed so far.

8.4 Use Variables in Rule Sets

Within rule sets you can use variables in different ways. Prominent use cases of variables are:

- Constants
- Fixed and dynamic thresholds
- Receptors for measurements
- Counters
- Containers for storing temporary or final results
- Abstract place-holders that stand for a class, a features, or an image object level

While developing rule sets, you commonly use scene and object variables for storing your dedicated fine-tuning tools for reuse within similar projects.

→ How to Use Variables in Rule Set Development on page 262

Class variables, image object level variables, and feature variables allow you to write rule sets in a more abstract form. You can create rule sets that are independent of specific class names or image object level names, or feature types. This is helpful when developing parameters of actions as used in **Definiens Architect**.

8.4.1 About Variables

Types of Variables

You can create five types of variables. Two of them, scene and object variables, are place holders for floating point numbers or text-strings:

→ Floating Point Number on page 479

Scene variables are global variables that exist only once within a project. They are independent of the current image object.

Object variables are local variables. In contrast to scene variables they store values for each individual image object. Think of it as each image object having its own version of the variable.There is one instance per object in a project.

The remaining three types of variables are place-holders for rule set items in the software. Those rule set items can then be addressed in process algorithms by referring to the respective variable:

Class variables use classes as values. In a rule set they can be used instead of an ordinary classes where needed.

Level variables have other levels as their values. Level variables can be used in processes instead of ordinary levels.

Feature variables have features as their values.Once a feature is assigned to a feature variable, the feature variable can be used like that feature. It returns the same value as the feature to which it points. It uses the unit of whatever feature is assigned as a variable. It is possible to create a feature variable without a feature assigned, but the calculation value would be invalid.

Shared Variables

The concept of shared variables allows easy reuse of code sections with variables in several rulesets. Once a variable is declared as shared, a unique identifier is assigned to it. This identifier is not visible to the end-user. When two rule sets are merged, variables with the same identifier will be merged automatically.

To change the identifier, clear the **Shared** check box and select it agai. A new unique identifier will be created to prevent merges with older versions that might have of different contents.

Cutting and Merging Image Objects with Object Variables

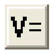
If you cut image objects with object variables, each fraction gets the same object variable value as the original image object.

If you merge two objects with object variables the merged image object gets the object variable value of the seed image object.

8.4.2 Create a Variable

Create a variable for usage within rule sets.

1. To open the **Manage Variables** dialog box, do one of the following:
    - On the **Tools** toolbar click the **Manage Variables** icon.
    - On the menu bar choose **Process > Manage Variables.**



Manage Variables

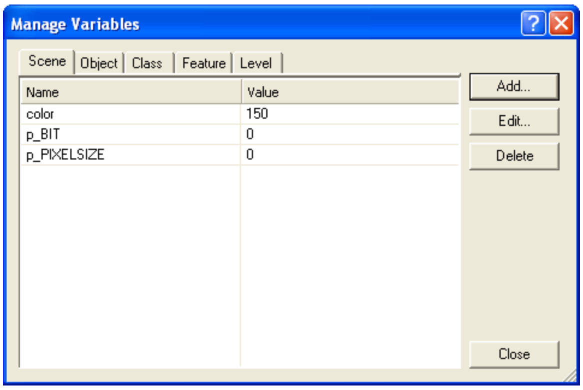


Figure 218: Manage Variables dialog box.

2. Select the tab for the type of variable you want to create. In the dialog box click **Add**. A **Create variable** dialog box opens. The fields are different for different types of variables.
3. Continue with creating a variable of the selected type.

8.4.2.1 Create a Scene or Object Variable

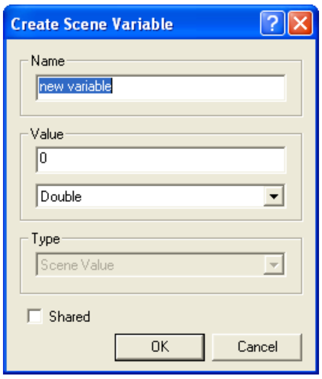


Figure 219: Create Scene Variable dialog box.

1. Insert a **Name** for the new variable.
2. Insert an initial **Value** for the new variable.
3. Choose whether the new variable is numeric (double) or textual (string).
4. The **Type** field is unavailable.
5. Select the **Shared** check box if you want to share the new variable among different processes.
6. Click **OK** to save the changes and return to the **Manage Variables** dialog box.
7. Confirm with **Close**. The new variable displays under **Object features** or **Scene features** in the feature tree of for example the **Feature View** window or the **Select Displayed Features** dialog box.

→ About Variables on page 257

→ Feature on page 24

8.4.2.2 Create a Class Variable

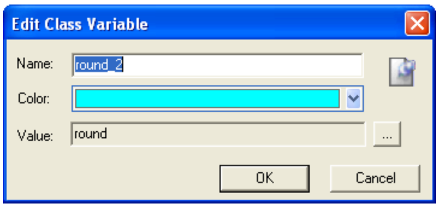


Figure 220: Create Class Variable dialog box

1. Insert a **Name** for the new variable.
2. Add a comment. Click the **Comment** button and add to the **Comment** dialog box to display when you open the class for editing.

Comment

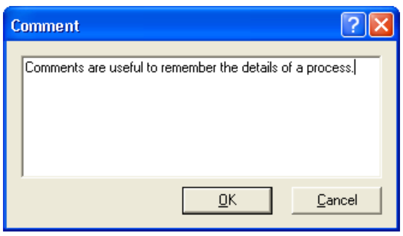


Figure 221: Comment dialog box

3. Select a **Color** for the new variable.

4. Select a **Value for the new variable.** Click the ellipsis button to select one of the existing classes as the value for the class variable.

(ellipsis button)

5. Click **OK** to save the changes and return to the **Manage Variables** dialog box.

   The new class variable displays here:

   - In the feature tree (used in several locations, for example, the **Feature View** window and the **Select Displayed Features** dialog box) under **Scene features > Class-Related**
   - In the **Class Hierarchy** window under **variables.**

→ Manage Classes in the Class Hierarchy on page 250

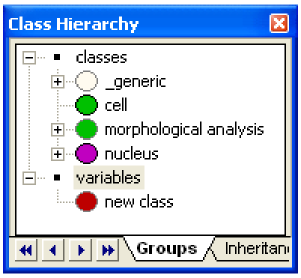


Figure 222: Class Hierarchy Window with Class Variable

8.4.2.3 Create a Level Variable

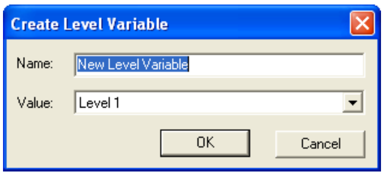


Figure 223: Create Level Variable dialog box

1. Insert a **Name** for the new level variable.
2. Use the drop-down arrow to select an existing level or leave the level variable unassigned.
3. Click **OK** to save the changes and return to the **Manage Variables** dialog box.
4. If the level variable is unassigned, you can use the drop-down arrow in the **Value** field of the **Manage Variables** dialog box to create one or more new names and type in a name for each. Click **Close** when you are finished.
5. The new variable displays in the **Edit Level Aliases** dialog box and in the **Select Level** window of the **Image Object Domain**.

→ Edit an Image Object Level or Level Variable on page 196

→ Navigate Within the Image Object Hierarchy on page 114

8.4.2.4 Create a Feature Variable

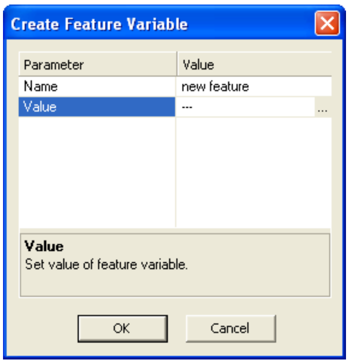


Figure 224: Create Feature Variable dialog box

1. Insert a **Name** for the new feature variable.
2. Click the ellipsis button in the **Value** field to open the **Select Single Feature** dialog box and select a feature as a value.

(ellipsis button)

3. Confirm with **OK**. The new variable displays under **Feature Variables** in the feature tree in several locations, for example, the **Feature View** window and the **Select Displayed Features** dialog box.

→ Feature on page 24

8.4.3 Manage Variables

The **Manage variables** dialog box also enables you to edit or delete variables.

→ Find and Replace Rule Set Items on page 147

1. For instructions on how to open the **Manage Variables** dialog box see **Create a Variable.**

→ Create a Variable on page 258

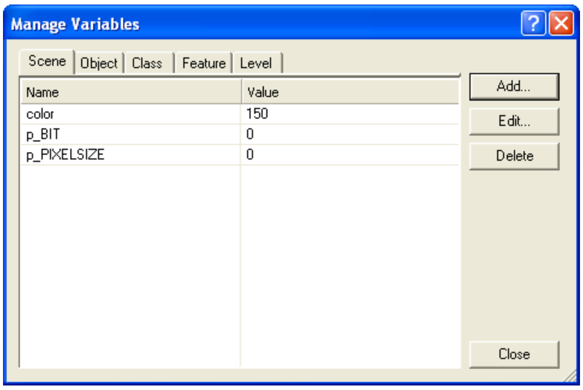


Figure 225: Manage Variables dialog box.

2. To edit a variable, select a tab (**Scene, Object, Class, Feature, Level**) for the type of variable you want to edit and then select the variable. Click **Edit** to open the **Edit** *Scene, Object, Class, Feature, Level* **Variable** dialog box, where you can modify the name and value.
3. To delete a variable, select it and then click **Delete**.
4. Click **Close** to close the dialog box.

8.4.4 How to Use Variables in Rule Set Development

During the development of a rule set, variables can be inserted instead of specific values in a number of situations. Several algorithms are specifically designed for the usage of variables.

Variables can be selected rather than typed in using several different dialog boxes.

Scene and object variables can be used instead of numbers in the following situations:

- Any parameters of processes in the Edit Process dialog box.

→ Specify Algorithm Parameters on page 134

- Threshold within the Edit Threshold Condition dialog box in the image object domain.

→ Use Thresholds for Classification with Class Descriptions on page 212

- Left and right border as well as maximum and minimum value within the definition of a membership function for a class.

→ Use Membership Functions for Classification on page 213

Scene variables, object variables and feature variables can be used as features. They display in the feature tree of for example the Feature View window or the Select Displayed Features dialog box.

→ Compare Feature Values by Using the Feature View on page 179

Class, level, and feature variables can be used wherever classes, levels, and features, respectively can be used.

Find Out More

Initialization of Variables

When a new variable is created, it is assigned an initial value as defined in the Value field of the Create Variable dialog box. In order to enable use of a value stored in a variable of a project in other projects within workspaces, variables generally are not initialized with default values. After processing a project, the last value of each scene variable is stored in the rule set and used as the initial value for the following project. The same is true when a rule set is saved as a file.

Note

In most cases you have to update the initial values of your variables. To do so, use the Update Variable algorithm. You will want to update each time you run the rule set, in addition to setting a value at the time of initialization. This is because the initial value is not automatically reset each time the rule set is executed.

8.4.4.1 Update Variable Algorithm

The update variable algorithm updates a variable.

1. To create a new process right-click on the **Process Tree** window and click **Append New Process**. The **Edit Process** dialog box opens.
2. Choose the **update variable** algorithm.

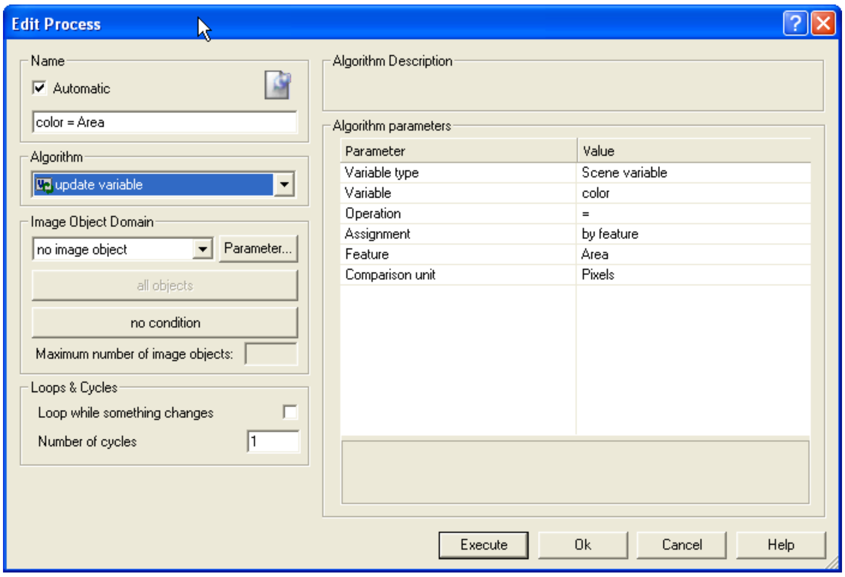


Figure 226: The Edit Process dialog box using the Update Variable algorithm.

3. In the **Algorithm parameters** group box, click the **Variable type** parameter and select **Object, Scene, Feature, Class** or **Level** from the drop-down list. Click the **Variable** parameter and select a variable from the drop-down list or enter a name to add a new one. If you have not previously created a Variable, or if you type a new name in the Variable field, the appropriate Create Variable dialog box will automatically open.

→ Create a Variable on page 258

4. Select the variable assignment, in the *Feature/Class/Level* field according to the variable type selected in the **Variable Type** field. This field does not display for **Object** and **Scene** variables.

To select a variable assignment, click in the field and do one of the following depending on the variable type:

- For feature variables, use the ellipsis button to open the **Select Single Feature** dialog box and select a feature or create a new feature variable.

 (ellipsis button)

→ Select Single Feature on page 459

- For class variables, use the drop-down arrow to select from existing classes or create a new class.

 (drop-down arrow button)

- For level variables, use the drop-down arrow to select from existing levels.

If you select **Scene** variable or **Object** variable, continue to steps 5 and 6 below.

5. Click the **Operation** parameter and select one of the following arithmetic operations from the drop-down list on the right side of the parameter:

Value	Operation
=	Assign a value.
+=	Increment by value.
−=	Decrement by value.
*=	Multiply by value.
/=	Divide by value.

6. Click the **Assignment** parameter and choose one of the two different means to assign a value to a variable:

- **By value** assigns a constant or another variable via the value parameter. To enter a new text value use quotes.
- **By feature** assigns the result of a feature evaluation. Click on the ellipsis button to open the **Select Single Features** dialog box .

If you choose to assign the variable **by feature**, and the selected feature is measured in different units, you can specify the unit you want to use for the comparison in the **Comparison unit** parameter. Click the drop-down arrow to display a list offering the available units.


(drop-down arrow button)

Tip

Select the Appropriate Image Object Domain

Ensure that no image object is assigned as image object domain for the following assignments:

- *Assign by value.*
- *Assign by scene-related features.*

Select the appropriate image object domain for features corresponding to the current image object, like following assignments:

- *Assign by object-related features.*
- *Assign by class-related features.*
- *Assign by process-related features.*

8.4.4.2 Compute Statistical Value Algorithm

The **compute statistical value** algorithm assigns a statistical value of a selected feature to a scene variable.

1. Open the **Edit Process** dialog box and select the **compute statistical value** algorithm.

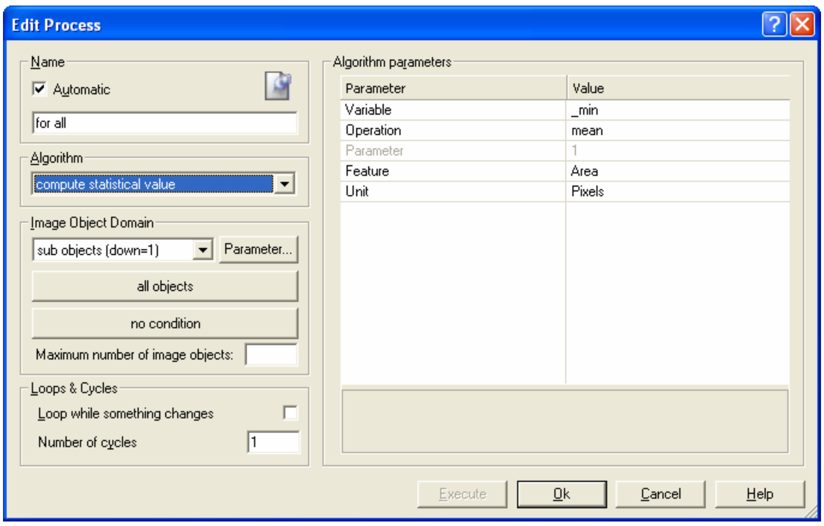


Figure 227: The Edit Process dialog box using the Compute Statistical Value algorithm.

2. In the **Algorithm parameters** group box click the **Variable** parameter and select an existing scene variable from the drop-down list or enter a name to add a new

one. If you have not previously created a **Variable**, the Create Variable dialog box will automatically pop-up.

3. Click the **Operation** parameter and select one of the following statistical operations from the drop-down list on the right side of the parameter:

→ Create a Variable on page 258

Number	counts the objects of the currently selected image object domain.
Sum	returns the sum of the feature values from all objects of the selected image object domain.
Maximum	returns the maximal feature value from all objects of the selected image object domain.
Minimum	returns the minimum feature value from all objects of the selected image object domain.
Mean	returns the mean feature value of all objects from the selected image object domain.
Std. deviation	returns the standard deviation of the feature value from all objects of the selected image object domain.
Median	returns the median feature value from all objects of the selected image object domain.
Quantile	returns the feature value, where a specified percentage of objects from the selected image object domain have a smaller feature value.

Options

Depending on the chosen operation you have the following options:

- If you have selected the **Quantile** operation specify the percentage threshold [0;100].
- If you have selected any **Operation** other than **Number**, you may click the ellipsis button to open the dialog box **Select Single Features**.
- If you have selected to assign the variable **by feature**, and the selected feature is measured in different units, then you can specify the unit you want to use for the comparison in the **Comparison unit** parameter. Click the drop-down arrow to display a list offering the available units.


(ellipsis button)

8.4.5 Example: Variables.dpr

Test your knowledge about variables with a small test project.

Try to solve the following sequence of tasks:

1. Create a new project with a provided image. The path is: **C:\Program Files\Definiens Developer** *Version number***\Examples \Variables\variables.tif**.
2. **Perform a chessboard** segmentation with object size 1.
3. Create scene variables **_lowerBoundaryForBlack = 110** and **_upperBoundaryForBlack = 200**.
4. Create a class named **red** with following class description: **Brightness < _lowerBoundaryForBlack**.
5. Create a class named **black** with following class description: **_lowerBoundaryForBlack < brightness < _upperBoundaryForBlack**.
6. Classify and merge **black** and **red** objects.

7. Create an object variable **enumeration = 0**.

8. Assign a different number to each black object using the variables **enumeration** and **_counter**. Increment **_counter** and assign it to each black image object. To do so, use a parent process and select **current image object** as image object domain in the **Edit Process** dialog box of the child process.

9. Check your result with the provided test project. Check the additional information in the comments of the processes. The project can be found at: **C:\Program Files\Definiens Developer** *Version numbe***r\Examples \Variables\variables.dpr**.

8.4.6 About Parameter Sets

Parameter sets are supersets of variables which allow you to save and reload the specific parameter configuration of a process tree independently of the rule set. This enables the adoption of an algorithm to a specific problem without changing the entire rule set.

Parameter set are especially important in rule sets using workspace automation and rule sets created for actions.

→ About Workspace Automation in Actions on page 308

When creating actions you should use special **Variables Operation Algorithms** to enable the automated exchange of parameter sets among actions.

→ Reference Book

A rule set can contain an arbitrary number of parameter sets which may overlap, in other words, a variable can be part of several parameter sets.

8.4.6.1 Create a Parameter Set

1. To create a parameter set, open the **Manage Parameter Sets** dialog box and choose **Process Commands > Manage Parameter Sets.**

Manage Parameter Set

→ Manage Parameter Sets on page 268

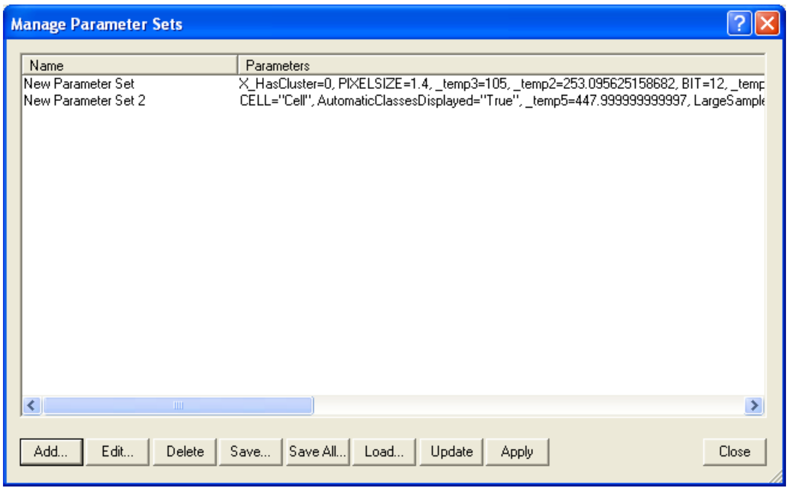


Figure 228: Manage Parameter Sets dialog box.

2. In the dialog box click **Add**. The **Select Variable for Parameter Set** dialog box opens. For instructions on how to add a variable see **Select Features.** After adding the variables the **Edit Parameter Set** dialog box opens with the selected variables displayed.

→ Select Features on page 457

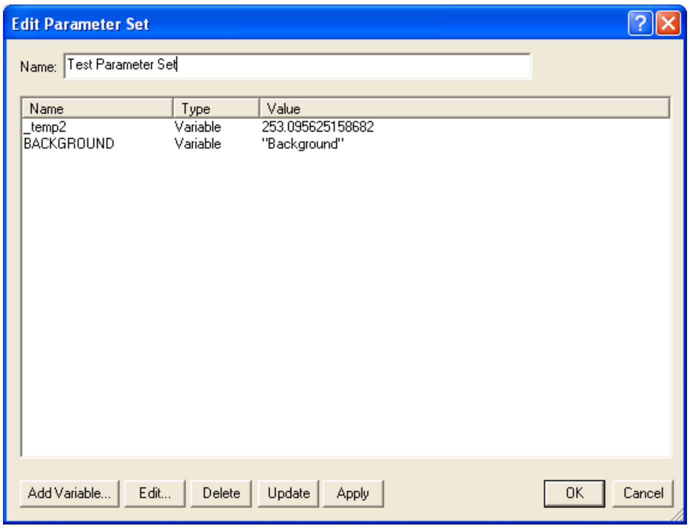


Figure 229: The Edit Parameter Set dialog box.

3. Insert a name for your new parameter set and confirm with **OK.**

8.4.6.2 Edit Parameter Set

To modify a parameter set you can edit it.

1. To edit a parameter set you need to open the **Manage Parameter Sets** dialog box.
2. Select a parameter set and click **Edit** to open the **Edit Parameter Set** dialog box.

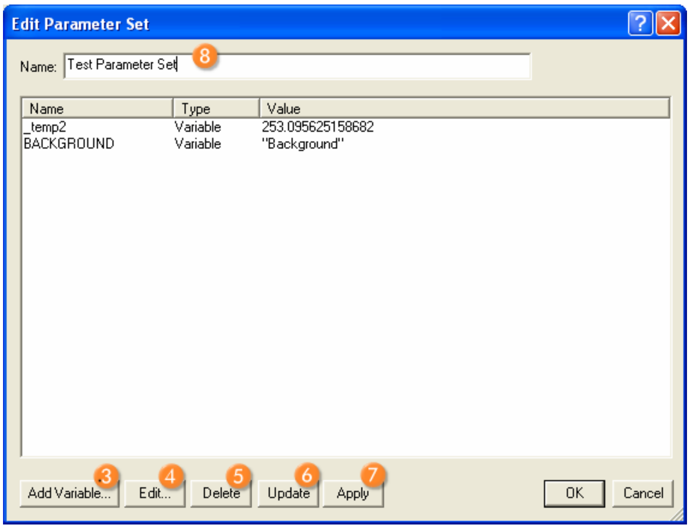


Figure 230: The Edit Parameter Set dialog box.

3. To add a variable to the parameter set click **Add Variable** 3. The **Select Variable for Parameter Set** dialog box opens for selecting features.

→ Select Features on page 457

4. To edit a variable select it and click **Edit** 4. The **Edit Value** dialog box opens where you can change the value of the variable.

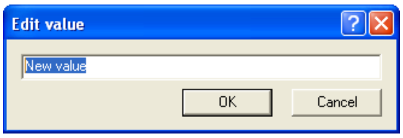


Figure 231: The Edit Value dialog box.

- If you select a feature variable, the **Select Single Feature** dialog opens,enabling you to select another value. Click **OK** to save your changes.
- If you select a class variable, the **Select Class** dialog opens,enabling you to select another value. Click **OK** to save your changes.
- If you select a level variable, the **Select Level** dialog opens,enabling you to select another value. Click **OK** to save your changes.

5. To delete a variable from the parameter set select it and click **Delete** 5.
6. Click **Update** 6 to modify the value of the selected variable according to the value of the rule set.
7. Click **Apply** 7 to modify the value of the variable in the rule set according to the value of the selected variable.

> **Note**
>
> Take care! The two preceding actions may change your rule set.

8. To change the name of the parameter set type in a new name 8.

8.4.6.3 Manage Parameter Sets

Managing parameter sets means not only to create and edit parameter sets but also to delete, load and save them. Additionally, you can update your parameter set according to the parameters of your rule set or apply the parameters of your parameter set to the rule set.

1. For instructions how to open the **Manage Parameter Sets** dialog box see **Create a Parameter Set**.

→ Create a Parameter Set on page 266

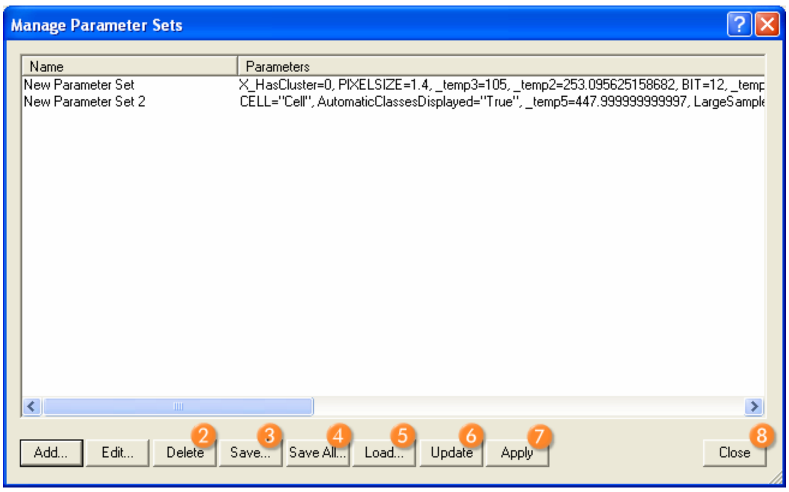


Figure 232: The Manage Parameter Sets dialog box.

2. To delete a parameter set you first need to select it and then click **Delete** 2.

3. To save a parameter set to a **.psf** file you first need to select it and then click **Save** 3.
4. Click **Save All** 4 when you want to save all parameter sets to one **.psf** file.
5. Click **Load** 5 to open existing parameter sets.
6. Click **Update** 6 to modify the values of the variables in the parameter set according to the values of the rule set.
7. Click **Apply** 7 to modify the values of the variables in the rule set according to the values of the parameter set.

> **Note**
>
> Take care! The two preceding actions may change your rule set.

8. Click **Close** 8 to close the dialog box.

8.5 Use Customized Features

Customized features allow you to create new features that are adapted to your needs. Customized features are composed of arithmetic and relational features. All customized features are based on the features shipped with **Definiens Developer** as well as newly created customized features.

- **Arithmetic features**, are composed of existing features, variables (**Definiens Developer** only), and constants, which are combined via arithmetic operations. Arithmetic features can be composed of multiple features but apply only to a single object.
- **Relational features**, are used to compare a particular feature of one object to those of related objects of a specific class within a specified distance. Related objects are surrounding objects (neighbors), sub-objects, superobjects, sub-objects of a superobject or a complete image object level. Relational features are composed of only a single feature but refer to a group of related objects.

8.5.1 Create Customized Features

The **Manage Customized Features** dialog box allows you to add, edit, copy, and delete customized features. It enables you to create new arithmetic as well as relational features based on the existing ones.

→ Find and Replace Rule Set Items on page 147

1. To open the **Manage Customized Features** dialog box, do one of the following:
   - On the menu bar click on **Tools** and then select **Manage Customized Features.**
   - On the **Tools** toolbar click on the **Manage Customized Features** icon.


Manage Customized Features

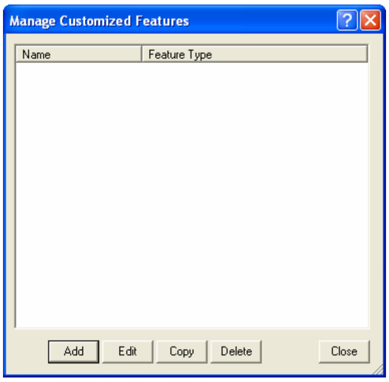


Figure 233: Manage Customized Features dialog box.

2. Click **Add** to create a new customized feature. The **Customized Features** dialog box will open, providing you with tools for the creation of arithmetic and relational features.
3. To edit a feature first you need to select it and then click **Edit** . This opens the **Customized Features** dialog in which you can modify the feature.
4. To copy or delete a feature you first need to select it and then depending on the action you want to perform you click either **Copy** or **Delete.**

Find Out More

Where Else to Find Customized Features

Newly created features can also be found under ***Customized*** *in the* ***Feature View****. To edit a customized feature, right-click the respective feature and select* ***Edit Feature****. To delete the feature, select* ***Delete Feature****.*

New customized features can be named and saved separately. Choose ***Tool > Save Customized Features*** *and* ***Tools > Load Customized Features*** *to save and reuse customized features. The file extension is* ***.duf****.*

8.5.2 Arithmetic Customized Features

The procedure below guides you through the steps you need to follow when you want to create an arithmetic customized feature.

1. Open the **Manage Customized Features** dialog box and click **Add**. The **Customized Features** dialog opens, make sure you currently viewing the **Arithmetic** tab.

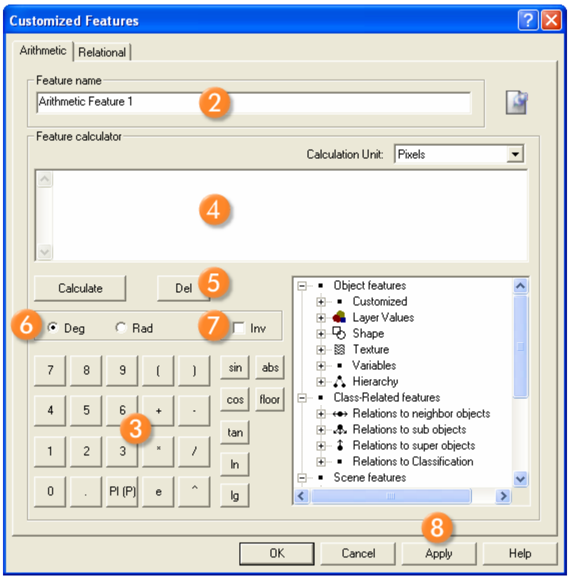


Figure 234: Creating an arithmetic feature in the Customized Features dialog box.

2. Insert a name 2 for the customized feature to be created.

3. Use the calculator 3 to create the arithmetic expression. You can:

   - Type in new constants.
   - Select features or variables (**Definiens Developer** only) in the feature tree on the right.
   - Choose arithmetic operations or mathematical functions.

Find Out More

About Calculating Customized Features

The calculator provides the following arithmetic operations and mathematical functions:

+ *addition*

– *subtraction*

***** *multiplication*

/ *division*

^ *power of (e.g. x^2 means x^2). You can use x^0.5 for the square root of x.*

sin *trigonometric function sine*

cos *cosine*

tan *tangent*

ln *natural logarithm to base e*

lg *logarithm to base 10*

abs *for absolute value*

floor *to round down to the next lowest integer (whole value). You can use floor(0.5+x) to round to the next integer value.*

4. The expression you create is displayed 4 at the text area above the calculator.

5. To calculate or delete 5 an arithmetic expression first you need to highlight the expression with the cursor and then click either **Calculate** or **Del** depending on the action you want to take.
6. You can switch between degrees (**Deg**) or radians (**Rad**) 6 measurements.
7. You can invert 7 the expression.
8. To create the new customized feature do one of the following:
   - Click **Apply** 8 to create the feature without leaving the dialog box or
   - Click **OK** to create the feature and close the dialog box.
9. After creation, the new arithmetic feature can be found in either one of the following locations:
   - In the **Image Object Information** window
   - In the **Feature View** window under **Object features>Customized.**

Note

Avoid invalid operations such as division by 0. Invalid operations will result in undefined values.

8.5.3 Relational Customized Features

The following procedure will assist you with the creation of a relational customized feature.

1. Open the **Manage Customized Features** dialog box and click **Add**. The **Customized Features** dialog opens, make sure you currently viewing the **Relational** tab.
2. Insert a name 2 for the relational feature to be created.
3. Select the relation 3 existing between the image objects.
4. Choose the relational function 4 to be applied.
5. Define the distance 5 of the related image objects. Depending on the related image objects, he distance can be either horizontal (units, e.g. pixels) or vertical (image object levels)
6. Select the feature 6 for which to compute the relation.

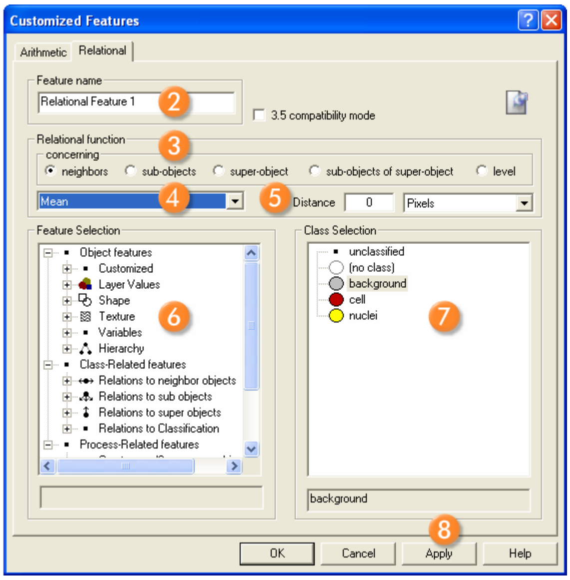


Figure 235: Creating a relational feature at the Customized Features dialog box.

7. Select a class, a group or **no class** 7 to apply the relation.
8. To create the new customized feature do one of the following:
   - Click **Apply** 8 to create the feature without leaving the dialog box or
   - Click **OK** to create the feature and close the dialog box.
9. After creation, the new relational feature can be found in the **Feature View** window under **Class-Related features > Customized.**

Note

As with class-related features, the relations refer to the groups hierarchy. This means if a relation refers to one class, it automatically refers to all subclasses of this class in the groups hierarchy.

Relations between surrounding objects can exist either on the same level or on a level lower or higher in the image object hierarchy:

neighbors	Related image objects on the same level. If the distance of the image objects is set to 0 then only the direct neighbors are considered. When the distance is greater than 0 then the relation of the objects is computed using their centers of gravity. Only those neighbors whose center of gravity is closer than the distance specified from the starting image object are considered. The distance is calculated either in metric units or pixels. For example, a direct neighbor might be ignored if its center of gravity is further away from the specified distance.
subobjects	Image objects that exist under other image objects (superobjects) whose position in the hierarchy is higher. The distance is calculated in levels.
superobject	Contains other image objects (subobjects) on lower levels in the hierarchy. The distance is calculated in levels.

sub-objects of superobject	Only the image objects that exist under a specific super-object are considered in this case. The distance is calculated in levels.
level	Specifies the level on which an image object will be compared to all other image objects existing at this level. The distance is calculated in levels.

The following table gives an overview of all functions existing in the drop-down list under the **Relational function** section:

Mean	Calculates the mean value of selected features of an image object and its neighbors. You can select a class to apply this feature or no class if you want to apply it to all image objects.Note that for averaging, the feature values are weighted with the area of the image objects.
Standard deviation	Calculates the standard deviation of selected features of an image object and its neighbors. You can select a class to apply this feature or no class if you want to apply it to all image objects.
Mean difference	Calculates the mean difference between the feature value of an image object and its neighbors of a selected class. Note that for averaging, the feature values are weighted by the area of the respective objects.
Mean absolute difference	Calculates the mean absolute difference between the feature value of an object and the feature values of its neighbors of a selected class. Note that for averaging, the absolute difference to each neighbor is weighted by the respective area.
Ratio	Calculates the proportion between the feature value of an image object and the mean feature value of its neighbors of a selected class. Note that for averaging the feature values are weighted with the area of the corresponding image objects.
Sum	Calculates the sum of the feature values of the neighbors of a selected class.
Number	Calculates the number of neighbors of a selected class. The feature you have selected is of no account. But it has to be selected for working of the feature.
Min	Returns the minimum value of the feature values of an image object and its neighbors of a selected class.
Max	Returns the minimum value of the feature values of an image object and its neighbors of a selected class.
Mean difference to higher values	Calculates the mean difference between the feature value of an image object and the feature values of its neighbors of a selected class, which have higher values than the image object itself. Note that for averaging the feature values are weighted by the area of respective image objects.
Mean difference to lower values	Calculates the mean difference between the feature value of an image object and the feature values of its neighbors of a selected class, which have lower values than the object itself. Note that for averaging the feature values are weighted by the area of the respective image objects.

Portion of higher value area	Calculates the portion of the area of the neighbors of a selected class, which have higher values for the specified feature than the object itself to the area of all neighbors of the selected class.
Portion of lower value area	Calculates the portion of the area of the neighbors of a selected class, which have lower values for the specified feature than the object itself to the area of all neighbors of the selected class.
Portion of higher values	Calculates the feature value difference between an image object and its neighbors of a selected class with higher feature values than the object itself divided by the difference of the image object and all its neighbors of the selected class. Note that the features are weighted with the area of the corresponding image objects.
Portion of lower values	Calculates the feature value difference between an image object and its neighbors of a selected class with lower feature values than the object itself divided by the difference of the image object and all its neighbors of the selected class. Note that the features are weighted with the area of the corresponding image object.

8.5.4 Example: CustomizedFeatures.dpr

Test your knowledge about customized features with a small test project.

Try to solve the following sequence of tasks:

1. Create a new project with a provided image. The path is: **C:\Program Files\Definiens Developer** *Version number***\Examples \CustomizedFeatures\CustomizedFeatures.tif**.
2. Perform a chessboard segmentation with object size 1.
3. Create a customized arithmetic feature normalizing the brightness to values within [0; 1].
4. Classify objects with a **normalized brightness >= 0.5** as **RelativeDark**.
5. Classify objects with a **normalized brightness < 0.5** as **RelativeBright**.
6. Create a customized relational feature (**relativeFeatureA**) that returns the maximum normalized brightness of all neighboring objects within 5 pixels.
7. Classify objects with **relativeFeatureA > 0.5** as **RelativeDark**.
8. Create a customized relational feature (**relativeFeatureB**) that returns the minimum normalized brightness of all neighboring objects within 5 pixels.
9. Classify objects with **relativeFeatureB <= 0.5** as **relativeBright**.

Check your result with the provided test project. Check the additional information in the comments of the processes.The project can be found under **C:\Program Files\Definiens Developer** *Version number***\Examples \CustomizedFeatures\CustomizedFeatures.dpr**.

8.6 Reuse Process Sequences with Customized Algorithms

Customized algorithms enable you to reuse process sequences several times in one or different rule sets. Based on a developed process sequence, representing the developed code, you can create and reuse your own customized algorithms.

In contrast to duplicating a process, the main advantage of creating customized algorithms, is that when you want to modify the duplicated process you need to perform the changes to each instance of this process. However, with customized algorithms you only need to modify the customized algorithm template and the changes will take effect to every instance of this algorithm.

8.6.1 About the Scope of Rule Set Items of a Customized Algorithm

A customized algorithm includes rule set items like classes, image object levels, features, and variables. You can define different scopes for most of these rule set items.

- Limit rule set items to a **local** scope:
  The concept of the customized algorithms enables you to limit the usage of rule set items to a local scope. This means that such rule set items are visible only inside the customized algorithm. Moreover, they are used only within the scope of the child process represented by the customized algorithm. The advantage of the local scope is that the main rule set where the customized algorithm is embedded is not polluted by rule set items.
  The local scope is realized by a copy of the rule set item. This copy is placed in the local scope of the customized algorithm. The local rule set item is only available within customized algorithm and cannot be viewed neither used outside.

- Define a rule set item to a **global** scope:
  Global rule set items are useful for small customized algorithms that always are used in the same environment and use the current status of variables. Global rule set items are not recommended if the customized algorithm shall be used in different rule sets

- Define a rule set item as **parameter**:
  You can define rule set items used in a customized algorithm as editable parameters within the customized algorithm. Defining customized algorithm parameter is a comfortable method to reuse a customized algorithm in different rule sets, analysis contexts, or for different image types.
  The parameter concept is realized by a local copy of the rule set item. Due to this, there are copies of the rule set item listed in the respective controls

Dependencies of Rule Set Items

Different rule set items can be related by dependencies. Such rule set items are called as follows:

- A rule set item that is used by another rule set item is called **dependent**. For example, **classA** which is using the feature **Area** and the customized feature **Arithmetic1** has two dependencies: **Area** and **Arithmetic1**.

- A rule set item that is using another rule set item is called **reference**. For example, the feature **Area** is used by **classA** and the customized feature **Arithmetic1**. Then **Area** has two references: **classA** and **Arithmetic1**.

Dependencies of rule set items used in customized algorithms concern their scope. For example, let us say a process is using the **classA**, which is using a customized feature **Arithmetic1**. If you want to define **Arithmetic1** as **local** within the customized algorithm, then **classA** should become **local** too. Otherwise an inconsistent situation occurs, where a global class is using a local feature of the customized algorithm.

Scope Consistency Rules

Scope dependencies of rule set items used in customized algorithms are handled automatically according to the following consistency rules. These rules ensure the consistent handling of scopes all over a rule set.

A. If a rule set items is defined as **global** then all its references and dependents have to become **global** as well. If at least one dependent or referring rule set item cannot be defined as global this usage type should not be allowed for the rule set item. An exception exists for features without dependents (like **Area**): If they are defined as **global** then their references are not affected.

B. If a rule set items is defined as **local** or as **parameter** then all its references and dependencies have to become **local**. If at least one dependent or referring rule set item cannot defined as **local**, then this scope should not be allowed for the rule set item. An exception exists for features without dependents (like **Area**): They still remain global, as it makes no sense to make a local copy of features like **Area**.

Your Scope Definition of a	Automatic Impact on Scope of	Automatic Impact on Scope of
Class	Class dependent features	Sub-/superclasses
global	global	global
local	local	local
parameter	local	local
Level	Level dependent features	
global	global	
local	local	
parameter	local	
Variable	Variable features	Features or classes using the variable
global	global	global
local	local	local
parameter	local	local
Independent feature	Dependent features	Classes using the feature
global	(Not affected)	(Not affected)
parameter	local	local
Class-related feature	Derived features	Classes using the feature
(auto setting: local[1)])	local	local
(auto setting: global[1)])	global	global
parameter	local	local
Feature dependent feature	Derived features	Classes using the feature
(auto setting: local[1)])	local	local
(auto setting: global[1)])	global	global
parameter	local	local

[1)] The selection of scopes is automatically restricted here because these features are references of another rule set item.

8.6.2 Create a Customized Algorithm

Create customized algorithms based on a process sequence.

1. To create a customized algorithm, go to the **Process Tree** window and select the parent process of the process sequence that you want to use as customized algorithm. Do one of the following:

   - Right-click the parent process and select **Create Customized Algorithm** from the context menu.
   - Select **Process** > **Process Commands** > **Create Customized Algorithm** on the main menu bar.

   The **Customized Algorithms Properties** dialog box opens.

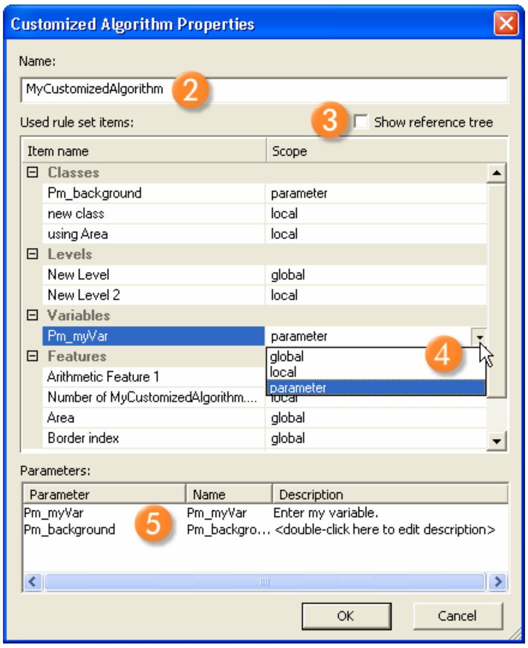


Figure 236: The Customized Algorithms Properties dialog box.

2. Assign a descriptive name for the customized algorithm.

3. The **Used rule set items** are arranged in groups. If you want to investigate their dependencies, select the **Show reference tree** check box.

→ About the Scope of Rule Set Items of a Customized Algorithm on page 276

4. You can modify the scope of rule set items. Select one in the list and then click the drop-down arrow button to select a scope of the rule set item.


(drop-down arrow button)

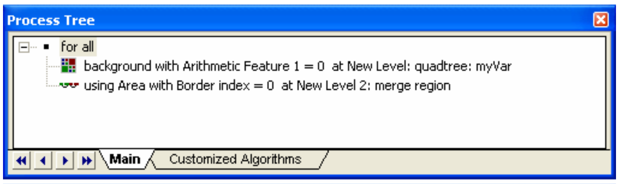
→ About the Scope of Rule Set Items of a Customized Algorithm on page 276

Scope	Usage
local	Uses the class, level, feature or variable only internally. Other processes will not be able to access them.
global	Use the class, level, feature or variable globally. They will be created and are accessible by other processes as well.
parameter	Allows assignment of a class, level, variable, or feature value within the **Algorithm parameters** of the **Edit Process** dialog box.

5. If you defined a rule set item as **Parameter** it is listed below. We recommended to assign a descriptive name and to add a description. This description will appear in

the help box at the bottom of the **Edit Process** dialog box when selecting this parameter.

6. After reconfirming with **OK**, the processes of the customized algorithm are displayed on a separate **Customized Algorithms** tab of the **Process Tree** window.



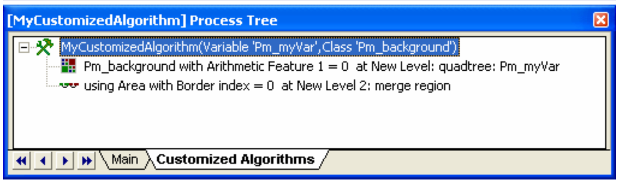


Figure 237: The original process sequence (above) is the base for the customized algorithm on a separate tab (below).

7. Customized algorithms can be selected at the bottom of the algorithm drop-down list box in the **Edit Process** dialog box as any other algorithm.

→ Define the Algorithm on page 133

The used classes are display in explicit sections within the **Class Hierarchy** window.

Figure 238: The Class Hierarchy window groups the classes of customized algorithms according to their scope.

The local features and feature parameters are displayed in the feature tree of the **Feature View** window using the name of the customized algorithm , for example **MyCustomizedAlgorithm.ArithmeticFeature1**.

The local variables and variable parameters can be checked in the **Manage Variables** dialog box. They use the name of the customized algorithm as a prefix of their name, for example **MyCustomizedAlgorithm.Pm_myVar**.

→ Manage Variables on page 261

The image object levels can be checked by opening the **Edit Process** dialog box of any process of the customized algorithm. Within the Image Object Domain select the current image object level or **New Level**. Click the **Parameter** button to open the **Select Level** dialog box. All image object levels are listed; the local ones and image object level parameters use the name of the customized algorithm as a prefix of their name, for example **MyCustomizedAlgorithm.Pm_myVar**.

Edit Parameter Default Values

8. Finally, you can define or change the default value of the parameters of the customized algorithm. The parameter defaults are used when the customized algorithm is selected in the **Edit Process** dialog box. Here they can be modified as usual.

To edit the parameter default values, select the customized algorithm and select **Process > Process Commands > Edit Customized Algorithm Default Parameters** on the main menu bar.

The **Default Parameter For '...'** . dialog box opens. Select a **Parameter** on the left and edit its default **Value** on the right.

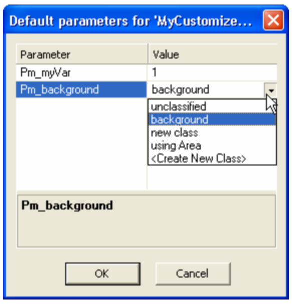


Figure 239: Default Parameter For '...' customized algorithms dialg box.

8.6.3 Use a Customized Algorithm in the Original Rule Set

Once you have created a customized algorithm, it appears on the **Customized Algorithms** tab of the **Process Tree** window. The objects you specified as **Parameter** are displayed in parenthesis following the algorithm's name.

The customized algorithm will not be directly executed, but rather, acts as a template for all new processes which are derived from it.

1. To create a new instance of a customized algorithm, right-click the **Process Tree** window, and from the context menu select **Append New**.

→ Create a Single Process on page 132

2. In the **Edit Process** dialog box, select the customized algorithm from the **Algorithm** drop-down list box.
3. Modify the **Algorithm parameters** according to your needs, then and click **OK**. Now, you have integrated the customized algorithm into your rule set and you are ready to execute it.

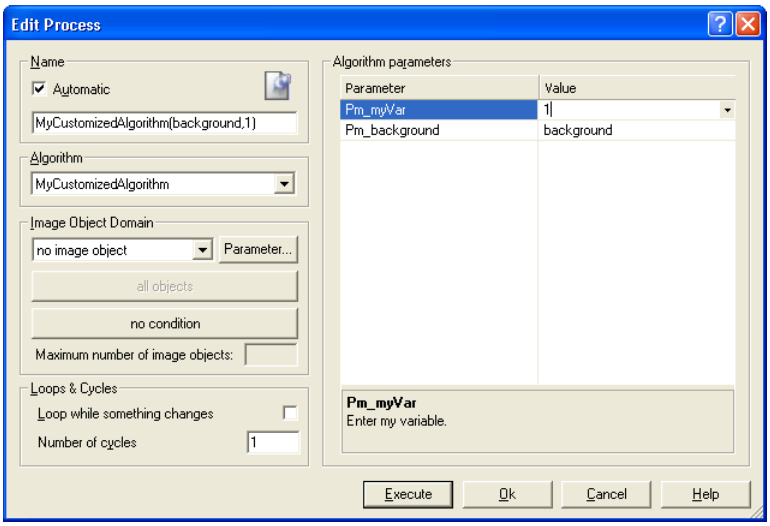


Figure 240: Edit Process dialog box with selected customized algorithm.

How Customized Algorithms are Displayed in the Process Tree

The new customized algorithm may be displayed in various locations in the process tree, in this example you can see three instances of the new algorithm:

1. The original algorithm 1 from which the new customized algorithm was created.
2. The new customized algorithm 2, which works as a template for all new processes derived from it.
3. The instance of the customized algorithm 3 created in order to be able to execute the new algorithm (multiple instances are possible).

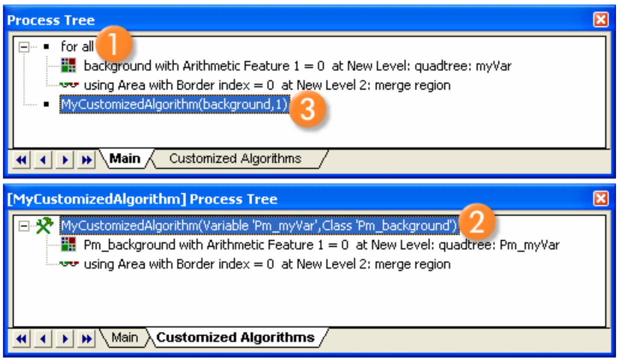


Figure 241: Occurances of a customized algorithm in the process tree.

8.6.4 Modify a Customized Algorithm

When you want to modify duplicated process, normally you need to perform the changes to each instance of this process. However, with customized algorithms you only need to modify the customized algorithm template and the changes will take affect to every instance of this algorithm.

> **Note**
>
> Modifying the processes inside the **Customized Algorithms** tab affects all instances derived from it within the rule set.
>
> **Note**
>
> If a feature is defined as **parameter**, the scope cannot be changed. As **parameter** is a special feature, it is not necessarily possible to reconvert it to the original. This restriction applies to the features only. However, this restriction automatically affects the choice for all rule set items that are references or dependents of this feature.
> As a workaround we recommend that you delete the customized algorithm and create it again.

1. To modify a customized algorithm select it on the **Customized Algorithms** tab of the **Process Tree** window. Do one of the following:

- Double click it.
- Select **Process** > **Process Commands** > **Edit Customized Algorithm** on the main menu bar.
- In the context menu, select **Edit Customized Algorithm**.

The **Customized Algorithm Properties** dialog box opens.

2. You can modify the scope of rule set items and change parameter settings.

→ Create a Customized Algorithm on page 278

8.6.5 Delete a Customized Algorithm

You can delete unwanted or unused customized algorithms.

Note

The customized algorithm and all its instances will be deleted without reconfirmation.

1. To delete a customized algorithm select it on the **Customized Algorithms** tab of the **Process Tree** window. Do one of the following:
   - Select **Delete** from the context menu.
   - Select **Process** > **Process Commands** > **Delete** on the main menu bar.
   - Press Del on the keyboard.
2. The customized algorithm and all its instances within a rule set is deleted. This will also remove them from the **Algorithm** drop-down list within the **Edit Process** dialog box.

8.6.6 Use a Customized Algorithm in Another Rule Set

In order to import a customized algorithm to other rule sets it is necessary to export it first.

1. Right-click on an instance process of your customized algorithm and choose **Save As** from the context menu. The parameters of the exported process serve as default parameters for the customized algorithm.
2. You may then load this algorithm to any project by selecting **Load Rule Set** from the context menu in the **Process Tree** window. The customized algorithm will appear at the end of your process tree.
3. To use the imported customized algorithm you only need to select it from the **Algorithm** drop-down list in the **Edit Process** dialog box.

8.7 Working with Thematic Layers

Use thematic layers to import external data to projects or to export result information to file.

Thematic layers are specific vector or raster files which have an associated attribute table. Generally, thematic layers are used for storing and transferring intermediate result information. For remote sensing purposes, importing and exporting thematic layers represent an interface to Geographic Information Systems (GIS).

→ Use Case Advanced: Transfer Results on page 301

Thematic layers include one type of objects only: polygons, lines, or points. While image layers contain continuous information, the information of thematic layers is discrete. Image layers and thematic layers have to be treated differently in both segmentation and classification.

8.7.1 Generate a Thematic Layer

Generate a thematic layer by exporting your image objects.

Image objects or classifications can be exported together with their attribute tables.

Use the following methods to export image objects as a thematic layer:

- Press the **Save Thematic Layer As** button on the **Manual Editing** toolbar.


Save Thematic Layer As

→ Save Thematic Objects in Thematic Layer on page 397

- Choose **Export > Export Results** from the main menu bar and select shapefile as export format.

→ Export Results as Shape File on page 404

- (**Definiens Developer** and **Professional** only): Insert a new process and choose **Export Vector Layers** from the **Algorithms** drop-down list to export your image objects as shapefile of type polygon, point or line.

→ Reference Book

- (**Definiens Developer** and **Professional** only): Insert a new process and choose **Export Thematic Raster Files** from the **Algorithms** drop-down list to export your image objects as **.tif** or other raster file format.

The georeferencing information as provided when creating a project will be exported as well.

8.7.2 Supported Thematic Layer File Formats

In principal all import formats supported for image layers are also supported for thematic layers; additionally the shapefile format of the type polygon is supported.

→ Supported Image Formats on page 47

8.7.3 Import a Thematic Layer

A thematic layer can be added when creating a new project or imported into a running project.

→ Create a New Project on page 44

→ Modify a Project on page 54

Note

It is only possible to load shape files of the shape type polygon. Other shape file types like point or line are not supported.

Vector data needs to be rasterized for import into **Definiens**. A thematic raster **.tif** file is created and saved on the hard disc in the same folder as the shape file. The folder where the shape file is stored cannot be a read-only file. Therefore, you cannot import a shape file from a CD-ROM drive, for example.

For polygon shape files, the appropriate **.dbf** file is automatically selected as attribute table. For all other formats, the respective attribute table must be specifically indicated in the **Load Attribute Table** dialog box which opens automatically.

From the **Load Attribute Table** dialog box, choose one of the following supported file formats:

- **.txt** (ASCII text files)
- **.dbf** (Dbase files)
- **.csv** (Comma-separated values files)

When loading a thematic layer from a multi-layer image file (for example, **.img** stack file), the appropriate layer that corresponds with the thematic information is requested in the **Import from multi-layer image** dialog box. Additionally, the attribute table with the appropriate thematic information must be loaded.

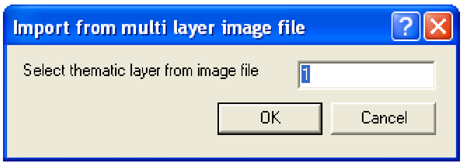


Figure 242: The Import From Multi Layer Image File dialog box.

If you import a thematic layer into your project and **Definiens** does not find an appropriate column with the caption ID in the respective attribute table, the **Select ID Column** dialog box will open automatically. Select the caption of the column containing the polygon ID from the drop-down menu and confirm with **OK**.

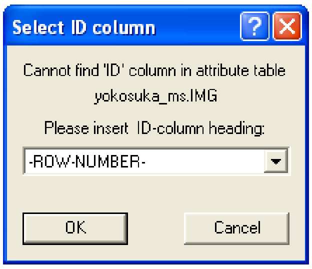


Figure 243: The Select ID Column dialog box.

8.7.4 Add a Thematic Layer to an Existing Project

1. To add a thematic layer into a running **Definiens** project, choose **File > Modify Open Project** from the main menu bar. The **Modify Project** dialog box opens.

→ Modify a Project on page 54

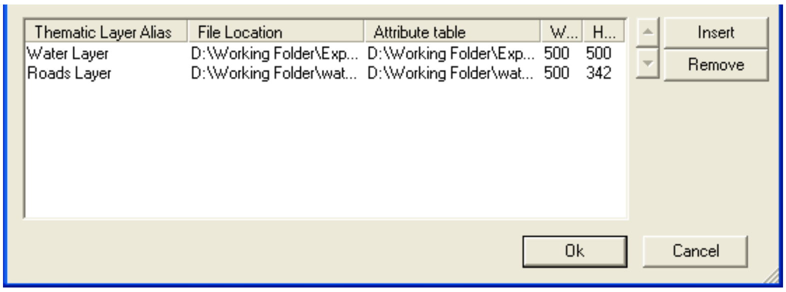


Figure 244: The Thematic Layer area of the Modify Project dialog box.

2. Insert a thematic layer into the **Thematic Layer** area as follows:

- Press the **Insert** button.
- Right-click inside the **Thematic Layer** area and choose **Insert** from the context menu.

3. The **Import Thematic Layer** dialog box opens. Browse to the thematic layer to be imported and press **Open**. If prompted, select the appropriate attribute table.

8.7.5 Display a Thematic Layer

Display a thematic layer in the **Project View**.

1. To display a thematic layer, open the **View Settings** window: Choose **View > View Settings** from the main menu.

→ View Settings on page 463

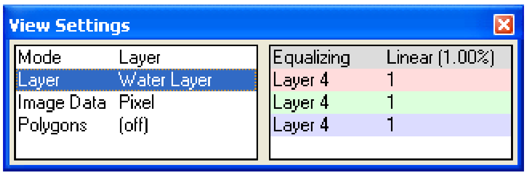


Figure 245: View Settings window with selected thematic layer.

2. Right-click the **Layer** row and select the layer you want to display from the context menu. The bottom part of the context menu lists the available thematic layers.

3. The thematic layer is displayed in the **Project View**.
   Each thematic object is displayed in a different random color.

Figure 246: Display of a thematic layer (type: polygon) in the Project View.

4. To return to viewing your image data, click the **Layer** row in the **View Settings** window and select **Image Data**.

8.7.6 Delete a Thematic Layer

To delete a thematic layer from a running project, open the **Modify Project** dialog box by choosing **File > Modify Open Project**. From the **Thematic Layers** area, select the thematic layer you want to delete and do one of the following:

- Press **Remove**.
- Right-click the thematic layer name and choose **Remove** from the context menu.

For more information on the **Modify Project** dialog box, see the **Modify a Project** section.

→ Modify a Project on page 54

8.7.7 View Thematic Layer Attributes

View the numeric attributes of thematic objects in the **Thematic Layer Attribute Table**.

To open the **Thematic Layer Attribute Table** window, select one of the following options from the main menu bar:

- **Tools >Thematic Layer Attribute Table**
- **View > Windows >Thematic Layer Attribute Table**

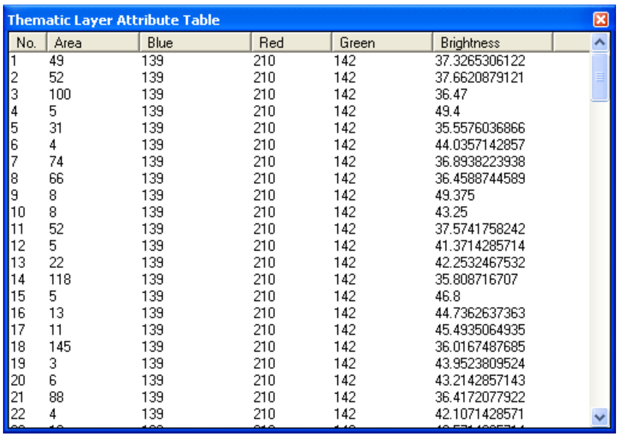
Thematic Layer Attribute Table

The **Thematic Layer Attribute Table** window opens without content.

To view the thematic attributes, open the **Manual Editing** toolbar. Choose **Thematic editing** as active editing mode and select a thematic layer from the **Select thematic layer** drop-down list.

→ Change Editing Mode on page 391

The attributes of the selected thematic layer are now displayed in the **Thematic Layer Attribute Table**. They can be used as features in the same way as any other feature provided by **Definiens**.



Thematic Layer Attribute Table

No.	Area	Blue	Red	Green	Brightness
1	49	139	210	142	37.3265306122
2	52	139	210	142	37.6620879121
3	100	139	210	142	36.47
4	5	139	210	142	49.4
5	31	139	210	142	35.5576036866
6	4	139	210	142	44.0357142857
7	74	139	210	142	36.8938223938
8	66	139	210	142	36.4588744589
9	8	139	210	142	49.375
10	8	139	210	142	43.25
11	52	139	210	142	37.5741758242
12	5	139	210	142	41.3714285714
13	22	139	210	142	42.2532467532
14	118	139	210	142	35.808716707
15	5	139	210	142	46.8
16	13	139	210	142	44.7362637363
17	11	139	210	142	45.4935064935
18	145	139	210	142	36.0167487685
19	3	139	210	142	43.9523809524
20	6	139	210	142	43.2142857143
21	88	139	210	142	36.4172077922
22	4	139	210	142	42.1071428571

Figure 247: Thematic Layer Attribute Table window.

The table supports the following formats: integers, strings and doubles. The column type is set automatically, according to the attribute. Table column widths can be up to 255 characters.

Class name and **Class color** are available as features and can be added to the **Thematic Layer Attribute Table** window.

→ Reference Handbook

8.7.7.1 Modify Thematic Layer Attribute Table

Modify a thematic layer attribute table by adding, editing or deleting table columns or editing table rows.

Add a New Column

Add a new table column by using the **Edit Attribute Table Column** dialog box.

1. Right-click the **Thematic Layer Attribute Table** and select **Add New Column** from the context menu.
   The **Edit Attribute Table Column** dialog box opens.

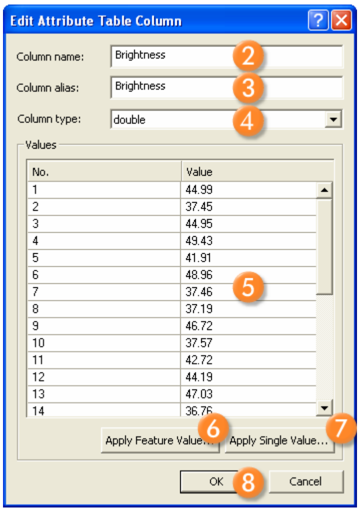


Figure 248: The Edit Attribute Table Column dialog box.

2. Enter a **column name** 2.
3. Enter a **column alias** 3.

4. Select a **column type** from the drop-down list 4:
   - double
   - integer
   - string
5. The **Values** group box displays the number and value of the attributes. Values can be entered or edited directly in the **Value** text fields 5.
6. To apply certain feature values, press the **Apply Feature Value** button 6.
   The **Select Single Feature** dialog box opens.
7. To apply the same value to all attributes, press the **Apply Single Value** button 7.
8. Press **OK** 8 to confirm your settings.

Edit Column

1. To edit an attribute table column, right-click the **Thematic Layer Attribute Table** and select **Edit Column** from the context menu. The **Choose Attribute Table Column** dialog box opens.

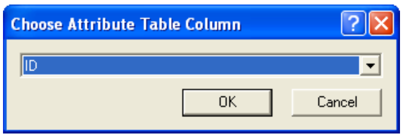


Figure 249: The Choose Attribute Table Column dialog box.

2. From the drop-down list, select the column that you want to edit and confirm with **OK**. This opens the **Edit Attribute Table Column** dialog box. For further details, see the related section.

→ Add a New Column on page 287

Delete Column

Delete a column from the attribute table.

1. Right-click the **Thematic Layer Attribute Table** and select **Delete Column** from the context menu. The **Choose Attribute Table Column** dialog box opens.

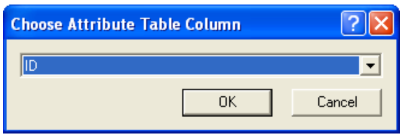


Figure 250: The Choose Attribute Table Column dialog box.

2. From the drop-down list, select the column that you want to delete and confirm with **OK**. The selected column is deleted from the attribute table.

Edit Row

Change the values of objects in the thematic attribute table.

1. Right-click the **Thematic Layer Attribute Table** and select **Edit Row** from the context menu.The **Edit Object Attributes** dialog box opens.

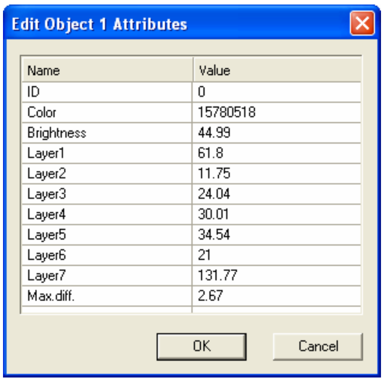


Figure 251: The Edit Object Attributes dialog box.

2. Enter appropriate values for the attributes in the **Value** field and press **OK** to confirm.

8.7.8 Use a Thematic Layer for Segmentation

Use a thematic layer for the segmentation of your input data into image objects.

Include the information of a thematic layer in the segmentation and thus influence the generation if image objects.

In contrast to image layers, thematic layers contain discrete information. This means that related layer values can carry different information, which itself is defined in the attached attribute list. To be able to clearly define the affiliation of an object to a thematic class given by the thematic layer, it is not possible to create image objects which belong to different thematic classes. To ensure this, the borders separating different thematic classes are restrictive for further segmentation whenever a thematic layer is used. For this reason, thematic layers cannot be given different weights, but can merely be selected for use or not.

If you want to produce image objects based exclusively on thematic layer information, you have to switch the weights for all image layers to 0. You can segment an image also using more than one thematic layer. The results are image objects representing proper intersections between the layers.

> **Note**
>
> If a thematic layer is not used for segmentation, it is consequently not possible to use the information given by this thematic layer for classification purposes.

1. To perform a segmentation using thematic layers, choose one of the following segmentation types from the **Algorithms** drop-down list of the **Edit Process** dialog box:

→ Create a Single Process on page 132

   - multiresolution segmentation
   - spectral difference segmentation
   - multiresolution segmentation region grow

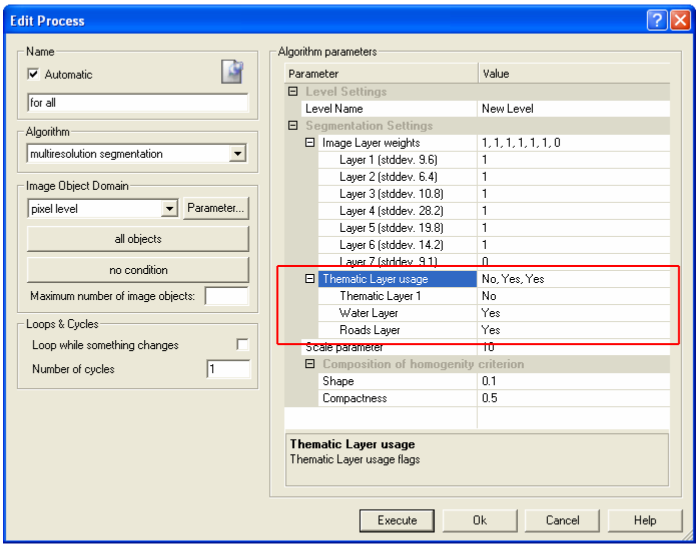


Figure 252: Define the Thematic layer usage in the Edit Process dilog box.

2. In the **Algorithm parameters** area, expand the **Thematic Layer usage** list and select the thematic layers to be considered in the segmentation. You can use both following methods:

   - Select an thematic layer and click the drop-down arrow button placed inside the value field. Define for each the usage by selecting **Yes** or **No**.
   - Select **Thematic Layer usage** and click the ellipsis button placed inside the value field to open the **Image Layer Weights** dialog box.

 (drop-down arrow button)

 (ellipsis button)

→ Image Layer Weights on page 447

8.8 Working with Polygons and Skeletons

Polygons are vector objects which provide new and more detailed information for the characterization of image objects by their shape.

Polygons are also needed for the visualization and export of image object outlines.

Skeletons, which describe the inner structure of an image object, help to describe an object's shape more accurately.

Use polygon and skeleton features to define your class descriptions or to refine your segmentation.

8.8.1 About Vectorization of Polygons

Image objects can be represented simultaneously as raster or vector objects. After segmentation, vectorization functionality allows the production of polygons for each image object. This vector information is produced in different resolutions for different purposes.

For rendering outlines of image objects independent of the zoom value, polygons along the pixel raster or slightly abstracted polygons are produced. The abstracted polygons are referred to as **base polygons**. They are created with respect to the topological structure of image objects and are used for exporting vector information as well. More abstracted vector information represents the shape of image objects independent of the topological structure and is used for the computation of shape features. Theses polygons are referred to as **shape polygons**.

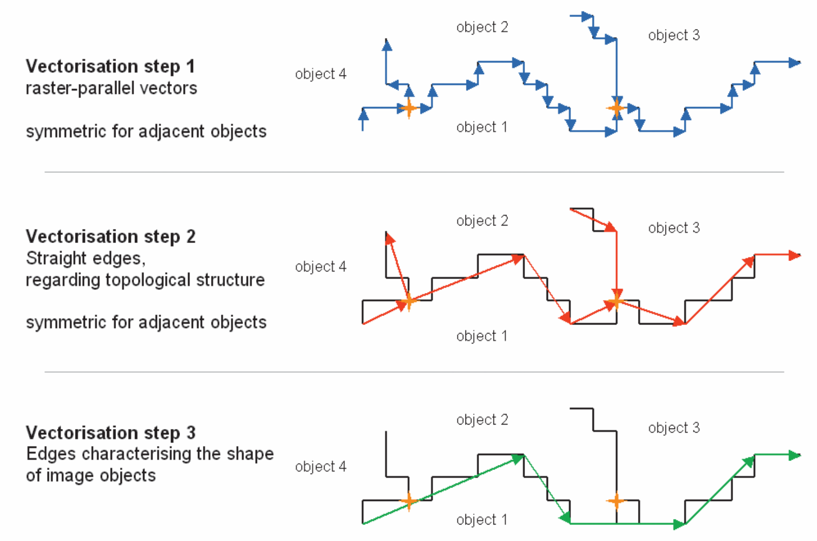


Figure 253: Different polygon types for the vectorization of segments / image objects

Base Polygons

The computation of base polygons is done by means of a **Douglas-Peucker** algorithm.The **Douglas-Peucker** algorithm is one of the most common procedures for polygon extraction. It is a top-down approach, which starts with a given polygon line and divides it iteratively into smaller sections. Given the two end points of a polygon line—typically, in **Definiens Developer** these two starting points are topological points (see yellow marks in the figure above)—the algorithm detects this specific point on the polygon line with the largest vertical distance to a line connecting the two end points.

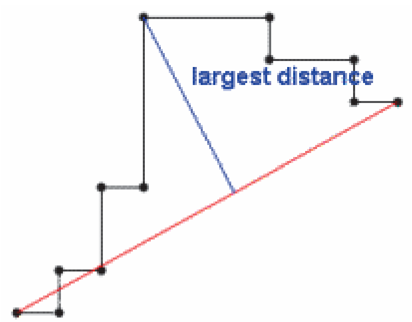


Figure 254: Start configuration and detection of largest distance

At this detected point, the polygon line is cut into two shorter polygon lines.

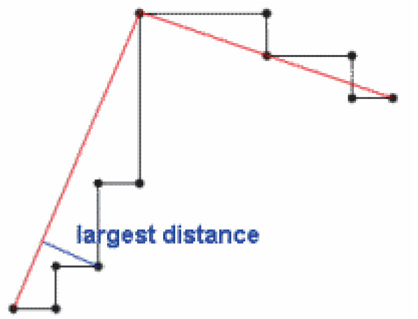


Figure 255: New state after dividing into two sections

This procedure continues until the longest vertical distance is smaller than a given threshold.

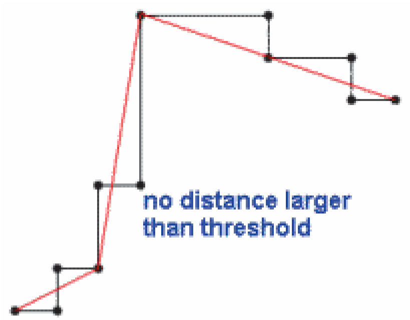


Figure 256: Final result. No further division as no distance is larger than the given threshold.

In other words: the threshold describes the strongest possible deviation of the polygon from the underlying raster.

In **Definiens Developer**, this threshold can be defined in the **set rule set options** algorithm parameters and is measured in pixel units.

The **Douglas-Peucker** algorithm in its pure application produces relatively acute angles in some cases. Therefore, in order to improve the result, angles smaller than 45 degrees are detected in a second run. From the two particular vectors at such an angle, that one is subdivided which will result in the largest angles. This procedure continues in iterative steps until there are no angles smaller than 45 degrees.

For high thresholds, which produce a strong abstraction from the original raster, slivers and intersections within and between base polygons can arise. This can be especially disturbing when these base polygons are used for export. In order to avoid this effect, an additional, optional algorithm detects intersections and fractionizes the affected vectors.

Shape Polygons

The shape polygons are created by means of a derivative of the multiresolution segmentation, in this case not applied to image regions but to single vectors. In contrast to the **Douglas-Peucker** algorithm, this procedure is a bottom-up approach. Starting with base polygons, the single vectors are subsequently merged, optimizing a homogeneity criterion. It is important to understand that the heterogeneity of single shape vectors is defined as deviation of the underlying base vectors. Thus, a threshold of 0 will always produce shape polygons identical to the underlying base polygons. The resulting shape therefore depends also on the threshold of the base polygons. A threshold bigger than 0 will result in a stronger abstraction than the base polygons.

Concretely, the deviation is computed as the maximum of the difference of length between shape vector and underlying base vectors and the sum of the lengths of the vertical parts of the underlying base vectors to the shape vector. Iteratively, the two adjacent vectors of a polygon which result in the smallest heterogeneity are merged. This continues until the predefined threshold is reached.

The resulting shape polygons are independent of the topological structure and therefore specific for each single image object. A straight edge of a segment is represented as one vector, even if it contains a topological point (see figure above, yellow marks). Thus, fractally shaped parts of the boundary of an image object are represented in a characteristic way by a number of short vectors, whereas straight edges are represented by long edges. Based on these shape polygons, a number of different shape features can be computed.

8.8.1.1 About the Display of Polygons

You can use the settings in the **set rule set options** algorithm to change the way that polygons display.

→ **View Polygons** on page 112
→ **Reference Book**

Polygons are available after the initial segmentation of an image. If the polygons cannot be clearly distinguished due to a low zoom value, they are automatically deactivated. Therefore, a higher zoom value must be chosen in order to see the polygons.

8.8.2 About Skeletons

Skeletons describe the inner structure of an object. By creating skeletons, the object's shape can be described more accurately.

To obtain skeletons, a **Delaunay** triangulation of the objects' shape polygons is performed. The skeletons are then created by identifying the mid-points of the triangles and connecting them. Thus, the mid-points of the triangles are determined by connecting the mid-points of the triangles' sides.

To find skeleton branches, three types of triangles are created:

- **End triangles** (one-neighbor-triangles):indicate end points of the skeleton 1.
- **Connecting triangles** (two-neighbor-triangles): indicate a connection point 2.
- **Branch triangles** (three-neighbor-triangle): indicate branch points of the skeleton 3.

To obtain the skeleton, the generated points become connected. In doing so, the main line is represented by the longest possible connection of branch points. Beginning with the main line, the connected lines then are ordered according to their types of connecting points.

The branch order is comparable to the stream order of a river network. Each branch obtains an appropriate order value: The main line always holds a value of 0 while the outmost branches have the highest values, depending on the objects' complexity.

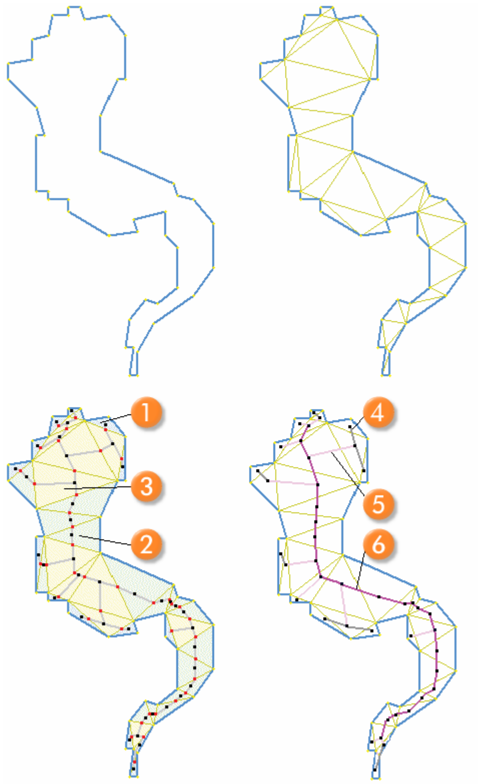


Figure 257: Skeleton creation based on a Delauney triangulation
The right image shows a skeleton with the following branch order:

4 Branch order = 2.
5 Branch order = 1.
6 Branch order = 0 (main line).

8.8.2.1 Display Skeletons

Polygons are created automatically, and the settings for polygons **Project Settings** group of the **Options** dialog box controls the way skeletons are abstracted. To display skeletons, click the **Show/Hide Skeletons** button.

Show/Hide Skeletons

→ Options on page 452

> **Note**
>
> An object's skeleton is only displayed by selecting it.

To view skeletons of multiple objects, draw a polygon or rectangle using the **Manual Editing** toolbar to select the desired image objects and activate the skeleton view.

→ Selection Tools on page 391

8.8.3 Use Polygon and Skeleton Features

A number of shape features based on polygons and skeletons are available. These features are used in the same way as other features. They are available in the feature tree

→ Reference Handbook

under **Object features > Shape > Based on Polygons** or
Object features > Shape > Based on Skeletons.

8.9 Automate the Workspace Processing

Automate and accelerate the processing of data loaded to a workspace, especially when analyzing large scenes.

Detailed processing of a entire high resolution image can be time consuming and sometimes always feasible, due to memory limitations. Often such an image has only limited regions of interest. Workspace automation enables you to automate processing steps that follow user interactions, such as the manual selection of subsets representing regions of interest. Additionally, you can automate multi-scale workflows, which integrate analysis of images at different magnifications/resolutions.

Workspace automation is recommended for image analysis tasks on large images, some of which are commonly used in histopathology, for example, to analyze certain tissue to determine the amount of insulin in pancreas islets. Another application might be in remote sensing, for analysis of regions of interest such as water bodies.

8.9.1 About the Concept of Workspace Automation

You can program a workspace automation by developing structured rule sets that run different analysis step on scene copies. A scene copy can be a regular copy, a rescaled copy, a tiled copy, or a subset copy.

The concept of workspace automation is realized by structuring rule sets into subroutines. A subroutine is a separate part of the rule set cut off from the main process tree to be applied to a copy of the main scene.

Subroutines organize multiple scene copies for automated processing. They can be clearly arranged in tabs of the **Process Tree** window, as in the figure below.

→ Organize a Process Sequence on page 141

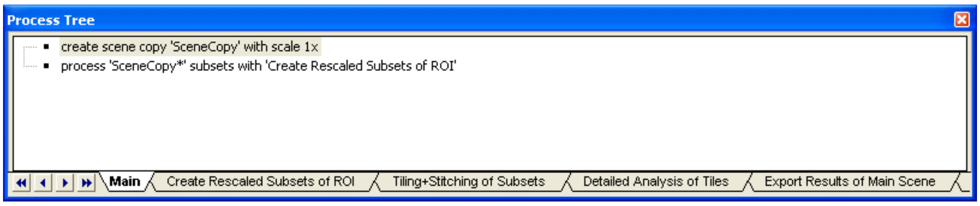


Figure 258: Subroutines are assembled on tabs in the Process Tree window.

Subroutines are executed by a dedicated process in the **Main** process tree or another subroutine.

Structuring a rule set into subroutines allows you to focus or limit analysis tasks to regions on interest. Thus, it avoids wasting processing performance for image areas of low interest. In addition, using subroutines enables you to create rule sets that run image analysis without any interruption for user input like the manual selection of subsets.

When processing occurs on different instances of the Analysis Engine Softwares, the main process tree can be executed simultaneously with defined processes of the subroutines.

8.9.2 Use Cases of Subroutines

To give you practical illustrations of structuring a rule set into subroutines, have a look at some typical use cases including samples of rule set code.

→ Notation on page 467

For detailed instructions, see the related instructional sections and the **Reference Handbook** listing all settings of algorithms.

→ Create Subroutines on page 301
→ Reference Handbook

8.9.2.1 Use Case Basic: Create a Scene Subset

Find regions of interest (ROI), create scene subsets, and submit for further processing.

In this basic use case, you use a subroutine to limit detailed image analysis processing to subsets representing the regions of interest (ROI). The image analysis processes faster because you avoid detailed analysis of other areas.

Commonly, you use this subroutine use case at the beginning of a rule set and therefore it is part of the main process tree on the **Main** tab. Within the main process tree, you sequence processes in order to find regions of interest (ROI) on a bright background. Let us say that the intermediate results are multiple image objects of a class **no_background** representing the regions of interest of your image analysis task.

Still editing within the main process tree, you add a process applying the **create scene subset** algorithm on image objects of the class **no_background** in order to analyze regions of interest only.

The subsets created must be sent to a subroutine for analysis. Add a process with the algorithm **submit scenes for analysis** to the end of the main process tree. It executes a subroutine that defines the detailed image analysis processing on a separate tab.

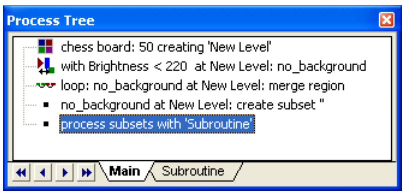


Figure 259: The Main process tree in the Process Tree window.

```
Main
    chess board: 50 creating 'New Level'
    with Brightness < 220  at New Level: no_background
    loop: no_background at New Level: merge region
    no_background at New Level: create subset ''
    process subsets with 'Subroutine'
```

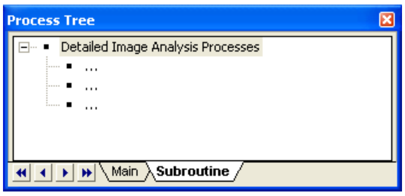


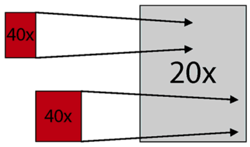
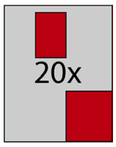
Figure 260: A subroutine in the Process Tree window.

```
Subroutine
    Further Processing
        ...
        ...
        ...
```

8.9.2.2 Use Cases: Multi-Scale 1–3

The multi-scale use cases sequence common subroutines use cases to describe the automation of a sample multi-scale workflows.

	Workflow	Scale	Subroutine	Key Algorithm	
1.	Create a scene copy at lower magnification	20x → 1x	Main	**create scene copy**	→ Multi-Scale 1: Rescale a Scene Copy on page 298
2.	Find regions of Interest (ROI)	1x	Create Rescaled Subsets of ROI	common image analysis algorithms	→ Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI) on page 299
3.	Create subsets of ROI at higher magnification	20x → 40x, 40x	Create Rescaled Subsets of ROI	**create scene subset**	
4.	Tile subsets	40x	Tiling+Stitching of Subsets	**create scene tiles**	→ Multi-Scale 3: Use Tiling and Stitching on page 300
5.	Detailed analysis of tiles	40x	Detailed Analysis of Tiles	several	
6.	Stitch tile results to subset results	40x, 40x	Detailed Analysis of Tiles	**submit scenes for analysis**	

7.	Merge subsets results back to main scene		Create Rescaled Subsets of ROI	**submit scenes for analysis**	→ **Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI)** on page 299
8.	Export results of main scene		Export Results of Main Scene	**export classification view**	

This workflow may serve you as a prototype of an analysis automation of an image at different magnifications/resolutions. However, when developing rule sets with subroutines, you must create a specific sequence tailored to your image analysis problem.

Multi-Scale 1: Rescale a Scene Copy

Create a rescaled scene copy at a lower magnification/resolution and submit for processing to find regions of interest.

In this use case, you use a subroutine to rescale the image at a lower magnification/resolution before finding regions of interest. In this way, you process faster because you will find regions of interest by running just a rough analysis instead of a detailed and time and performance consuming analysis. For the first process use the **create scene copy** algorithm.

→ **Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI)** on page 299

Note

About Merging Results of Subsets Back to the Main Scene

When working with subroutines you can—but you do not have to—merge back selected results to the main scene. This enables you to reintegrate results to the complete image and export them together.

To fulfill a prerequisite to merging results back to the main scene, you need to create a scene copy as the first step of the subroutine sequence. Likewise you keep the main scene free of any processing that creates image object levels.

→ ***Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI)*** *on page 299*

With the second process—based on the **submit scenes for analysis** algorithm—you submit the newly created scene copy to a new subroutine for finding regions of interest (ROI) at a lower scale.

Commonly this subroutine use case is used at the beginning of a rule set and therefore is part of the main process tree on the **Main** tab.

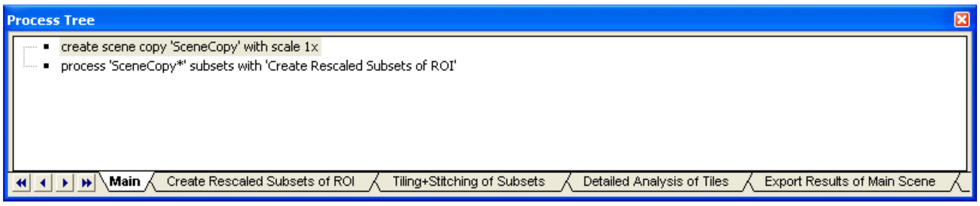


Figure 261: Subroutines are assembled on tabs in the Process Tree window.

```
Main
    create scene copy 'SceneCopy' with scale 1x
    process SceneCopy*' subsets with 'Create Rescaled Subsets of ROI'
```

Note

In this case the processed **subset** refers to a scene copy, in other words, a subset that covers the complete scene.

Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI)

Create rescaled subsets copies of regions of interest at a higher magnification/resolution.

In this use case, you use a subroutine first to find regions of interest (ROI) and classify them, for example, as **ROI**.

→ Multi-Scale 1: Rescale a Scene Copy on page 298

→ Multi-Scale 3: Use Tiling and Stitching on page 300

Based on the image objects representing the regions of interest (ROI) you create subset copies of regions of interest. Using the **create scene subset** algorithm, you can rescale them at a higher magnification/resolution. Because you want to get a detailed analysis of limited regions of interest, this scale will require more processing performance and time compared to the original scene scale.

Finally, you submit the newly created rescaled subset copies of regions of interest for further processing to the next subroutine. Use the **submit scenes for analysis** algorithm for such connections of subroutines.

```
Create Rescaled Subsets of ROI
    Find Regions of Interest (ROI)
        ...
        ...
        ...
    ROI at ROI_Level: create subset 'ROI_Subset' with scale 40x
    process 'ROI_Subset*' subsets with 'Tiling+Stitching of Subsets' and
stitch with 'Export Results of Main Scene'
```

Determine the Merging of Results of Subsets Back to the Complete Scene

Within the **submit scenes for analysis** algorithm, you can determine whether the results of the analysis of subset copies will later be merged back to the complete main scene. In addition, you can continue processing within a later subroutine, for example, to export the merged results of the main scene.

```
Export Results of Main Scene
    at ROI_Level: export classification by color to
ClassificationExportFile
```

Preconditions for merging back the results are:

- The main scene has not been processed and consequently has no image object level.
- All subsets have been processes with the same rule set.
- All processed subsets have the same number of image object levels.

→ Multi-Scale 1: Rescale a Scene Copy on page 298

If different subsets overlap, you can either merge the overlapping image objects or create intersection image objects of the overlapping area.

The method works similarly for scene copies as for scene subsets. However, because scene copies overlap completely, the handling of intersections becomes more important.

If you look for collecting and merging statistical analysis results of subscenes (tiles or subsets), use the **read subscene statistics** algorithm. It enables you to export a single statistic related to one scene.

→ Collect Statistical Results of Subscenes on page 305

Intersection handling of the **submit scenes for analysis** algorithm may result in performance intensive operations. Alternatively, you may consider the advanced **Transfer Results** use case.

→ Use Case Advanced: Transfer Results on page 301

Multi-Scale 3: Use Tiling and Stitching

Create tiles, submit for processing, and stitch the result tiles for postprocessing.

In this use case, you use a subroutine to create tiles using the **create scene tiles** algorithm.

→ Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI) on page 299

→ Tile and Stitch Large Scenes on page 375

Using the **submit scenes for analysis** algorithm, you submit them to a time and performance consuming processing which, in our example, is a detailed image analysis of rescaled region of interest (ROI) subsets at a higher scale. Generally, creating tiles before processing enables the distribution of the analysis processing on multiple instances of Analysis Engine Software.

In this example, after processing the detailed analysis within a separate subroutine, the tile results are stitched and submitted for postprocessing to the next subroutine. Stitching settings are done using the parameters of the **submit scenes for analysis** algorithm.

```
Tiling+Stitching Subsets
    create (500x500) tiles
    process tiles with 'Detailed Analysis of Tiles' and stitch
Detailed Analysis of Tiles
    Detailed Analysis
        ...
        ...
        ...
```

8.9.2.3 Use Case Advanced: Transfer Results

Transfer intermediate result information by exporting to thematic layers and reloading them to a new scene copy.

→ Working with Thematic Layers on page 283

This subroutine use case presents an alternative for using the merging results parameters of the **submit scenes for analysis** algorithm because its intersection handling may result in performance intensive operations.

→ Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI) on page 299

Here you use the **export thematic raster files** algorithm to export a geocoded thematic layer for each scene or subset containing classification information about intermediate results. This information, stored in a thematic layers and an associated attribute table, is a description of the location of image objects and information about the classification of image objects.

After exporting a geocoded thematic layer for each subset copy, you reload all thematic layers to a new copy of the complete scene. This copy is created using the **create scene copy** algorithm.

→ About Geocoding on page 50

The subset thematic layers are matched correctly to the complete scene copy because they are geocoded. Consequently you have a copy of the complete scene with intermediate result information of preceding subroutines.

Using the **submit scenes for analysis** algorithm, you finally submit the copy of the complete scene for further processing to a subsequent subroutine. Here you can use the intermediate information of the thematic layer by using thematic attribute features or thematic layer operations algorithms.

```
Advanced: Transfer Results of Subsets
    at ROI_Level: export classification to ExportObjectsThematicLayer
    create scene copy 'MainSceneCopy' (with additional Thematic Layers)
    process 'MainSceneCopy*' subsets with 'Further' (The subset covers the complete scene.)
Further
    Further Processing
        ...
        ...
        ...
```

8.9.3 Create Subroutines

Create a new subroutine if you want to create processes that will be applied to a scene copy. A scene copy may be a regular, rescaled, tiled or subset copy. In addition, you can create a new subroutine if you want ensure that all processes of a preceding subroutine have been processed completely.

1. Right-click a subroutine tab (for example **Main**) of the **Process Tree** window and select **Add New** on the context menu. A new **Subroutine** tab is added right of the existing subroutine tabs.
2. Type a new name to change the default name **Subroutine** *number*.
3. On the new subroutine tab, add processes to the subroutine.

→ Use Processes to Build Rule Sets on page 130

→ Connect Subroutines and Determine Stitching on page 302

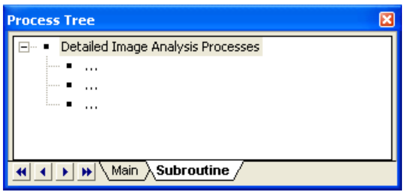


Figure 262: A subroutine in the Process Tree window.

4. To enable the execution of the subroutine you need to connect it with the main process tree or other subroutines. To do so, add a dedicated process to the main process tree or another subroutine that will trigger the execution.

8.9.4 Connect Subroutines and Determine Stitching

To enable the execution of a subroutine, you need to connect it with the main process tree or with other subroutines.

To do so, you have to add a dedicated process to the main process tree or another subroutine that will trigger the execution. Commonly, this is the previous subroutine or, if you have only one subroutine, the main process tree.

The process that triggers the execution of a subroutine uses the **submit scenes for analysis** algorithm. You can address any process in the process tree of a subroutine by using a slash mark **/** before hierarchy steps.

→ Reference Handbook

To transfer variables to following subroutines you can select parameter sets.

You can also determine whether the results of the analysis of subset copies are stitched together to obtain a complete main scene including results.

> **Note**
>
> **About Debugging**
>
> Developing and debugging with open projects by step-by-step execution of single processes or by setting break points does not work between subroutines. Within a subroutine, make sure that the correct project is open.

8.9.5 Create Scene Copies

A subroutine can be used to run processes on a copy of the scene. There are different kinds of scene copies available:

- Regular scene copy
- Rescaled scene copy
- Copy of a scene subset
- Rescaled copy of a scene subset
- Tiled copy

All scene copies use the given coordinates (geocoding or pixel coordinates) of the source scene. Copies include any thematic layers that are included in the scene.

Because a scene is the combined input data from a project or a workspace, scene copies do not include any results such as image objects, classes or variables.

8.9.5.1 Regular Scene Copy

A regular scene copy is just a duplicate of a project with image layers and thematic layers at the same magnification/resolution, but without any results like image objects, classes, or variables.

> **Note**
>
> If you create a scene copy for another subroutine, results are not copied as well. If you want to transfer results to another subroutine, you need to export a thematic layer describing the results before.
>
> → **Use Case Advanced: Transfer Results** on page 301

To create a scene copy, use the **create scene copy** algorithm. For detailed information have a look at the **Reference Handbook**.

→ **Reference Handbook**

8.9.5.2 Rescaled Scene Copy

A rescaled scene copy is a duplicate of a project, including image layers and thematic layers, that is rescaled at another, higher or lower magnification/resolution. It does not include any results such as image objects, classes or variables.

→ **Multi-Scale 1: Rescale a Scene Copy** on page 298

> **Note**
>
> To rescale at another magnification/resolution, you always have to create a scene copy.

To create a scene copy, use the **create scene copy** algorithm. For detailed information have a look at the **Reference Handbook**.

→ **Reference Handbook**

8.9.5.3 Copy of a Scene Subset

A copy of a scene subset is a project with a subset of image layers and thematic layers at the same magnification/resolution. It is used to create subsets of regions of interest (ROI).

→ **Use Case Basic: Create a Scene Subset** on page 296

The cutout position is the portion of the scene to be copied.

Depending on the selected **Image Object Domain** of the process you can define the cutout position and size:

- Based on coordinates: If you select **no image object** in the **Image Object Domain** drop down list box, the given coordinates (geocoding or pixel coordinates) of the source scene are used.
- Based on classified image objects: If you select an image object level in the **Image Object Domain** drop down list box you can select classes of image objects. For each image object of the selected classes a subset is created based on a rectangular cutout area around the image object.
  Other image objects of the selected classes are commonly located inside the cutout rectangle, typically near the border. You can choose to include or to exclude them

from further processing. Thus, you can extract regions of interest as separate subsets by extracting classified image objects as subset scenes.

Results are not included. You can create subset copies of an existing subset.

To create a scene copy, use the **create scene subset** algorithm. For detailed information have a look to the **Reference Handbook**.

→ Reference Handbook

8.9.5.4 Tiled Copy

A tiled copy of a scene is a collection of multiple projects, each with a tile of the image layers and thematic layers. Together the tile projects represent the complete scene as it appeared before creating the tiled copy. The given coordinates (geocoding or pixel coordinates) of the source scene of the rule set are used. Results are not included before processing of the tiles.

After processing, you can stitch the tile results together and add them to the complete scene within the dimensions as it has been before creating the tiled copy.

→ Multi-Scale 3: Use Tiling and Stitching on page 300

You can tile scenes and subsets several times.

To create a scene copy, use the **create scene tiles** algorithm. The stitching options are defined with the **submit scenes for analysis** algorithm. For detailed information, refer to the **Reference Handbook**.

→ Reference Handbook

8.9.5.5 Rescaled Copy of a Scene Subset

A rescaled copy of a scene subset is a project with a subset of image layers and thematic layers that is rescaled at another magnification/resolution. It is used to create rescaled subsets of regions of interest (ROI).

→ Multi-Scale 2: Create Rescaled Subset Copies of Regions of Interest (ROI) on page 299

Note

To rescale at another magnification/resolution, you must create a subset copy.

To create a scene copy, use the **create scene subset** algorithm.

→ Copy of a Scene Subset on page 303

8.9.6 Edit Subroutines

To edit a subroutine, you create a process sequence as usual.

→ Use Processes to Build Rule Sets on page 130

In addition you can use specific subroutines commands. Right-click a subroutine tab of the **Process Tree** window and select a command on the context menu.

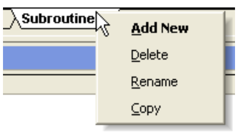


Figure 263: Subroutine commands on the context menu of the Process Tree window.

8.9.6.1 Move a Process to Another Subroutine

Move a process, including all child processes, from one subroutine to another subroutine by the usual copy-and-paste commands via context menu or keyboard shortcuts.

8.9.6.2 Rename Subroutine

By default, subroutines are named **subroutine** *number*. You can rename the subroutine to assign a descriptive name.

1. To rename a subroutine do one of the following:
   - Right-click a subroutine tab of the **Process Tree** window and select **Rename** on the context menu.
   - Double-click the subroutine tab label.
2. Edit the name and press Enter or just click outside the subroutine tab label.

8.9.6.3 Duplicate Subroutine

To duplicate a selected subroutine including all processes, right-click a subroutine tab of the **Process Tree** window and select **Duplicate** on the context menu.

8.9.6.4 Delete Subroutine

To delete a selected subroutine from the rule set, right-click a subroutine tab of the **Process Tree** window and select **Delete** on the context menu.

8.9.6.5 Save Subroutine

Subroutines are saved together with the rule set. Right-click the process tree and select **Save Rule Set** on the context menu.

8.9.7 Collect Statistical Results of Subscenes

Subscenes can be tiles or subsets. If the analysis of subscenes results in exporting statistics for each scene, you can collect and merge the statistical results of multiple files. The advantage is that you do not need to stitch the subscenes results for result operations concerning the main scene.

Preconditions:

- For each subscene analysis, at least one project or domain statistic has been exported.
- All preceding subscene analysis including export has been processed completely before the **read subscene statistics** algorithm starts any result summary calculations. To ensure this, result calculations are done within a separate subroutine.

→ Create Subroutines on page 301

After processing all subscenes, the **read subscene statistics** algorithm reads in the exported result statistics of the subscenes and performs a defined mathematical summary operation. The resulting value, representing the statistical results of the main scene, is stored as a process variable. This process variable can be used for further calculations or export operations concerning the main scene.

Example:

→ Notation on page 467

```
Main
    create (2000x2000) tiles
    process tiles with 'Tiles'
    process current scene with 'Read Tile Statistics'
Tiles
    Analysis
        ...
        ...
        ClassA at Level1: export domain statistics [Sum, Mean]
Read Tile Statistics
    read statistics for 'MainScene*' tiles: std. dev. of Sum Area (Pxl)
of export item 'DomainStatistics'
    Further Processing Options:
        calculations based on 'ClassAStdDev' variable
        export project statistics (exporting the 'ClassAStdDev'
variable)
```

8.9.8 Execute Rule Sets with Subroutines

A rule set with subroutines can be executed only on data loaded to a workspace. This enables you to review all projects of scenes, subset, and tiles. They all are stored in the workspace.

→ Start Automated Image Analysis on page 363

8.10 Create Action Libraries

An action library is a collection of action definitions which are just unconfigured actions. Action definitions enable the user of applications in products like **Definiens Architect** to specify actions and to assemble solutions. To make rule sets usable as unconfigured actions in action libraries you have to package them and give them a user interface.

In the **Analysis Builder** window you package pieces of rule sets, each of them solving a specific part of a solution, into action definitions. Here you group action definitions into libraries and define dependencies on actions. Further, you can create different user interface components (called widgets) the user of an action library may use to adjust action parameters.

For testing the created action libraries with relevant data, you can build analysis solutions in the **Analysis Builder** window.

→ Build Analysis Solution on page 315

8.10.1 About Creating Rule Sets for Actions

When developing rule sets for actions you should to keep some particularities in mind.

8.10.1.1 Create Parameters to Be Set by the User of Action Libraries

As the parameters of an action can be set by users of action libraries using products like **Definiens Architect**, you need to place adjustable variables in a parameter set.

You should use unique names for variables and must use unique names for parameter sets.

We recommend developing adjustable variables of a more general nature (eg. **Low Contrast**) which have influence on multiple features instead of having one control for one feature.

8.10.1.2 Use Unique Process Names

In rule sets that should be used for actions, avoid to have identically named parent processes.

This is especially important for proper execution if an **Definiens** action refers to inactive parts of a rule set.

8.10.1.3 About Creating a "Quick Test" Button

When creating a **Quick Test** button in an action, you need to implement a kind of internal communication to synchronize actions with the underlying rule sets. This is realized by integration of specific algorithms to the rule sets that organize the updating of parameter sets, variables, and actions.

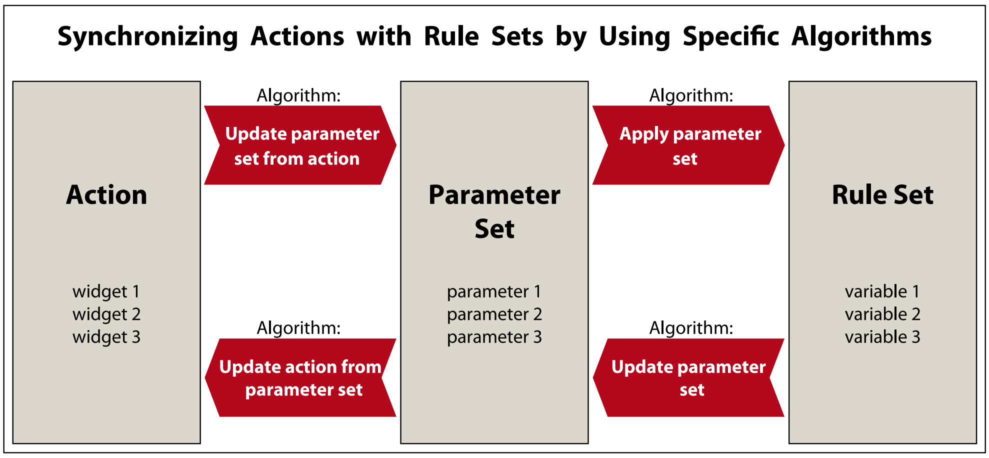


Figure 264: The communication between action and rule set is organized by algortihms (arrows).

These four specific algorithms are:

- **Update parameter set from action** and
- **Update action from parameter set**

to transfer values from the action to the parameter set and vice versa.

- **Update parameter set**
- **Apply parameter set**

to transfer values from the parameter set to the rule set and vice versa.

To get all parameters from the action to the rule set before you execute a **Quick Test**, you need a process sequence like this:

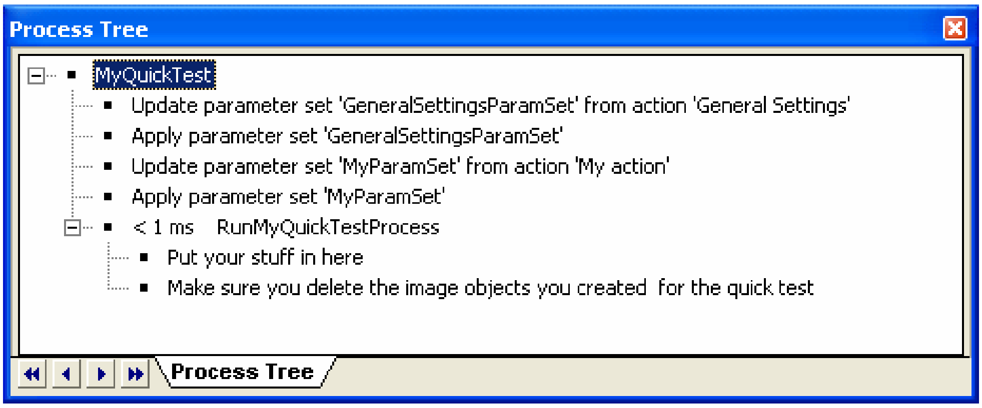


Figure 265: Sample process sequence for a Quick Test button within actions.

Note

You also need to update the general settings when your rule set relies on them.

You should restore everything to the previous state when the quick test is done.

8.10.1.4 About Maintenance of Rule Sets for Actions

The developed rule set (**.dcp** file) will probably be maintained by other developers. Thus we recommend that you structure the rule set clearly and document it by using meaningful names of process groups or comments. A development style guide may assure consistency in naming of processes, classes, variables, and customized features and provide conventions for structuring rule sets.

8.10.1.5 About Workspace Automation in Actions

An action can contain workspace automation subroutines and produce subsets, copies, or tiles as a internal activity of an action. Such actions can be executed as rule sets that include workspace automation subroutines.

→ Automate the Workspace Processing on page 295

If several actions that contain multiple workspace automation subroutines are assembled in one solution **.dax** file, each action is submitted for processing after the foregoing action with subroutines is completely processed. Otherwise a following action might search for tiles, that do not yet exist because the foregoing action is still processing.

Information kept in parameter sets are transferred between the different stages of the workspace automation. Different subroutines of different actions are able to access variables of parameter sets, that have been set in subroutines in different actions.

When creating actions you should use special **Variables Operation Algorithms** to enable the automated exchange of parameter sets among actions.

→ Reference Book

8.10.2 Create a New Action Library

Before wrapping a rule set as action definition, you have to create a new action library.

1. To create an action library choose **Library** > **New Action Library** on the main menu bar. The **Create New Action Library** dialog box opens.
2. Select a **Name** and a **Location** for the new action library. Click **OK** to create the new **.dlx** file.
3. The action library is loaded to the **Analysis Builder** window. **The Analysis Builder** window changes its name to **Edit Library:** *Name of the library*. As the editing mode is active, you can immediately start editing the action library.

8.10.3 Assemble and Edit an Action Library

When assembling a new action library, you wrap rule sets as action definitions and give them an user interface. Later, you may modify an existing action library.

Precondition: Create a new action library or alternatively load an existing action library by choosing **Library** > **Open Action Library** on the main menu bar. The name of the action library is displayed in the title bar of the **Analysis Builder** window.

→ Create a New Action Library on page 309
→ Open Action Library on page 316

Figure 266: The title bar of the Analysis Builder displays the name of the loaded action library.

1. To activate the action library editing mode, choose **Library** > **Edit Action Library** on the main menu bar. The **Analysis Builder** window changes its title bar to **Edit Library:** *Name of the Loaded Action Library*. Additionally, a check mark left of the menu command indicates the editing mode.

Figure 267: In the library editing mode, the title bar of the Analysis Builder changes to Edit Library.

2. Go to the **Analysis Builder** window and right-click any item or the background for available editing options. Depending on the right-clicked item you can add, edit, or delete one of the following:
   - General settings definition
   → Edit General Settings Definition on page 310
   - Action groups grouping actions
   → Edit Action Group on page 310
   - Action definitions including various **Export** actions
   → Edit an Action Definition on page 310
   - Widgets (user interface components) for the properties of actions
   → Edit Widgets for Action Properties on page 313
3. Do not forget to save the edited action library by choosing **Library** > **Save Action Library** on the main menu bar.
4. To close an action library, choose **Library** > **Close Action Library** on the main menu bar.
→ Close Action Library on page 316
5. To deactivate the editing mode, choose again the **Library** > **Edit Action Library** (checked) on the main menu bar. The **Edit Library:** *Name of the Loaded Action Library* window returns to its title bar **Analysis Builder**.

8.10.3.1 Edit General Settings Definition

A general settings definition enable users of action libraries to associate their image data with the appropriate actions. General settings definitions may include interface components (widgets) for defining parameters like the image size and bit depth as well as content of the image layers.

1. To create a general settings definition, go to upper pane of the **Analysis Builder** window, select and right-click any item or the background and choose **Add General Settings**. The **General Settings** definition item is added at the top of the **Analysis Builder** window.
2. For editing, select **General Settings** definition item and edit the widgets in the same way as for action definitions.

→ Edit Widgets for Action Properties on page 313

8.10.3.2 Edit Action Group

Every action is part of a certain action group. If the appropriate action group does not yet exist, you have to create it.

1. To create an action group, go to upper pane of the **Analysis Builder** window (now called **Edit Library:** *Name of the Loaded Action Library*) and right-click any item or the background and choose **Add Group**. The new action group is added at the bottom of the existing action group list.

2. To modify an action group, double-click it or right-click it and select **Edit Group**. The **Edit Group** dialog box opens.

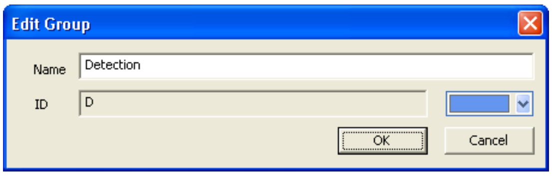


Figure 268: Edit Group dialog box.

3. Edit **Name**, **ID** and label color of the action group. After adding any action definition, the ID cannot be modified.
4. Before changing ID or deleting an action group, you have to delete all contained action definitions.
5. To move an action group, right-click it and select **Move group Up** or **Move Group Down**.
6. To delete an action group, right-click it and select **Delete Group**.

8.10.3.3 Edit an Action Definition

Action definitions are unconfigured actions enabling users of action libraries to specify actions as building blocks of a specific solution solving the user's image analysis task. Define an action definition by transforming a rule set that is related to a specified part of the solution.

Alternatively, you can import an action definition from an **.xml** file to an action library.

→ Import Action Definition from File on page 323

Precondition: You need to load a **Rule Set** file (**.dcp** file) to the **Process Tree** window containing a rule set that is related to a specified part of the solution. The rule set has to include a parameter set providing variables to be adjusted by the user of the action library.

1. To create an action definition, go to the **Analysis Builder** window, select and right-click any action group or the background and choose

   - **Add Action Definition**

   or one of the standard export action definitions:

   - **Add Export Domain Statistics**
   - **Add Export Object Data**
   - **Add Export Project Statistics**
   - **Add Export Result Image**

   The new action definition item is added at the bottom of the selected action group.

2. If you have sequenced two actions or more in an action group, you may rearrange them in between their action group by using the arrow buttons on the right of each action item.

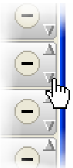


Figure 269: Arrow buttons for moving actions in between an action group.

For editing, select a action definition item, right-click it and choose **Edit Action Definition**. Alternatively, you may double-click the item. The **Action Definition** dialog box opens.

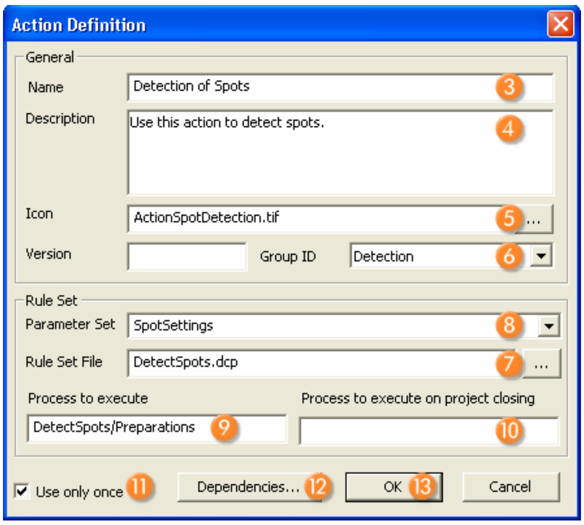


Figure 270: Action Definition dialog box is used for editing unconfigured actions.

3. Edit the **Name** of the action definition.
4. Provide a helpful **Description** about the usage of the action.
5. You can display an **Icon** on the action user interface element.
6. The **Group ID** reflects the current group the action belongs to. To move it select another group from the drop-down list box.

> **Note**
>
> Standard export actions (see step 1) are predefined. Therefore the underlying processes cannot be edited and some of the following options are unavailable.

7. Select the **Rule Set** file (**.dcp**).
8. Select the appropriate parameter set holding the related variables. The **Parameter Set** combo box offers all parameter sets listed in the **Manage Parameter Sets** dialog box.

→ About Parameter Sets on page 266

9. **Process to execute**: Enter the name and the path of the process that should be executed by the action. Denote the path using **/** (slash marks) to indicate hierarchy in the process tree. Example: **DetectSpots/Prepare** will execute the process **Prepare** which is a child process of the **DetectSpots** process group.
10. **Process to execute on project closing**: Enter the name and the path of the process that should be executed by the action when closing a project. This can be used to implement specific clean up operation that need to be executed when the users closes a projects, for example after sample input. Denote the path as mentioned in step 9.
11. Determine if an action may be used once or multiple times per solution. Clear the **Use Only Once** check box to allow multiple actions of this type in a solution.
12. Providing default actions for building solutions requires consideration of dependencies on actions. Click the **Dependencies** button to open the **Edit Action Dependencies** dialog box.

→ Edit Action Dependencies on page 312

13. Confirm with **OK**.

Edit Action Dependencies

Providing action definitions to users of action libraries requires consideration of dependencies on actions because actions often are mutually dependent on each other.

Example of a dependency: For a sub-dividing classification action of image objects of the class **Lengthy**, you need to require the class **Lengthy**.

Dependency items on actions for which dependencies can be expressed are image layers, thematic layers, image object levels, and classes.

To enable the usage of default actions for building solutions, the dependencies on actions concerning dependency items have to be defined. Dependencies can be defined as follows:

- The dependency item is **Required** for an action.
- The dependency item is **Forbidden** for an action.
- The dependency item is **Added**, created, or assigned by an action.
- The dependency item is **Removed** or unassigned by an action.

1. To edit the dependencies, go to the **Edit Action Definition** dialog box and click the **Dependencies** button. The **Edit Action Dependencies** dialog box opens.

→ Edit an Action Definition on page 310

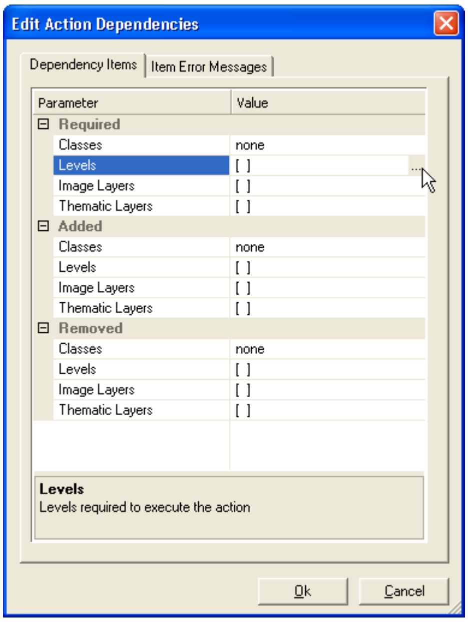


Figure 271: Edit Action Dependencies dialog box.

2. The **Dependency Item** tab gives an overview which items are **Required**, **Forbidden**, **Added**, or **Removed**. To edit the dependencies, select an item. Click the ellipsis button located inside the **Value** column. One of the following dialog boxes opens:

(ellipsis button)

   - **Edit Classification Filter** dialog box allowing you to configure **Classes** to be required, added or removed by an action.

→ Edit Classification Filter on page 443

   - **Select Levels** dialog box allowing you to configure image object **Levels** to be required, created or removed by an action.
   - **Select Image Layers** dialog box allowing you to configure dependencies concerning **Image Layers** to be required, assigned or unassigned by an action.
   - **Select Thematic Layers** dialog box allowing you to configure **Thematic Layers** to be required, assigned or unassigned by an action.

3. In the **Item Error Messages** tab you can edit messages that are displayed in the properties panel to the users of action libraries in cases where the dependency on actions cause problems. If you edit nothing here, a default error message is created: *Item* **is needed to execute the action!**

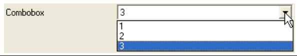
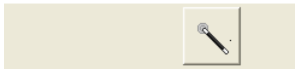
8.10.3.4 Edit Widgets for Action Properties

For each action, define the related parameters that the user of an action library can adjust. Same applies for general settings definition.

1. Select an action definition in the upper pane of the **Analysis Builder** window.
2. You have to structure the related parameters in at least one property group in the lower pane of the **Analysis Builder** window. To create a new property group here, right-click the background and select **Add Group** on the context menu. To create another property group, right-click the background again.

Add Group		Heading for grouping property widgets

3. Select a group or widget and right-click it. On the context menu, you can select the following user interface components called widgets:

Add Checkbox		
Add Drop-down List		
Add Button		Regular
		Radio button
Add Editbox		
Add Select Class		
Add Select Feature		
Add Select File		Open or save a file.
Add Select Level		
Add Select Folder		
Add Slider		
Add Layer Drop-down List		Select the number of combo boxes according to the number of layers.

4. To create a widget, choose one of the **Add** *(widget)* commands on the context menu. The **Widget Configuration** dialog box opens.

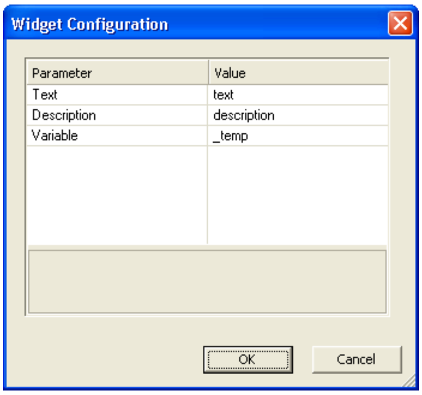


Figure 272: Widget Configuration dialog box.

5. Select a variable and, if available, configure further settings of the widget.
6. Edit a **Description** text for each widget. Only when the **Edit Action Library** mode is switched off, the **Description** text is displayed if the mouse is located over the related widget area.
7. The new widget is added at the bottom of the selected group or below the selected item.
8. Do not forget to save the edited action library by choosing **Library** > **Save Action Library** on the main menu bar.

→ Assemble and Edit an Action Library on page 309

Options

9. To move the widget within its group right-click it and choose **Move Up** or **Move Down** on the context menu.

10. To modify a widget, just double-click it or right-click it and choose **Edit** on the context menu.

11. To delete a widget, right-click it and choose **Delete** on the context menu.

8.10.3.5 Export Action Definition to File

Export an action definition to file. This can be used to extend action libraries of **Definiens Architect** users with new actions definitions.

→ Import Action Definition from File on page 323

1. To export an action definition, select an action in the **Analysis Builder** window and choose **Library > Export Action** on the main menu.

2. Select the path and click **OK**.

8.10.4 Build Analysis Solution

Test your developed actions with relevant data.

Before you can analyze your data, you must build an analysis solution in the **Analysis Builder** window.

To construct your analysis solution, you can choose from a set of predefined actions for object detection, classification and export. By testing the effect on the open project, you can configure actions to meet your needs. With the **Analysis Builder**, you assemble and configure these actions all together to form a solution. Finally, you run the solution to process your data, or save the solution to a file.

8.10.4.1 Open the Analysis Builder

Use the **Analysis Builder** window to assemble your image analysis solution.

To open the **Analysis Builder** window, do one of the following:

- Choose **View > Windows > Analysis Builder** on the main menu.

Analysis Builder

- Choose **Analysis > Analysis Builder** on the main menu.

- Choose **View > Analysis Builder View** on the main menu to select the preset layout.

Analysis Builder View

The **Analysis Builder** window opens. Make sure that the name of the desired action library is displayed in the title bar of the **Analysis Builder** window. If not, change the action library.

→ Change the Action Library on page 316

Figure 273: The title bar of the Analysis Builder displays the name of the loaded action library.

The **Analysis Builder** window consists of two panes: In the upper sequencing pane, you assemble actions to build solutions. In the lower properties pane you can configure them by customizing specific settings.

→ Assemble a Solution from Actions in the Analysis Builder on page 317

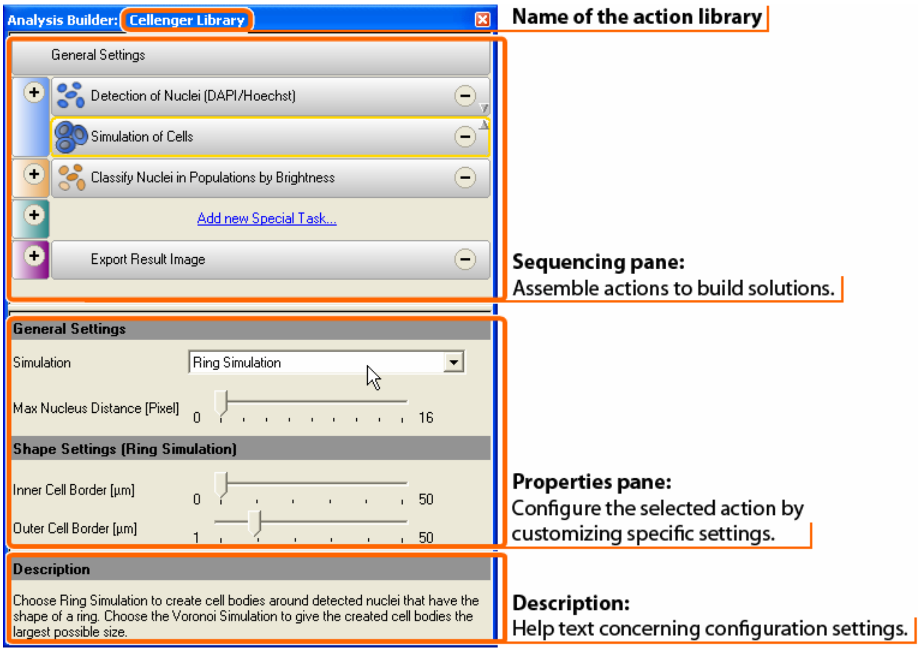


Figure 274: Analysis Builder window with sample actions from Cellenger action library.

Depending on the selected action, the lower properties pane shows the associated settings to define. The **Description** area displays help text to assist you with the configuration.

→ Set Properties of an Action on page 319

8.10.4.2 Change the Action Library

You can close the current action library and open another to get access to another collection of analysis actions.

Close Action Library

To close the currently open action library, choose **Library > Close Action Library** on the main menu bar.

The action groups in the upper pane of the **Analysis Builder** window disappear.

> **Note**
>
> When closing an action library with an assembled solution, the solution is removed from the upper pane of the **Analysis Builder** window. If not saved, it has to be reassembled.

Open Action Library

To open an existing action library, choose **Library > Open Action Library** on the main menu bar. Browse for a folder containing a desired action library and confirm with **OK**. The name of the loaded action library is displayed in the title bar of the **Analysis Builder** window.

Figure 275: The title bar of the Analysis Builder displays the name of the loaded action library.

The action groups of the library are loaded in the upper pane of the **Analysis Builder** window.

Caution

If you open an action library after opening a project, all rule set data will be deleted. A warning message will display. To restore the rule set data, close the project without saving changes, and then reopen it.

Note

Legacy Solutions

If you are using a solution built with a older action library, browse to that folder and open that action library before opening your solution.

8.10.4.3 Assemble a Solution from Actions in the Analysis Builder

In the **Analysis Builder** window, you assemble a solution from actions and configure them in order to analyze your data.

1. If not visible, open the **Analysis Builder** window.

→ Open the Analysis Builder on page 315

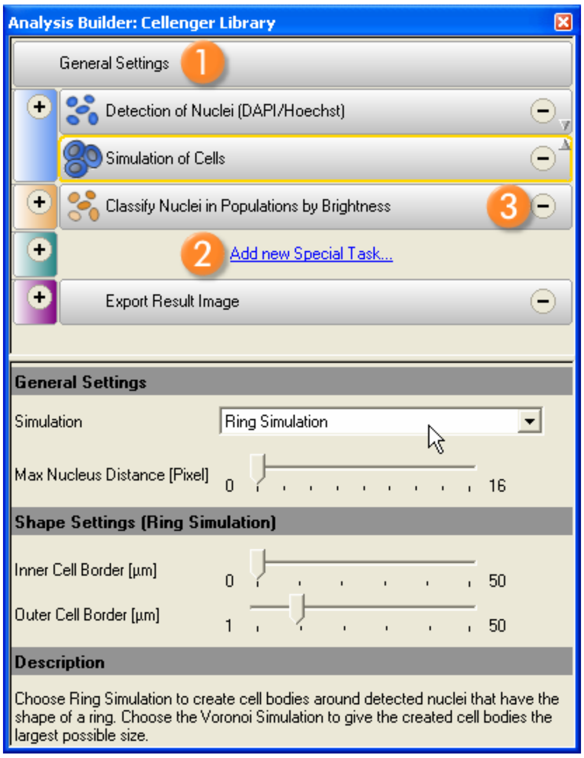


Figure 276: Analysis Builder window displaying sample data of an application library.

The **Analysis Builder** contains a button for the **General Settings** 1 and groups in different colors for different kind of actions: detection, classification, export and special tasks.

→ Set General Settings of a Solution on page 319

Special tasks are customized actions exclusively designed for your special needs.

→ Import Action Definition from File on page 323

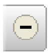
2. To add an action, click the button with a plus(+) sign on the sub-section header or, in an empty section click **Add new ...** 2. The **Add Action** dialog box opens.

Add Action

→ Select an Action on page 318

3. Select an action from the **Add Action** dialog box and click **OK**. The new action is added to the solution. According to the type of the action, it is sorted in the corresponding group. The order of the actions is defined by the system and cannot be changed.

4. To remove an action from your solution, click the button with a minus (–) sign on the right of the action bar 3.



Remove Action

5. Icons inform you about the state of each action:

→ Set Properties of an Action on page 319

   - The red error triangle indicates that you must specify this action before it can be processed or another action must be processed before.

   - The green check indicates that the action has been processed successfully.

Select an Action

Select an action for the analysis solution of your data.

1. In the **Analysis Builder** window, click on the plus sign on the left of the analysis sections. The **Add Action** dialog box opens.

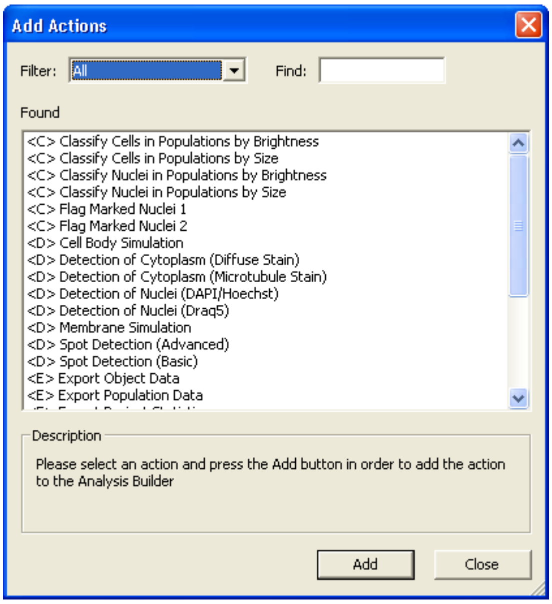


Figure 277: Add Action dialog box with sample actions.

2. The filter is set for the action subset you selected. You can select a different filter or display all available actions. The **Found** area displays only those actions that satisfy the filter setting criteria. Depending on the action library, each action is classified with a token for its subsection, for example, **<C>** for classifications, **<D>** for detections, **<E>** for export actions and **<ST>** for special tasks.

3. To search for a specific action, enter the name or a part of the name in the **Find** text box. The **Found** area displays only those actions that contain the characters you entered.

4. Select the desired action and confirm with **OK**. The new action is displayed as a bar in the **Analysis Builder** window. You must now set the properties of the action.

> **Note**
>
> Some actions can be selected only once. If such an action is already part of the analysis, it does not appear in the **Add Action** dialog box anymore.

Set Properties of an Action

After selecting an action, you must configure its settings. Since each action has its own properties, the options vary accordingly.

1. Go to the **Analysis Builder** window.
2. In the upper sequencing pane, select the action you want to customize.
3. Configure the action in the lower properties pane by customizing various settings.

→ **Open the Analysis Builder** on page 315

Set General Settings of a Solution

For each solution you must define the general settings. These settings associate your image data with the appropriate actions.

General settings include the pixel size and bit depth. Alternatively, you can load a calibration file that corresponds to the settings of your image reader, camera, or instrument.

With general settings, you further set the content of the image layers.

Precondition: Open the workspace with the data you want to analyze and open a project that works as a good example for the whole workspace.

1. To define the general settings, open the **Analysis Builder** window and select the **General Settings** button. In the lower properties pane of the **Analysis Builder** window you can do the settings.

→ **Open the Analysis Builder** on page 315

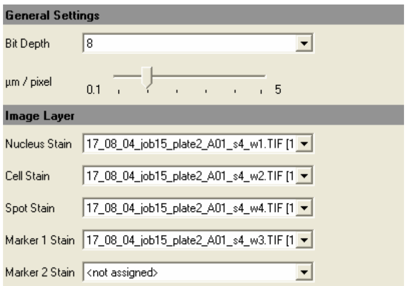


Figure 278: Sample properties of the General Settings of a solution.

General Settings

You can define the bit depth and the pixel size manually or by loading a calibration from file.

2. Select one of the following **Bit Depth** for your project:

→ **Load Calibration Corresponding to Image Reader Settings** on page 320

- **8** Bit images have a gray value range from 0 to 255
- **12** Bit images have a gray value range from 0 to 4095
- **16** Bit images have a gray value range from 0 to 65535

> **Note**
>
> Although some image formats are saved as 16 bit, they don't use the full range of 65536 gray values. In case you are using image data whose gray value range might have values of approximately 8000 we generally recommend using the 12 bit setting.

3. Select the pixel size in µm.

Load Calibration Corresponding to Image Reader Settings

Instead of manually defining the **General Settings** of a solution, you may load a calibration from file corresponding to an image reader and its image recording settings.

Loading a calibration sets the following **General Settings** properties:

- **Bit Depth**
- Pixel resolution in **µm/pixel**
- The scheme of **Image Layers** used for image analysis.


Load Calibration

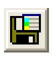
1. To load a calibration choose **Library > Load Calibration** on the main menu bar. The **Load Calibration Parameter Set** dialog box opens.
2. Browse for a Definiens **Parameter Set** file with the extension **.psf**. Common calibrations are stored in the **Calibration** folder within the corresponding application folder. The default path is:
C:\Program Files*Definiens Developer Version number***\bin \Applications\Cellenger\Calibration**.
3. Confirm by clicking the **Open** button. You can check the loaded **General Settings** properties in the lower properties pane of the **Analysis Builder** window.

> **Note**
>
> Loading a calibration deletes the currently displayed general settings and substitutes the loaded calibrated general settings.

→ Create Calibration Parameter Set File on page 323

Save Analysis Settings as Solution

You can save the analysis settings in the **Analysis Builder** as a solution file (extension **.dax**) and load them again.


Save Solution to a File

1. To save the analysis settings, do one of the following:
    - Click the **Save Solution to a File** button on the **Architect** toolbar.
    - Choose **Library > Save Solution** on the main menu bar.

    Alternatively, you can encrypt the solution by using one of the following:

- Click the **Save Solution Read-only** button on the **Architect** toolbar.
- Choose **Library > Save Solution Read-only** on the main menu bar.

Save Solution Read-only

2. Enter a file name and a directory location and click **OK**.

Load Solution

Load an already existing solution with all analysis settings from a solution file (extension **.dax**) to the **Analysis Builder** window.

> **Note**
>
> **Legacy Solutions**
>
> If you want to change a solution built with an older Definiens action library, make sure that the according action library is open before loading the solution.
>
> → **Open Action Library** on page 316
>
> When opening a solution file (extension **.dax**), the contained actions are compared with those of the current Action Library. If in the current Action Library already exists an action of the same name as in the solution file, the one of the current Action Library is loaded to the analysis builder. This does not apply when using a solution file for automated image analysis.

1. To load a solution, do one of the following:

→ **Change the Action Library** on page 316

- Click the **Load a Solution From File** button on the **Architect** toolbar.
- Choose **Library > Load Solution** on the main menu bar.
- To use a Solution that was built with another Action Library, open the Action Library before opening your solution.

Load a Solution From File

2. Select the desired file and click **OK**. The solution is displayed in the **Analysis Builder** window.

Test and Improve Solution

Before an analysis solution completely satisfies your needs, you must test and improve it on an open project or on a subset. This cycle might take some time. Here are some tips to help you to improve the results:

- Use the **Quick Test** some actions provide to instantly display the results of a certain setting in the **Project View**.
- To execute all assembled actions, click the **Run Solution** button on the **Architect** toolbar. Alternatively, choose **Analysis > Run Solution** on the main menu bar.
- To execute all actions up to a certain stop, select an action and click the **Run Solution Until Selected Action** button on the **Architect** toolbar. Alternatively, choose **Analysis > Run Solution Until Selected Action** on the main menu bar. All actions above and including the selected action will be executed.
- To improve the test processing time, you can test the actions on a subset of your project data.

Run Solution

Run Solution Until Selected Action

- Execute selected actions. If you use many actions, a test run might take a lot of time. For faster testing use the **Run Selected Action** button on the **Architect** toolbar. Alternatively, you can remove already tested actions. You can simply delete them from the **Analysis Builder** window and add them later again. The settings remain until changed. You can also save the actions and the settings as solution to a **.dax** file. When removing single actions you must make sure that the analysis job remains complete. For instance, you must first detect nuclei before you can classify them, or you must first detect cells or simulate cell bodies before you can classify cells.

Run Selected Action

→ Execute Solution Manually for Testing on page 362

- To execute a configured solution not locally but on the **Definiens eCognition Sever**, select a project in the workspace window. Click the **Run Solution on eCognition Server** button on the **Architect** toolbar. This option is needed if the solution contains actions with workspace automation algorithms.

Run Solution on eCognition Server

View a Subset for Configuration

When configuring actions that are intended to process scene subsets or tiles only, you can open a subset of the project and using it for configuration.

1. Go to the right pane of the **Workspace** window that lists all projects of a workspace.
2. Right-click a project and choose **Open Subset**. The **Subset Selection** dialog box opens.

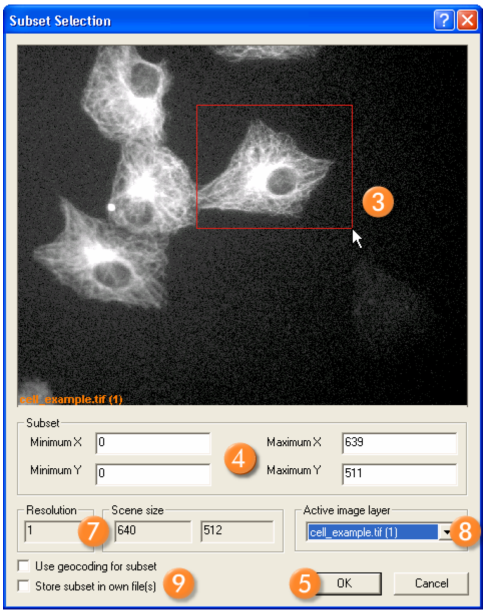


Figure 279: Subset Selection dialog box.

3. Click in the image 3 and drag to select a subset area.
4. Alternatively, you may enter the subset coordinates 4. You can modify the coordinates by typing.
5. Confirm with **OK** 5.

   The subset displays in the project view. This subset is not saved with the project and does not modify the project. After closing the project view of the subset, the subset is lost.

   However, you can save the subset as a separate project.

→ Save a Project on page 55

Import Action Definition from File

To get access to new, customized, or special actions, you have to load action definitions, which just are unconfigured actions. If not yet available in the **Add Actions** dialog box, you can load additional action definitions from file to an action library.

This can be used to update action libraries by externally defined action definitions.

Action definitions can be created with the **Definiens Developer**. Alternatively, Definiens offers consulting services to improve your analysis solutions. You can order special task actions for your individual image analysis needs.

→ Export Action Definition to File on page 315

To use an additional action definition you have import it.

Precondition: Beside the **.xml** file describing the action definition, you need a **Rule Set** file (**.dcp** file) providing a rule set that is related to a specific part of the solution. The rule set has to include a parameter set providing variables to be adjusted by the user of the action library.

1. Copy the action definition files to the system folder of your installation.
2. Choose **Library** > **Import Action** on the main menu. Select an **.xml** file to load.
3. Now the new unconfigured action can be selected in the **Add Actions** dialog box.

8.10.5 Create Calibration Parameter Set File

You can create a calibration to store the **General Settings** properties as Definiens **Parameter Set** file. Thus you can save and provide common settings for common image readers, for example, as part of an application.

→ Load Calibration Corresponding to Image Reader Settings on page 320

A calibration set stores the following **General Settings** properties:

- **Bit Depth**
- Pixel resolution in **µm/pixel**
- Zero-based IDs of the physical image layers within the scene. Example: If a scene consists of three physical image layers, the first image layer will have ID 0, second layer - ID 1 and third layer - ID 2. Thus you can store the scheme of used **Image Layers** for image analysis.

1. To create a calibration, set the **General Settings** properties in the lower properties pane of the **Analysis Builder** window.
2. Choose **Library** > **Save Calibration** on the main menu bar. By default, calibration parameter set files with the extension **.psf** are stored in the **Calibration** folder within the corresponding application folder. The default path is:
   C:\Program Files*Definiens Developer Version number***\bin**
   \Applications\Cellenger\Calibration.
   You may want to browse to another folder before clicking the **Save** button and creating the calibration parameter set file.

9 About Development Strategies

This chapter focuses on composing rule sets based on the **Definiens** Cognition Network Language (CNL). You can learn about the extended workflow of rule set development including the strategic approach. This chapter enables you to develop rule sets that suit your need for both insight and efficiency and supports you in handling common problems.

→ **How to Approach Developing a Rule Set** on page 324
→ **Generic Development Strategies** on page 337
→ **Find and Handle Image Structures** on page 351

This chapters is the third of three chapters about rule set development.

→ **Three Chapters About Rule Set Development** on page 123
→ **How to Learn Developing Rule Sets** on page 121

9.1 How to Approach Developing a Rule Set

This section outlines how to approach a strategy before you start editing processes. You learn to ask important questions and make the right choices during the development process.

A typical workflow of rule set development looks like this:

	Workflow Step	Substep
1.	Understand the task.	
		Formulate regions of interest.
2.	Explore the image.	
		Make assumptions concerning the strategy. Consider the available data. Estimate the image variability.
3.	Design the rule set.	
		Plan the image object hierarchy. Plan the class hierarchy. Define the modules of the rule set.
4.	Set up an implementation sequence.	
		Consider the dependencies: ▸ If needed, go back to step 3.

→ **Understand the Task** on page 325
→ **Explore the Images** on page 326
→ **Design the Rule Set Architecture** on page 329
→ **Set Up an Implementation Sequence** on page 332

5.	Implement modules of the rule set.	
		Generate image objects. Identify and create features. Formulate and verify critical assumptions. Try a first detection. Compose processes and process sequence. Qualify remaining problems to submodules.
6.	Test the implementation.	
		Evaluate improvement strategies: ▸ Restart with step 1. ▸ Optimize the approach and go back to step 2. ▸ Optimize the implementation and go back to step 5. ▸ Finished.

→ Implement Modules of the Rule Set on page 333

→ Test the Implementation on page 336

9.1.1 Understand the Task

It is very critical to understand what the desired outcome of a specific image analysis task is:

- Which different regions of interest (ROI) you need to identify?
- What type of information should be extracted from the image?
- What is the question that is to be answered?

Learn to Visually Identify The Regions Of Interest (ROI)

Make sure that you can reliably identify the different regions of interests yourself. This may require you to consult an domain expert, as they often rely on rather subtle cues that are difficult to communicate. However, if you start with the wrong conception of the regions of interest, the whole project is bound to fail. It is quite common that new questions arise while you progress on the development, because your way of looking at the images changes with increasing experience.

Understand the Scientific Question

It is also important that you are aware of the purpose of the study, what the question is and what conclusions will be drawn. This is necessary to avoid bias when making assumptions.

→ Formulate and Verify Critical Assumptions on page 334

Define Acceptance Criteria

Having an acceptance criteria helps to approach the image analysis task and to develop rule sets efficiently. Simple acceptance criteria may base on manual counting of regions or objects of interest. If you are not sure about such criteria we recommend to contact an domain expert.

9.1.1.1 Formulate Regions of Interest

When starting a development approach, first look at your image data. Based on your image analysis task, define regions of interest (ROI) within the images.

To approach a good strategy, first formulate how you see the **Region of Interest (ROI)** with your eyes without using technical terms. Formulating the **Region of Interest** is a very crucial step in rule set development, since your strategy will depend on it.

Tip

Use Simple Words to Describe the Region of Interest

*Use simple words like **blue ellipse** to describe what you see instead of predefined terms like **nucleus** or **tree**. This allows you to be more precise and easily translate this strategy into code.*

9.1.2 Explore the Images

When developing rule sets for image analysis, we recommend starting with a visual investigation of sample images. Knowing the properties of the full range of image material in as much detail as possible is a prerequisite for coming up with an efficient design for your rule set.

The goal is to recognize the complex visual information which is contained within an image. This enables you to categorize helpful from useless image information and to define your development strategy. It can be useful at this stage to execute selected segmentations; that way you can look at how well different features separate your regions of interest.

At the end of this stage, you should have a good understanding of how difficult different aspects of the image analysis task are. Some regions of interest may be easily separable by features, and perhaps there are strong assumptions which are transferable from image to image and make identification easy. Other aspects of your task are likely to be more challenging, as differences between regions of interest may be subtle and complicated to express.

→ **Consider the Available Data** on page 326
→ **Estimate Image Variability** on page 329
→ **Make Assumptions Concerning the Strategy** on page 327

9.1.2.1 Consider the Available Data

Before actually starting to develop a rule set, you need to define the data source on which to develop. In most cases the data source is defined by the application, which means you basically have to use what is available. In other cases, there is a certain freedom with regard to which data will be used. In such cases you should consider the following points:

Use Representative Image Data

Make sure the data set used to develop a rule set is representative of the data to be analyzed in an operational mode later on. Include treated and control data in Life Sciences whereas in Earth sciences you might include different times of day and weather conditions. If development is done on superoptimal data, the results in a later operational mode will be suboptimal. If the data used to develop rule sets is substandard, extraction potential is wasted.

Include Additional Data

Look into what data is available and think about possible use. Generally, the more data is included in the analysis, the better the results that can be achieved. If auxiliary information like thematic data or metadata is available reliably, it is worth while to consider including it. For example, rule sets can be made more robust by including meta data providing information on the data acquisition circumstances like staining, magnification, flight elevation, sun angle and others.

Define Image Acquisition Parameters

In an ideal setting, where there is a opportunity to influence the data acquisition, you should try and define acquisition parameters to best suit your analysis task.

9.1.2.2 Make Assumptions Concerning the Strategy

When writing a rule set on only a few images or of a constant environment, you can use fixed thresholds, for example, **Area < 1000** or **Mean Layer 1 > 153**. Underlying this strategy is your assumption, that the objects you want to find are always smaller than **1000**. On a small image set, this is easily verified and knowing your assumption is correct, you can safely use these thresholds.

Therefore, it is important to realize that whenever you write a rule set, you make assumptions and based on those assumptions you set up a strategy. Your strategy and the resulting accuracy of your analysis are only as good as the assumptions you have made.

Sample Approach

1. Look at the image below. Imagine, the task is to find the blue rectangle as annotated by the arrow.

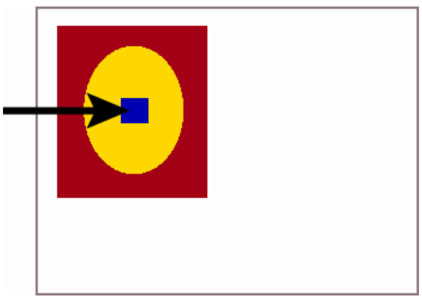


Figure 280: Image 1

   Seeing only this first image you might set up the strategy: **"Find all blue areas in the image"**.

2. But seeing the next images, you will realize that your first strategy needs adjustment: There are more blue rectangles in the images, so first you need to know which ones to find and second, how to differentiate between those.

Let us say you need only find the blue ones indicated by arrows. Have a look at all three images and try to formulate your strategy, so that it fits for all images.

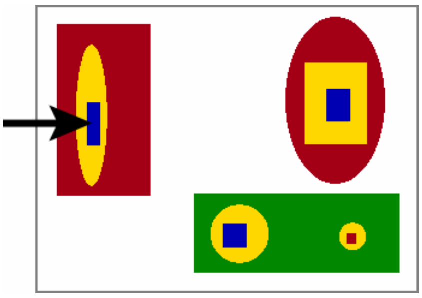


Figure 281: Image 2

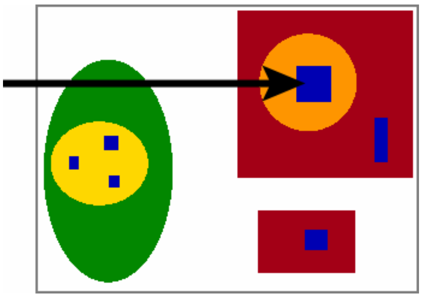


Figure 282: Image 3

Your final approach could be: **"Find blue rectangles in yellow or orange elliptic area in red rectangles."**

3. For here you can then easily extract the single steps you have to do to find the **Region of Interest (ROI)**:

   - 3.1 Find red area.
   - 3.2 Filter for the rectangles.
   - 3.3 Find yellow or orange area inside red.
   - 3.4 Filter the elliptic ones.
   - 3.5 Find blue rectangle in yellowish (yellow or orange) areas.

   This is the strategy you can use for analyzing the images.

Now you can also clearly see the assumptions you have made for image analysis:
If you had seen only the first two images, you might assume, the blue rectangle is inside a bright yellow ellipse, but seeing the third image you had to adapt this, since here the yellow is actually orange. In an even larger image set, it might happen that the red rectangle, for example, has a darker color and your assumption will not be valid anymore. Obviously, you cannot adapt the rule set to this situation without seeing the images.

That is why it is so important to have lots of images representing all types of situations occurring in the study.

→ Formulate and Verify Critical Assumptions on page 334

For large image sets or complex tasks you might need to use **and** or **or** operators in your formulated strategy steps to cover all situations. You might also draw a flowchart to visualize your strategy and simplify the check of assumptions.

9.1.2.3 Estimate Image Variability

A critical step in robust rule set development is to be aware of the full range of data variability within your workspace.

One easily falls into the trap of spending days on achieving a perfect classification on a couple of images only to realize that the assumptions on which the rule set was based do not hold for all images. Consequently, you need to start again from scratch.

For estimating the image variability, it is good practice at this step to:

- Look at all the images; that means perform a visual inspection:
  Try to formulate some basic assumptions and see whether they hold over the whole workspace. For example, the background may cover about 50% in almost all images, but there maybe a handful where this is not the case and this is something you want to be aware of.
  ••If you cannot look at all images because there are too many, try to select the images you do look at **randomly** as this gives you a better chance of capturing the whole variability.
  ••If necessary, write a rule set that exports all the images (or image tiles) in a format that can be easily inspected.

- Export some basic image statistics:
  Create a rule set that consists of a simple segmentation and a statistical export, for example **max pixel value**, **min pixel value**, **mean pixel value**. This will give you additional information about the variability you must handle.

Develop Robust Rule Sets for Multiple Varying Images

For a robust rule set that is usable in highly automated image analysis environments you may need up to several hundred images representing the diversity of the final image set.

The first thing to do is to get an overview over the diversity of the images and their quality. In life sciences, the staining intensity might change among the images as well as the focus or the images are even from different sources. Similarly, in earth sciences there is sometimes great variability due to changes such as sun angle, cloud cover, vegetation intensity and much many others.

Avoid developing with a small subset of the images, you might make assumptions which are only valid for the subset but not for the total image set and thereby your strategy might not work for all images. This is more like to be true of images with poor quality.

At the same time, you might need to consciously limit the variability that is supported by one rule set. In other words, it is often more practical to develop several rule sets, each designed for particular conditions, than to write one rule set that covers a large range of variability.

9.1.3 Design the Rule Set Architecture

Design the architecture of the rule set including the image object hierarchy and the modules of the rule set.

In doing so, you need to clarify the implications for the design of your class hierarchy, where commonly you aim for a class corresponding to each region of interest.

If you also need to access structures at various scales, such as tissues, cells, and cell components, or forests, tree groups and single trees, you may consider those scales in the planned image object hierarchy.

→ Use Cases: Multi-Scale 1–3 on page 297

Tip

Recheck the Image Analysis Task

While designing the rule set architecture, you should reexamine the image analysis task to make sure that your rule set meets all requirements.

→ ***Understand the Task*** *on page 325*

9.1.3.1 Plan the Image Object Hierarchy

Plan the image object hierarchy according to different scales of image objects you want to use for extracting information.

A crucial hint about how many image object levels you need is the number of different image objects to which a pixel belongs. In a cell image, for example, a pixel that is part of a nucleus should belong to an image object of class **Nucleus**. At the same time, it should belong to a (bigger) image object of class **Cell**. Thus, two image object levels are needed as a minimum if you want to describe a cell and its nucleus. In an aerial image, a pixel might have to belong to an image object of class **City** as well as to a smaller image object of class **House**. Again, two image object levels are needed to describe both.

Sometimes it is useful to add temporary image object levels which are deleted later. For example, you might need them when you want to do merging operations on one level, but want to keep the smaller image objects created in an earlier segmentation procedure for later use.

9.1.3.2 Plan the Class Hierarchy

Plan the class hierarchy according to the regions of interest.

When you know what kind of image objects you want to extract, it should be straight-forward to define some of the classes that you need. You may also group them in a class hierarchy. For example, in cellular biology you may have to find nuclei, but you find it easier to find round nuclei and lengthy nuclei separately, using different strategies. It makes sense then, to have two classes, for example, **RoundNucleus** and **LengthyNucleus**, plus a parent class **Nucleus**. Hierarchical ordering of classes allows you both to address each nucleus type separately or all nucleus types at once.

In addition to the classes describing the final image objects of interest, you might also need some classes that are necessary along the way between the initial segmentation and the final result. It is useful to give those temporary classes similar names and group the under a parent class to simplify maintenance. For example, you can unclassify all temporary classes with a simple process that only addresses the parent class.

The class hierarchy may also contain classes that are never used to classify an image object, but are used only for the evaluation of its class description via the respective class-related feature. We recommend that you group all classes with that function under a common parent class.

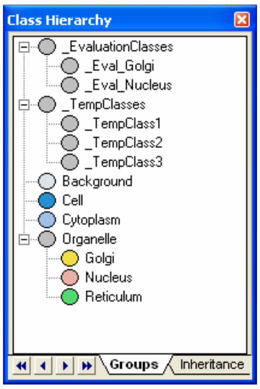


Figure 283: Example of a class hierarchy with grouped regular, evaluation, and temporary classes. For a better overview, we recommend that the names of temporary and helper classes start with an _underscore.

9.1.3.3 Define Modules of the Rule Set

Modularization refers to the practice of dividing a task into several tasks that can be solved largely independently from each other. Each sub-task is carried out by a distinct module.

You may benefit in the following ways:

- Code is easier to understand and to debug:
  For instance, if in a project you find that nuclei are not well separated, you will know immediately that you have to work on the code section **find nuclei**, subsection **separate touching nuclei**.

- Code reusability is improved:
  For instance, a basic nucleus finder module can be reused in a variety of assays. This is the underlying concept of the **Definiens Architect**.

- Image analysis tasks can be split among different developers:
  However, it is difficult to truly parallelize an image analysis task, because the development of downstream modules is difficult while upstream modules are not at least in a working draft state.

Commonly, modules focus on regions of interest (ROI). For example:

- **Find Background**
- **Find Regions of Interest**
- **Find Obvious Objects**
- **Find Analysis Targets**

9.1.4 Set Up an Implementation Sequence

After designing the rule set and the image object hierarchy, you set up an implementation sequence. This enables you to structure and schedule the realization.

From Easy to Difficult

A good rule of thumb is to start a project by classifying the obvious image structures, such as the image background. Then you proceed to increasingly difficult structures. Even if the easy structures are not of interest in the context of the study, this usually is reasonable because:

- Easy image structures can simply be ignored during subsequent processing, which saves processing time.
- You can use the initial classifications to provide context information that helps for subsequent classifications. Examples:
  You are interested in an epithelium structure that is adjacent to the background.
  You know that the objects of your interest are just slightly brighter than the background.

Separate the Modules

The order in which image structures are to be classified then forms the basis for modularization. The goal is to have modules which operate largely independently of each other.

→ Define Modules of the Rule Set on page 331

To ensure a clean separation of your module from other modules, consider implementing it as a customized algorithm or an action for the **Definiens Architect**.

Consider Dependencies Among Modules

Sequencing the modules has limitations because of dependencies among modules. If you cannot resolve dependencies you might think about redesigning the rule set.

→ Design the Rule Set Architecture on page 329

- A strict modularization is often not optimal from a performance (processing time) point of view, because steps have to be repeated in different modules. This is true especially for segmentation and merging algorithms.
- Sometimes it is difficult to classify a structure sufficiently well in the first attempt. Perhaps a structure A cannot be classified well without any information on a structure B. But at the same time, the structure B cannot be classified well without any information on the structure A.
  Here, a common solution is to alternate modules, for example:
  1. Find initial A
  2. Find initial B (using information on already classified A)
  3. Find improved A (using information on B) and so on.
  Taken to the extreme, these are not isolated modules anymore, but will represent a single module **Find structures A and B**.

Consider the Performance

In most cases, the modules will have to be implemented in the order in which they will be executed. One should aim for a high level of performance of one module before moving on to implementing the next module, because:

- A mediocre initial classification generally makes later classifications more difficult.
- A slow module slows down development as it has to be evaluated every single time a downstream module is tested.
- Changing the modules later will normally require adjustment of downstream code. Thus previous adjustment and fine tuning work was wasted.

At the same time, keep in mind the following:

- The whole strategy may be a dead-end street, in which case all work put into perfecting the initial modules was wasted.
- If later changes of the early modules have a huge effect on downstream modules it can be a valid indicator that the code is not robust enough.

In any event, it is a good habit to spend some time on cleaning up the code and trying to improve processing speed and code readability before moving on to the next module.

9.1.5 Implement Modules of the Rule Set

A single rule set module may be implemented by following a five-step pattern outlined here. Of course, this is just a hint about how to approach the coding of a module. Typically, experienced developers come up their own coding dialect, but especially beginners might profit from having a guideline.

→ About the Development Workflow on page 129

The implementation of each rule set module usually follows a basic pattern:

1. Generate image objects.
2. Identify features that enable you to distinguish objects and regions of interest.
3. Formulate and verify critical assumptions.
4. Try a first detection to classify as many image objects of your regions of interest (ROI) as possible.
5. Create processes and assemble process sequences to rule sets.
6. Qualify the remaining problems to submodules and fix them separately.

9.1.5.1 Generate Image Objects

Image objects must be generated before they can be classified.

This is done using the various available segmentation algorithms. Depending on the requirements of your task, the segmentation is either focused on performance or on extraction quality. Optimizing both is the difficult part in creating image objects.

→ Create Image Objects by Segmentation on page 157

9.1.5.2 Identify and Create Features

The first step in implementing modules such as isolating or segmenting a specific structure is to ask what features can be used to distinguish your regions of interest.

→ Get Information on Image Objects on page 167

The main tool here is the **Feature View** window. The goal is to find and/or to create features that separate different image object classes. Ideally, you will find a feature that make one region of interest (ROI) blue, and the other ROI green, with little overlap.

→ Compare Feature Values by Using the Feature View on page 179

Proceed as follows:

- As a starting point, consider a fine chessboard segmentation to see how well the different **Layer Values** features separate your regions of interest.
- Try out all available features offered by **Definiens Developer** and see which will best do the job. Possibly you could combine some of the better ones in customized features.
- Try to find out how you yourself can distinguish the regions and then consider how those visual qualities can be implemented in processes.
- Try to combine features a in class description. You can think of the classification value of this class as a quasi-customized feature. This is an extremely powerful method that often also significantly enhances code readability.
- Think, think, think ...
- Consider exporting feature values over the whole workspace, to get a feeling for image-to-image variability.

9.1.5.3 Formulate and Verify Critical Assumptions

Every strategy you come up with reflects underlying assumptions about the properties of the images you are analyzing. Formulate and verify critical assumptions to avoid the failure of rule sets.

→ Make Assumptions Concerning the Strategy on page 327

Rules of thumb:

- **Formulate** major assumptions as explicitly as possible.
- **Verify** critical assumptions on all images if they lead to critical design choices.

Do Not Make Assumptions That May Bias the Study

Scientific studies rely on assumptions and lead to conclusions. We highly recommend that you not conclude something from a study that has been assumed when the experiment was designed. While this may sound trivial, mistakes can easily occur.

A simple example: in breast cancer, investigators often want to know the amount of brown staining in a tumor region. However, brown staining is also the most obvious feature to identify a tissue as tumor tissue. If your rule set uses brownness for finding the tumor, it will be more likely to find brown stained tumor regions than non-stained regions. Therefore, your rule set will consistently overestimate the amount of brown staining in the tumor tissue. Such a bias may lead to the wrong treatment choice, possibly fatal for the patient.

In the context of rule set development, it is therefore critical that you, the developer, understand the nature of the conclusions that should be drawn from the study to enable you avoid inadvertently making assumptions about them. Likewise, you may inform the researcher about the major assumptions you make to avoid drawing any conclusions based on them.

→ Understand the Task on page 325

9.1.5.4 Try a First Detection

Try a first detection to classify as many image objects of your regions of interest (ROI) as possible with a focus on the easy ones. This will limit the number of remaining problems to be treated in subsequent modules.

Approach to a first detection as follows:

1. Decide whether you better start with a

- conservative detection, with the consequence that identified ROI image objects are correct, but many are overlooked that have to be found later.

or a

- liberal detection, with the consequence that all ROI image objects are found, but there are lots of false positives that have to be removed later.

or a

- minimum-error detection, with the consequence that the total number of incorrectly identified image objects, false positives and false negatives, is as small as possible.

2. Choose a segmentation.
3. Identify useful features and create a class description to distinguish your ROI from other image objects.
4. Classify your image objects using the classification value of this class description.

→ About the Interplay between Segmentation and Classification on page 128

9.1.5.5 Compose Processes and Process Sequence

Composing processes and sequencing them to rule sets represent the editing activities of rule set development.

The environment used for developing are the **Process Tree** window and the **Edit Process** dialog box. In addition there are several tools which support rule set development.

→ Use Processes to Build Rule Sets on page 130

The **Reference Book** supports you with details about available tools, algorithms and features.

→ Reference Book

The typical approach when starting a new rule set can be summarized as follows:

- Add parent process to function as a group for a set of processes dedicated to one analysis module.
- Define a name reflecting the task which is solved by the process group.
- Add child processes defining the actual task.
- Use the commenting tool to annotate processes of the rule set.

Some hints on rule set development:

- Try to avoid unstructured process sequences.
- Introduce and stick to coding conventions, for example on naming of classes, variables, or image object levels. This is very helpful especially when developing in a group.

9.1.5.6 Qualify Remaining Problems to Submodules

After solving most of the problems of a module implementation in a few lines of code, qualify remaining problems to submodules and fix them separately. This organizes the rule set development in a structured way.

A good rule of thumb is that no module should have more than 10 lines of code. You might have a submodule for example called **clean-up** at the end of each module, and one more module **clean-up** at the bottom of the rule set.

9.1.6 Test the Implementation

Evaluate the quality of your result after each major step in the development.

→ Examine Results on page 379

Visual Evaluation of Classification Quality

A good way to estimate classification quality is to export classification images and view them. As this can be a time-consuming process, you may put some effort into optimizing your visualization, so that you can see the critical features more easily.

→ Export Data on page 401

To get a quick overview, you may want to export result images to different folders, depending on the setting. For example, images with uninteresting results could be identified and exported to one folder, and images with interesting content to another folder. This speeds up evaluation because you do not have to repeatedly look at uninteresting images.

If there are very many images, one may do initial tests only on a randomly chosen subset of images. However, it is worthwhile to test all images from time to time, and certainly before completing a module.

Statistical Comparison of Manual and Automatic Classification

In some cases, results of manual classifications are available. In those cases, quality of classification can also be estimated by comparing the results of automated and manual analysis.

9.1.6.1 Evaluate Improvement Strategies

Before improving the detection of your region of interest (ROI) image objects, evaluate the consequences of the following improvement strategies.

- Optimize the approach and reinvestigate the image.

→ Explore the Images on page 326

- Optimize the strategy and change the rule set architecture.

→ Design the Rule Set Architecture on page 329

- Optimize the implementation and adjust parameters

→ Implement Modules of the Rule Set on page 333

To choose among these alternatives, consider the following questions:

- How much time have I already spent on optimizing my thresholds? How close am I to the local optimum?
  At some point it is wise to assume that more adjustments do not result in improvements, and maybe there is a problem with the strategy.
- Do the test results indicate some problem with the underlying assumptions?
  If so, you should consider changing your strategy.
- How many different problems do I still have to treat?
  If there are more than two or three, it is probably better to change the strategy to get rid off a few, rather then to create many submodules what commonly increases the code complexity dramatically.

9.2 Generic Development Strategies

This section supports you in finding answers and strategies for typical problems in image analysis mainly on a more general level.

→ **When to Use Which Segmentation?** on page 337
→ **Identify Features that Help Solving Your Task** on page 341
→ **How to Handle Large Data Sets** on page 346
→ **About Robustness of Rule Sets** on page 344
→ **Improve the Performance** on page 348

9.2.1 When to Use Which Segmentation?

This section provides strategic information about segmentations. Each segmentation strategy has different typical application fields.

→ **Create Image Objects by Segmentation** on page 157
→ **Segmentation Principles: Top-down and Bottom-up** on page 159

Further you can learn about advanced segmentation strategies.

→ **Advanced Segmentation Methods** on page 338

9.2.1.1 Chessboard Segmentation Application Fields

Because the chessboard segmentation algorithm produces simple square objects, its main usage focuses on the cutting of images and image objects.

The following are some typical uses.

- Tiling and Stitching:
  Images that are too large to be handled in one piece are cut into equal squares; each square is then treated individually. Afterwards, the tile results are stitched together.

→ **Tile and Stitch Large Scenes** on page 375

- Refining relatively small image objects that have been found already:
  Apply a chessboard segmentation with a small square-size parameter to those image objects for further detailed analysis.
- Applying a new segmentation to an image object:
  Let us say you have an image object that you want to cut into multiresolution-like image object primitives. Because multiresolution segmentation is a bottom-up technique, you must first apply chessboard segmentation with a small square size, such as 1 (one). Then you can use those square image objects as starting image objects for a multiresolution segmentation.

9.2.1.2 Quad Tree Based Segmentation Application Fields

Compared to multiresolution segmentation, quad tree based segmentation is a reasonable trade between much better processing performance and still-useful results.

The following are examples of typical uses:

- Images with a well-separated background/foreground situation, for example, some cellular assays.
- Images with a significant proportion of background that is easy to separate.
  To increase the processing performance significantly for this type of image, apply a quad tree based segmentation before separating the background. Then use a different segmentation for the non-background areas of interest.

9.2.1.3 Multiresolution Segmentation Application Fields

With any given average size of image objects, multiresolution segmentation yields the best abstraction and shaping in any application area. However, it has higher memory requirements and significantly slower performance than some other segmentation techniques and therefore is not always the best choice.

9.2.1.4 Application Fields for Segmentation by Reshaping

Reshaping algorithms can not be used for initial finding of image object primitives; they require image objects and thus can not be not used on a plain image. However, they are fine for getting closer to regions and image objects of interest.

9.2.1.5 Advanced Segmentation Methods

This section provides an overview how the segmentation can be improved respectively optimized. With some little modifications, the segmentation can be sped up or the extraction quality can be improved.

Image Layer Weighting

Image layers can be weighted differently to consider image layers depending on their importance or suitability for the segmentation result.

The higher the weight which is assigned to an image layer, the more of its information will be used during the segmentation process , if it utilizes the pixel information. Consequently, image layers that do not contain the information intended for representation by the image objects should be given little or no weight.

Example: When segmenting a geographical LANDSAT scene using multiresolution segmentation, the segmentation weight for the spatially coarser thermal layer should be set to **0** in order to avoid deterioration of the segmentation result by the blurred transient between image objects of this layer.

The image layer weights are defined in the **Algorithm parameters** area of the **Edit Process** dialog box. Here you set the image layer weights using the **Image Layer Weights** dialog box.

→ Create a Single Process on page 132

→ Image Layer Weights on page 447

Adapt the Scale Parameter to Reduce the Influence of Bit Depth

With increasing bit depth (radiometric resolution) of your raster image data the spectral similarity between adjacent pixels decreases.

That means for example, that you are more likely to find two adjacent pixels with identical spectral values in 8 bit data than in 11 bit data.

When using a Multiresolution segmentation or using a Quadtree based segmentation in **Color** mode, you consequently have to choose a larger scale parameter for generating image objects on data with high radiometric resolution.

How to Handle Different Radiometric Resolutions

If your project contains image layers with different bit depths, information from image layers with higher bit depths will have more influence on the segmentation result, since one scale parameter is applied to all image layers.

To equally provide information from image layers with higher bit depths, different weights have to be assigned to the image layers for the segmentation process: image layers with the lower radiometric resolution have to get higher weights.

Use Different Data for Segmentation and Classification

Often image layers are weighted with a 0 during the segmentation process, which means that none of the information provided is used for the generation of image object primitives. However, the information provided can be utilized without restraint for classification.

When working with image data of different resolution combined in one project, it is recommended that the image objects are created based on the high-resolution image data, while the coarser data are merely used as additional information for the classification.

Figure 284: Segmentation result using an aerial photograph

In this example, a high-resolution aerial photograph was used for segmentation. Image objects quite well suited for the separation of houses, roads and types of land cover were created. As the separation of rooftops and roads always poses a problem in shadowed areas, a surface model was used to provide additional information.

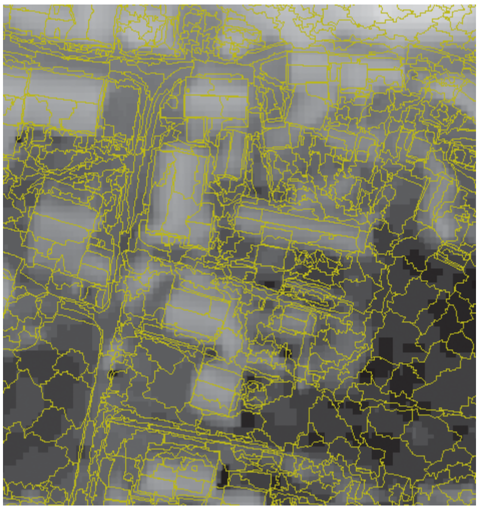


Figure 285: Use the information of low-resolution data for classification

As you can observe, the image information of the surface model is not suited to the extraction of sensible image objects due to its low resolution. The object borders displayed in this window are the same as in the window above, which were built using only the high-resolution spectral information. Nevertheless, the information contained in the elevation image can be well used for discriminating elevated rooftops from low lying roads.

Classify Differently Scaled Objects

Image information is scale-dependent. It is not always possible to extract image objects with only one segmentation run that fits all classes of the classification scheme. An example would be when working on high resolution imagery with the task of classifying rooftop and forested areas at the same time:

In a first segmentation, rather small objects have been produced, which would be suitable for the classification of forested areas or treetops, respectively:

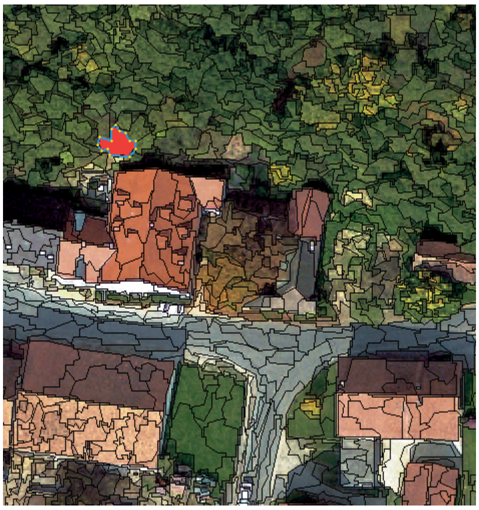


Figure 286: Fine segmentation to classify small image objects

In another, coarser segmentation, objects better suited for the classification of the rooftops are shown. As you can observe, the forested areas are represented by a few objects which cover several treetops:

Figure 287: Coarse segmentation to classify larger image objects

An advisable approach to solving this problem is to classify treetops on the finer image object level and rooftops on the coarser one. In a subsequent refinement procedure, both image object scales can be combined in one single image object level.

Influence the Borders of Subobjects

Use the structural information from a coarser image object level to limit the borders of subobjects throughout the segmentation.

If a sublevel is created for an existing image object level, all new subobjects will be situated within the borders of the superobjects. This principle is very useful for applications like change detection. If you intend to compare a classified image object level of the current year with an already existing thematic land use classification (previous year), you can use the borders of both levels for the segmentation of a new sublevel. This new level is segmented as a sublevel of the current level, with segmentation depending only on the thematic layer (by weighting all image layers with 0). Because the borders of the super-objects from the current classified level have to be considered in the segmentation, the new sub-level is divided as shown below. In doing so, changes between the thematic layer information and the current image object level are extracted as separate image objects. These image objects can then be further classified.

9.2.2 Identify Features that Help Solving Your Task

At the beginning of developing rule set modules, such as isolating or segmenting a specific image structure, you have to identify the **Definiens** features that can be used to distinguish your regions of interest. Based on this investigation, you further implement the rule set module.

The basic methods focus on how to:

- Find features that make the difference
- Combine features.

9.2.2.1 Find Features that Make the Difference

A frequent task in image analysis is to reduce the number of features utilized for classification to the essential ones.

Use the Feature View

To find suitable features, try the effectiveness of several ones provided by the feature tree in the **Feature View** window.

→ Compare Feature Values by Using the Feature View on page 179

Preconditions:

- At the beginning of a rule set, you need to do an initial segmentation of the image. This provides image objects of which feature values can be calculated automatically.
- In the middle of a rule set, that is you have already done a number of segmentation and classification steps, make sure that the available image object primitives are of suitable size and shape in order to return useful feature values.

The goal is to find or to create features that separate different image object classes. Ideally, the found feature displays in the **Feature View** window one region of interest (ROI) blue and the other green, with minimum overlap.

The feature view setting will always color all image objects of an image object level based on their feature values. This is inconvenient in cases where you want to leave a large part of the image as it is and you are interested in separating only a small fraction of the image into two classes. An alternative trick to using the **Feature View** window is to create a block of processes that allows you to do this separation by one execution. This could be as easy as the following:

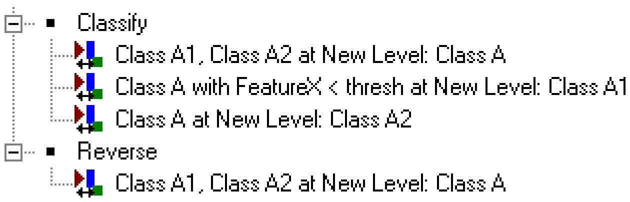


Figure 288: Process block for testing features.

That way, you quickly can change a feature (**FeatureX** in the graph) and threshold used and try what the effect is by executing the **Classify** block. You can also run the **Reverse** block and go back to the situation before the classification.

Try Multiple Features

Usually you have to try quite a number of features before you find the one that solves your class separation problem.

Generally, formulate a human language word model of how you would separate classes. Ask yourself or a domain expert, how do you, as a human being, distinguish the different regions of interest? Consider how these visual qualities translate into **Definiens** features.

It is a good idea to become familiar with the large choice of available features and how they are structured in the feature tree. See the reference book for a complete features dictionary.

→ Features Overview on page 168
→ Reference Book

9.2.2.2 Combine Features

If you can not separate two classes using a single standard feature, combine them by the following methods:

- Create a customized feature.
- Use the class description.
- Combine both methods by applying subsequent classification steps with different features.

We recommend that you combine features that are independent from another. That means they express different qualities of the object class that you are trying to describe. Using an additional feature that is strongly correlated to a feature that is already used does not improve the separability of regions of interest.

Further, do not use too many features. Using more features than necessary leads to overfitting. You may be able to classify the image objects in a correct way in a single image, but the ruleset will most like not be transferable to other images.

→ About Robustness of Rule Sets on page 344

Example: If you know that your objects of interest, for example nuclei have elliptical shape you might add a whole bunch of different features that somehow relate to ellipticity. This does not necessarily improve the class separation, as all these different features may also just add noise. You may spend considerable time trying to identify the feature that best captures ellipticity in your images, and then work with that one feature.

Combine Features into Customized Features

You can combine any number of features by creating a customized feature.

→ Use Customized Features on page 269

This gives you the freedom to combine them with any standard mathematical formula.

Combine Features Using the Class Description

A method of combining several features is the use of class descriptions. They can be seen as a different type of customized features. The feature value returned is represented by the class evaluation, a number between 0 and 1.

→ About the Class Description on page 244

Coming up with good class descriptions is critical for efficient rule set development. Good class descriptions will improve first-shot-detections and considerably reduce the need for later clean-up. Class descriptions describe the so-called image reality often in a concise and conceptually simple way. That leads to rule set code that is much easier to read than if the same information is represented within the **Process Tree** window.

Apply the logic of your assumptions in order to figure out which logical operators to choose. For example: Is the nucleus **round AND small**, or **round OR small**? In a case like this, where you know what the logical operator should be, apply that operator rather than trying all possible operators and analyzing the result. Instead of the AND operator you may also use the geometric mean. It can be thought of as a soft **AND**.

Mixed Methods for Combining Features

Instead of combining several features into one customized feature or a class description, it can make sense to use the features one after another.

During rule set development, this makes it easier to understand how a single feature effects the classification. It also gives you an idea how much of the problem has been solved using one feature and how much of the development task is left to be solved using the remaining features.

Concerning procesing performance, applying features one after another can lead to shorter run times. That is especially the case when one or more features are rather performance expensive to calculate.

9.2.3 About Robustness of Rule Sets

A rule set is robust if it can be applied successfully on a set of similar images including outliers.

9.2.3.1 Image Variability and Robustness

This section discusses sources of variation and its relationship to robustness.

Sources of Variation

Some examples for sources of image variation are

- Noise, or speckle may stem from the specific imaging modality like dust on the lens, be features of the structures themselves, or come from other disturbing factors like clouds in remote sensing images.
- Intensity shading might occur within or between different images. Reasons can be non-constant staining or illumination, a variation in tissue thickness. In remote sensing, season, daytime, or weather conditions may cause variation.
- A single class of objects can show different structural or phenotypical appearance. In biological systems, such variations are inherent. Also, cutting three-dimensional objects into two dimensions can result in different appearances of similar objects. In remote sensing, buildings or streets of the same kind can have a variation of architectures.

Aiming for Robustness in Production Mode

In order to achieve the best possible robustness, it is crucial that you have a good overview about the variability of images and other data in your workspace. With that knowledge, assemble a representative training subset of images for development.

Of course the achievable robustness depends crucially on the quantity and the quality of variation occurring with that training set of images. Thus the images used for development should show a good representation of the expected variation in the production mode. This has to include so-called error images, that are images in which the information can not be read in the regular way. Also, empty images should return emptiness as a result rather than false numbers. In short: The rule set has to "learn" the problems that might occur. Otherwise it will not be able to cope with them.

→ **Estimate Image Variability** on page 329

Generally, one can expect that the accuracy of the production mode is enhanced by the following:

- Using a large training set.
- Using a training set that represents the image variability in an optimal way.
- Developing a rule set that leads to a high accuracy across the training set.

9.2.3.2 Use Robust Descriptors

→ Features Overview on page 168

When you use spectral features as descriptors, be aware that features with absolute values, like **Mean layer**, **Mean Brightness**, are rather sensitive to intensity shading.

In many cases it is better to use ratio (that is layer values divided by brightness) or any other spectral arithmetics that are more robust against overall brightness variation within an image or between several images.

Another good trick is to use features that compare spectral attributes of an object with those of objects in a neighbourhood. All features found in the group **Object features > Layer values > To neighbors** fall in this group, but you may as well create customized features that do the job.

In general, shape features are relatively robust, since most types of image variability affect spectral information rather than shape. For the same reason, class-related features support robustness.

9.2.3.3 About Methods of Settings Thresholds

If you can use correlating features, you may prefer adjustable thresholds instead of fixed ones because adjustable thresholds can be used more flexible in rule set development.

For the implementation of flexible thresholds you can use different methods, which are explained in this section:

- Use statistical values.
- Start Conservative, Then Grow!
- Work relatively.
- Use Class Descriptions.
- Use customized features.

About Fixed Thresholds

→ Make Assumptions Concerning the Strategy on page 327

In many cases you should avoid fixed threshold as they are based on narrow assumptions.

However, in some cases though fixed thresholds may be helpful. When you are using extreme values, which you assume will always fit your assumption, you can use these as a seed to classify the brightest objects, for example, and from there use the parent process object (PPO) and let the seed grow into similar bright regions (**Mean diff. to PPO(x)**) or until the **Standard Deviation** feature rises significantly.

Use Statistical Values

Use the **Compute statistical value** algorithm for getting a threshold specific for every image.

For example, if you assume that more than 10% of your image are always covered by bright areas, you might use the quantile (90) to get the value to separate these highest 10%. If the assumption of the 10% is incorrect though and an image has only 5% covered by bright area, you will get a wrong threshold.

This method is usually very powerful for background detection. If there is a strong spectral background variation not only between images but also within images, apply the method of statistical measurements in a local way. Cut the images in squares or - if the variation occurs purely in the horizontal or purely in the vertical direction - in stripes and find threshold values for each segment individually.

Start Conservative, Then Grow!

There is no guarantee that you find a way to calculate threshold values that distinguish between two classes without mistakes.

A trick then is to start with a conservative classification of one let us say **Class A**. For a conservative classification, you choose the threshold such that you have false negatives but no false positives. Then grow the number of image objects classified as **Class A** by comparison with direct neighbor image objects in an iterative way: For each non-**Class A** image object that borders **Class A** objects, compare the color difference between the image object and **Class A** neighbors to the difference between the image object and non- **Class A** neighbors. If the latter is larger, classify as **Class A**.

Work Relatively

By using the Parent Process Object (PPO) concept and the referring features like **Diff. to PPO(x)** and **Ratio to PPO(x)**, you can treat every image object independently and react very specific to different situations.

→ Use Parent Process Objects (PPO) on page 200

For example you can fuse objects with other objects having 1/10th of their size instead of using a fixed size which might not always fit.

Use Class Descriptions

Use the class descriptions to collect different conditions for classes.

→ About Classification with Class Descriptions on page 212

In most cases objects are defined through a combination of different features which supplement each other. An object can be classified e.g. if its very dark, but also if its less dark but than it needs to be more round. Adding these conditions in the class description allows you to reflect these dependencies. Using the **Classification value of** algorithm then helps you to detect to which extend the objects fulfills the class conditions combined. As operator within the class description you should select one of the **Mean** operators. Using the **Class Evaluation** tab of the **Image Object Information** window, you can easily control how far objects fulfill the single conditions.

Use Customized Features

To reflect the fact that objects can be let us say dark blue or homogeneous bright, you should use the customized features to simplify detection. Here you can combine all features like **Mean Values with Ratio** or **Standard Deviation**.

→ Use Customized Features on page 269

9.2.4 How to Handle Large Data Sets

Many image analysis tasks require large data sets to be analyzed. Large data sets beeing raster images of a size of 10.000 x 10.000 pixel and above. Applying object based image analysis solutions to data sets of such size, implicate the challenge of being able to handle the resulting large amount of image objects.

Since **Definiens** software is restricted to a 32bit address space the amount of image objects addressable is as well. The upper limit of addressable image objects varies with number of image layers, bit depth of the image data, segmentation algorithm used to produce the image objects and complexity of image objects. The limit lies between 2 and 5 million image objects. If more objects are generated, the segmentation is

→ Maximum Image Size on page 49
→ About Processing Limitations on page 370

aborted and the image object hierarchy removed. To avoid this, you have to take care to limit the number of image objects. To do this, several methods are available.

9.2.4.1 General Hints to Limit the Number of Image Objects

When developing rule sets, we recommend that you consider various simple rules you may follow to keep the number of image objects:

- Avoid **chessboard segmentations** with small chessboard size.

→ Chessboard Segmentation on page 160

- Avoid **quadtree based segmentations** with small scale parameter.

→ Quad Tree Based Segmentation on page 161

- Avoid **multiresolution segmentations** with small scale parameters.

→ Multiresolution Segmentation on page 162

- Instead of segmenting large objects as a whole, try segmenting image object borders only.
- Work sequentially.

9.2.4.2 Tile Images

A quick way to limit the maximum number of image objects is cutting a large data set into smaller subsets, called tiles. There are several basic approaches for tiling a dataset:

Manual Tiling

With the manual tiling method, the data set is manually tiled using some preprocessing software.The newly generated, smaller data sets are loaded to **Definiens** software and processed; the results are merged outside of **Definiens** software. This method can be implemented with a minimum knowledge of rule set development, but has the disadvantage that it is the most labor intensive approach and consumes more storage, since data sets are duplicated.

Local Tiling

The local tiling method is suitable for working with a standalone installation of **Definiens Developer** only, or you do not have **Definiens eCognition Server** available. That means you have to process data locally.

→ Installation Guide

Here, you tile the image by generating an image object hierarchy representing the tiles and locally processing one tile after another. This is done by running a suitably large chessboard segmentation and subsequently fine segmenting one tile after another. A practical approach is to start by clasifying the top left corner image object as class **active** using the **find domain extrema** algorithm, fine segment it, classify it and merge adjacent image objects of the same class. The whole sequence can be programmed in a loop so that out of the non-processed image objects, the topmost is identified and processed until no unprocessed image objects remain. This approach allows limiting the maximum amount of image objects but will not speed up the processing.

Server Based Tiling

Precondition: In case you have **Definiens eCognition Server** available, you can used server based tiling methods.

Here tiling and stitching can be performed easily using either workspace automation or the manual tiling and stitching command provided in toolbar and main menu. Both methods allow to tile an image in a defined tile size. Each tiles the analyzed by the with a defined rule set. After processing is finished, the individual tiles can be stitched. If necessary, the stitched result can be analyzed by a further rule set, removing unwanted tiling effects. Server based tiling methods hold the advantage, that they commonly are easier to implement and further allow to automatically distribute computing among multiple processing nodes of a **Definiens eCognition Server** installation.

→ Use Cases: Multi-Scale 1–3 on page 297
→ Tile and Stitch Large Scenes on page 375

9.2.4.3 Local Segmentation

Local segmentation is basically the same approach as local tiling. A defined image object is segmented, classified and image objects of the same class are subsequently merged.

The usual approach is to generate a rough classification using large scale image objects generated by multiresolution segmentation. Then either one by one or classwise, image objects are processed.

→ Multiresolution Segmentation on page 162

Local segmentation has the advantage, that it is usually easier to implement than a local tiling approach. Depending of how it is ralized, it may be slower, since a first large scale segmentation consumes extra time.

9.2.5 Improve the Performance

The reduction of the computing time often is required for analyzing images. The major way of doing this is to write smart rule sets.

To do so, you may follow some generic strategies, ideally combined in a way that suit your performance needs:

- Reduce the number of image objects.
- Avoid unnecessary calculation of performance expensive features.
- Avoid image objects with large neighbourhood lists.

9.2.5.1 Reduce Number of Image Objects

Limiting the number of image objects created by segmentation processes may be a prerequisite for successful image analysis due to limited memory. In addition to that, rule sets with reduced numbers of image objects usually also run faster.

→ How to Handle Large Data Sets on page 346

Two basic methods aim to:

- Speed up the initial segmentation.
- Prefer local operations.

Speed Up the Initial Segmentation

Initial segmentation of the image is a basic element in rule sets. Depending on the selected algorithm it can cost a lot of computing time.

For example, the **multiresolution segmentation** algorithm returns the best image object primitives, but at the same time takes longer than all other segmentation algorithms.

There are a couple of tricks to speed up the initial segmentation without significant loss of quality of the resulting image objects. You should invest some time in playing around with different segmentation algorithms and settings and try to find the best for your first step in the rule set.

Compare Algorithms for Initial Segmentation

With a small sized **chessboard segmentation** you can treat every pixel independently, but this creates lots of image objects which slows down the performance.

A small sized **quad tree based segmentation** will result in similar small image objects where needed, but provides bigger image objects in homogeneous areas. This produces less image objects, which is preferable.

If you have image objects which are homogeneous in color and also typical in shape, the **multiresolution segmentation** algorithm may give you very good results.

→ Segmentation Principles: Top-down and Bottom-up on page 159
→ When to Use Which Segmentation? on page 337

Combine Quadtree Based and Multiresolution Segmentations

Instead of starting a rule set with a slow **multiresolution segmentation**, you can do a **quadtree based segmentation** first before applying a **multiresolution segmentation** algorithm.

The runtime of multiresolution segmentation is roughly proportional to the number of image object mergers. Thus, when you apply it to a pixel image, most computing time is spent to create rather small image objects of 2-4 pixels. The quadtree based segmentation with a scale parameter that results in image objects of that size is much faster. That means: if you do such a quadtree based segmentation first, then merge those small image objects with multiresolution, you can save a lot of computing time wheras the result does not differ significantly.

Use Multiresolution Segmentation Region Grow

The **multiresolution segmentation region grow** algorithm is a simplified version of the **multiresolution segmentation** algorithm. Thus, it produces image objects that are very similar to those produced by the multiresolution segmentation algorithm, while it runs much faster.

This algorithm however can not be applied to the basic pixel level; in order to use it, you have to do another segmentation of the image first. This may be a **quadtree based segmentation** with a small **Scale parameter** or a **cheesboard segmentation** using a small **Object size** parameter.

Use Contrast Filter Segmentation

Contrast filter segmentation is a very fast algorithm for initial segmentation.

→ Reference Book

In some cases, it is possible to find objects of interest in a single step using it. That also means that you do not have an intermediate step with image object primitives that are much smaller than objects of interest, and thus you keep the number of image objects lower than with usual approaches that involve the creation of image object primitives.

Prefer Local Operations

Sometimes you need to do an operation that requires you to cut down image objects of interest into small pieces and merge them subsequently. Try to do both steps for each image object individually rather than cutting down all image objects and then applying the merger to all image objects.

That way, you will still get a large number of image objects for the currently treated obeject of interest. But by doing the required merger before cutting the next object if interest, you avoid a large total number of image objects in the image.

9.2.5.2 Avoid Unnecessary Calculation of Performance Expensive Features

Some features available in the **Definiens Developer** are rather expensive to calculate. This is especially true for features that retrieve information about image objects in the neighborhood, for example **distance to Class**.

Evaluate Less Performance Expensive Features First

When more than one feature is used to sort out a large number of image objects, try to use the less expensive feature first.

By doing so, you reduce the number of image objects to be evaluated in the first step. The more expensive feature, which is used in the second step, then needs to be evaluated only for the smaller number of remaining image objects.

Use Local Processing to Reduce Calculation of Expensive Features

Use classes with few image objects as anchor for operations on image objects of a class with many image objects.

Consider a situation where you have a large number of image objects of **Class A**, a small number of image objects of **Class B**. You are only interested in those **Class A** image objects that have a Class B neighbor. Instead of calculating the feature **Existence of Class B nghb(0)**, create a parent and a child process, where the parent process only defines the **Class B** image objects, the child process goes to all **Class A** neighbor of the parent process object. That way, the large number of **Class A** image objects that are not neighbors of **Class B** image objects, are never addressed.

→ Focus Processes by Using the Domain Concept on page 197

Use Variables to Store Values of Performance Expensive Features

When you can not avoid using expensive features for a large number of objects, make sure you do not calculate the feature more than once.

Instead of using the feature directly in the code, store the return of those features in objects variables and then use the object variables downstream in the code.

→ Update Variable Algorithm on page 262

9.2.5.3 Avoid Image Objects with Large Neighbourhood Lists

Try to avoid operations on large image objects that have many small neighbors.

A typical situation is during background finding, where one class **Background** image object branches across large parts of the image and directly touches a rather large number of small **Background** and other image objects. Then you run a **merge region** process and you have to get a cup of coffee before the program reacts again.

The reason is that in **Definiens Developer**, each image object knows its direct neighbors. The merging is done image object by image object, so that after each single merging step, the list of neighbors of the large **Background** image object has to be updated. That takes a lot of time.

A workaround here is a three-step merging procedure:

1. Apply a **chessboard segmentation** to the large **Background** image object with an intermediate scale, for example 10. This results in a number of intermediate size **Background** image objects, none of which has a very long neighborhood list.
2. Run a **grow region** process with the larger **Background** image objects as the seed, the smaller ones as candidates.
3. Merge all together.

→ Segmentation by Reshaping Algorithms on page 165

In some situations this procedure can save a lot of time. It can, in other situations, waste some time, but never a lot.

9.3 Find and Handle Image Structures

This section provides you with general implementation strategies focusing on finding and handling image structures.

Most strategies require existing image objects, which are then modified or extended to result in the final image objects of interest. The strategies are applicable in a generic way to find or improve image objects for the most diverse tasks. This collection represents the most commonly used approaches and is meant to give you a starting point for optimizing you rule set. The strategies can be can be freely combined, modified or extended as you see fit.

9.3.1 Shrink-and-Grow Strategy

There are a variety of different contexts in which this technique is used, for example:

- Smoothing the border by removing noses comparable to the **morphology** algorithm **opening**.
- Separating image objects on narrow bridges comparable to the **watershed transformation** algorithm.

Implementation for Border Smoothing

1. Chessboard the border of the regions of interest (ROI).
2. Reclassify all image objects that currently represent the image object border to a **TempClass**. Possibly, you need several iterations.
3. Reclassify **TempClass** image objects touching the ROI. Typically, you need the same number of iterations.

Take a look at the following screenshots to get an impression how this approach works:


Figure 289: Region of Interest (ROI).


Figure 290: Step 2.


Figure 291: Step 3.


Figure 292: Step 4.

Implementation for Object Separation:

1. Chessboard the border of the ROI.
2. Reclassify all image objects that currently represent the image object border to a **TempClass** . The border should be deep enough to contain the bridges where the image objects should be cut.
3. Merge ROI image objects, forming your seeds.
4. Grow the seeds into the **TempClass** image objects using the grow region algorithm. Make sure to define only the **Tempclass** as candidate. Limit the number of steps only if you want to smooth the image objects.


Figure 293: Region of Interest (ROI).


Figure 294: Step 2.


Figure 295: Step 3.

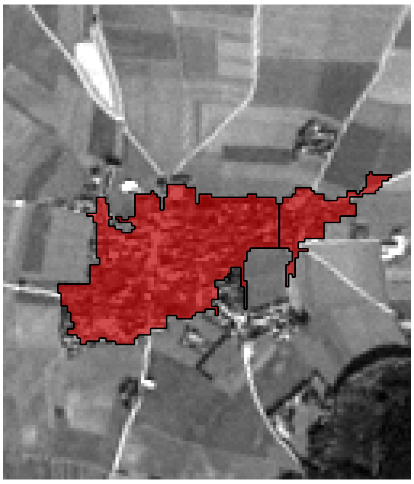


Figure 296: Step 4.

9.3.2 Grow-and-Shrink Strategy

Among the purposes of this versatile strategy are:

- Smoothing the object border by filling grooves similar to the morphology algorithm closing operation.
- Identifying densely clustered regions by aggregating seed areas to larger areas of interest.

Implementation for Border Smoothing

Chessboard the border of classes touching the regions of interest (ROI):

1. Chessboard the image objects touching the ROI.
2. Reclassify all image objects that touch the ROI to **TempClass**. Possibly, you need several iterations.

3. Reclassify **TempClass** image objects not touching the ROI to their original classes. Typically, you need the same number of iterations.

4. Classify remaining **TempClass** image objects as ROI

If you want to reclassify the image objects to their original class, the information regarding the original class identity must still be available. If the class identity is not obvious, it could be stored in an object variable, or in a separate image object level.


Figure 297: Region of Interest (ROI).


Figure 298: Step 2.


Figure 299: Step 3.


Figure 300: Step 4.

Identify Densely Clustered Regions

This strategy aggregates seed areas to result in larger areas of interest.

1. Chessboard the image objects touching the seed areas.

2. Reclassify all image objects that touch the seed areas to **TempClass**. Possibly, you need several iterations.

3. Reclassify **TempClass** image objects not touching the seed areas to their original classes. Typically, you need the same number of iterations.

4. Classify remaining **TempClass** image objects as ROI.


Figure 301: Region of Interest (ROI).

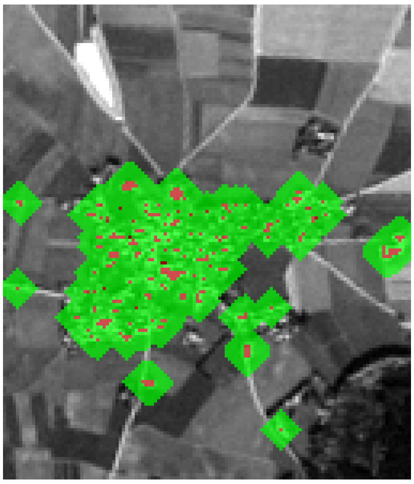


Figure 302: Step 2.

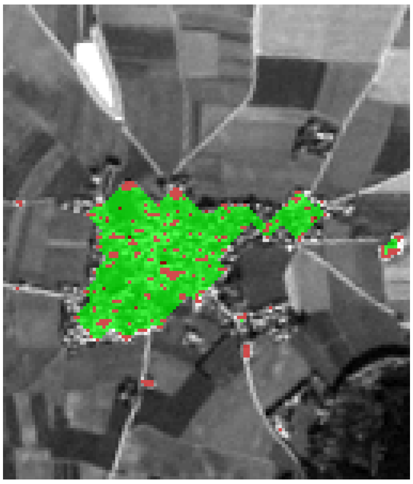


Figure 303: Step 3.


Figure 304: Step 4.

9.3.3 Grow-From-Seed Strategy

In many situations your regions of interest (ROI) cannot be separated cleanly from the rest of the image using a single classification process.

This is usually because there is no feature (or combination of features) which allows you to clearly separate the entire ROI. In these cases, a useful strategy is to first identify candidates which you can safely assume belong to your ROI (the seed), and then grow these seeds into "less likely" candidates. This only works if the seed and the new candidates are in spatial vicinity.

Using a Single Class Description

A particularly simple way to implement this strategy is to create a class description for your ROI. The grow-from-seed strategy then comprises two processes:

1. Find a seed (use a conservative classification value threshold).

2. Grow seed (using a more liberal classification value threshold for the candidates).

Note that it is often a good idea to not only grow into candidates, but also to grow into seeds, so that continuous regions are merged even though they may originate from different seeds.

Example: In the pictures, **Water** seeds are defined based on conservative classification values. Image objects bordering the seeds are then classified as **Water** candidates on more liberal classification settings. In this example the candidates are classified instead of merged for better visualization.

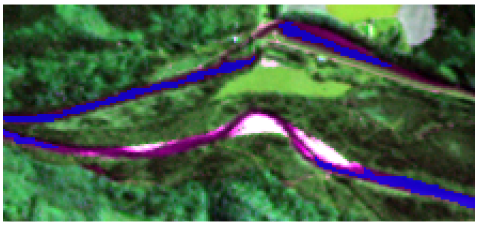


Figure 305: Water seeds.

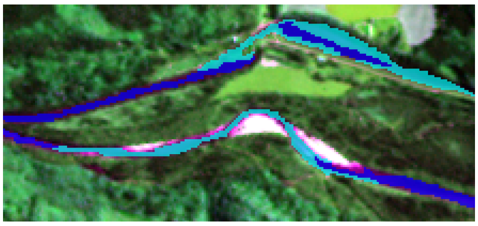


Figure 306: Water seeds and candidates.

Using PPO Features

A particularly common requirement is that you want to grow a seed into everything that is very similar to the seed. For example, you may know that the candidate should have a color "similar to the seed" without being able to specify the desired color in absolute terms. This can be implemented elegantly by using PPO features within the **image object fusion** algorithm.

→ Use Parent Process Objects (PPO) on page 200

Note that in cases where the features show a gradient across the image (often the case for background regions), it may be wise to not merge all the seed elements together, but instead to keep seed elements local to some degree (applying a relatively coarse chessboard to seed elements may do). In this way, the gradient computation refers to the local gradient, and is not affected by any global gradients that may be present in another part of the image.

Growing Across Weak Brightness Gradients

For a requirement concerning similar brightness, consider using a **ratio PPO** feature, as the ratio is generally a more robust estimate of brightness feature gradients, compared to the difference. As a result, we would like to merge all image objects together with a **ratio PPO** similar to 1.

Note two important issues:

- If there is any risk that the denominator of the ratio is 0, the feature should be modified in a way to avoid this. A straightforward (though probably not always optimal) solution is to create an arithmetic feature that adds 1 to the original feature.
- The **image object fusion** algorithm enables us to merge objects with a symmetrical feature range around 0 (by setting the **Use absolute fitting value** option), but it does not allow us to merge candidates with a feature range around 1. Therefore, instead of using the PPO feature directly, one should use the logarithm of the PPO feature, which then will map similar items closely to zero.

Growing Across Weak Color Gradients

A straightforward way of describing color differences is to compute the Euclidean distance of the two color representations.

To achieve this, create a different PPO feature for each image layer. (Set the **Distance** to zero (0).) Then create an arithmetic feature that adds the squares of your newly created PPO features and evaluates the square root of the result. (The square root of x is described as **x^0.5**).

9.3.4 Chessboard-the-Border Strategy

Image processing algorithms often require that the image object or class border of an image object be finely segmented, for example, when using the shrink-and-grow strategy.

Sometimes, a one pixel deep segmentation is sufficient, sometimes a higher pixel (or image object count) is desired. While it is possible in principle to chessboard the border region of an object by simply chessboarding the whole object, this is typically a time-consuming process. Therefore it is worthwhile to consider the more sophisticated implementations detailed below.

Chessboard the Class Border

1. Begin with coarse chessboard segmentation of the regions of interests (ROI).
2. Classify the border image objects as a **TempClass**.

   Useful features are:

   - **Existence of** non-ROI objects: It will not detect image border.
   - **Rel border to** ROI < 1: It will detect image border.
3. Repeat step 2 until the border is deep enough.
4. Apply fine **chessboard segmentation** of the **TempClass** image objects.
5. Reclassify **TempClass** objects which are not touching non ROI image objects as **ROI**.
6. Merge the **ROI** image objects.


Figure 307: Region of Interest (ROI).


Figure 308: Step 1.

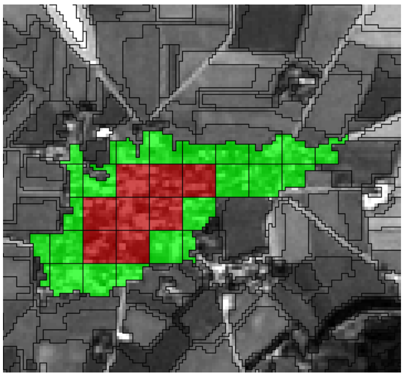


Figure 309: Step 2.

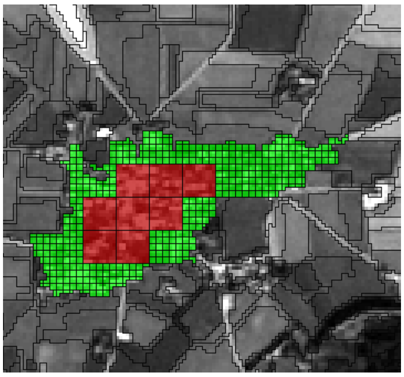


Figure 310: Step 3.

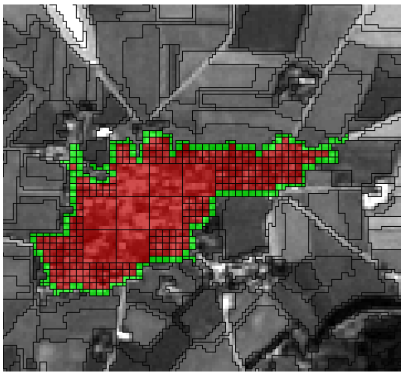


Figure 311: Step 4.

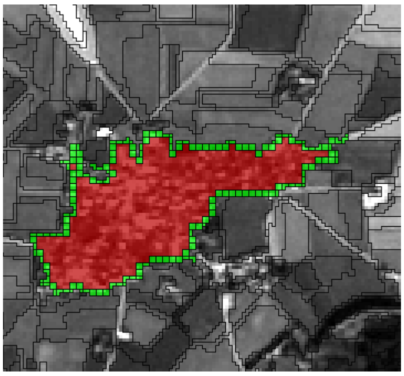


Figure 312: Step 5.

Chessboard the Object Border

Here, the obvious challenge is to keep adjacent image objects of the class **ROI** as separate image objects. Once a chessboard segmentation is carried out, information on the original object shapes is lost. Therefore, the following modification is suggested:

1. Copy the image object level and work on the lower of the two.
2. Proceed as above, but use **Rel. inner border to superobject** < 1 to initially identify the border region.

➔ Reference Handbook

This will allow keeping the original image object shapes, since they are available in the higher level.

9.3.5 Border Smoothing

The purpose of the border smoothing algorithms is to achieve a smooth object outline.

Noses are removed and grooves are filled. It is important to be aware that while the result may look nicer than the original classification image, it may in fact reflect the image data to a somewhat lesser degree. The relative importance of esthetics versus accuracy should be considered carefully.

Implementation

- Use the **morphology** algorithm.
- Use the **border optimization** algorithm.
- Use the shrink-and-grow strategy or the grow-and-shrink strategy.
- Use Gaussian smoothing:

  1. Create a temporary image layer using the **Classified as Class A** feature.
  2. Run a Gaussian filter using the **convolution filter** algorithm on the temporary image layer.
  3. Resegment the existing image object level to represent the image objects of the temporary image layer.
  4. Classify all image objects with value above 0.5 in the temporary image layer as **Class A**.
- Use the **pixel frequency filter** algorithm for smoothing:

  1. Create a temporary image layer using the **Classified as Class A** feature.
  2. Run a **Pixel Frequency Filter** on the temporary image layer.
  3. Resegment the existing image object level to represent the image objects of the temporary image layer.
  4. Classify all image objects that have a value of 1 in the temporary image layer as **Class A**.

→ Shrink-and-Grow Strategy on page 351
→ Grow-and-Shrink Strategy on page 353

9.3.6 From Coarse to Fine Image Object Scale Classification

When trying to isolate relatively large regions, a good approach is to first look on a relatively coarse image object scale and identify the regions of interest roughly. In a next step you then refine the boundaries of the found region of interest. To do so,the initial classification often then needs to be refined. In some cases a simple border smoothing may be sufficient. In other cases, it is desirable to also increase the border resolution, in other words, to move from a coarse image object scale to a finer image object scale.

Implementation

This refinement can be done in two steps:

1. Increase resolution.
   For instance, assume that you have separated classes: **tissue A** and **tissue B**. Now you reclassify the border between the two classes, possibly several pixels deep.

2. Assign border region to adequate classes.
   In the most simple case, this is based on a simple threshold. To choose a suitable threshold, you may evaluate the median of the feature used for separation (typically some combination of staining features) and assign the **_TempClass1** image objects according to whether they are above or below the threshold.

Note, that you may need to do some further cleaning up afterwards, such as removing islands that you have just newly created and border smoothing.

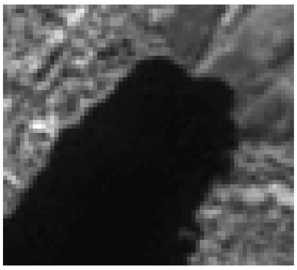


Figure 313: Source image.

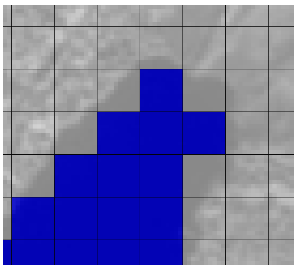


Figure 314: Rough image obejct scale classification.

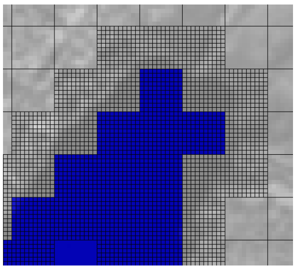


Figure 315: Increased border image object scale on ROI.

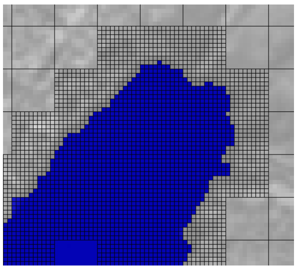


Figure 316: Refined ROI.

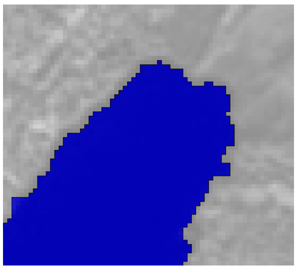


Figure 317: Final ROI.

10 Process Data

Contents in This Chapter

Processing applies ruleware, which may be a rule set or a solution, to the image data. Thus, you receive image analysis results, and the results in turn enable you to evaluate the ruleware.

There are different methods for processing data:

- Analyze a single project by executing manually a rule set in the **Process Tree** window.
- Analyze a single project by executing manually a solution in the **Analysis Builder** window.
- Analyze single or multiple projects in a **Workspace** in an automated environment.
- When you analyze single projects with large image size, the image analysis maybe speed up by creating subsets or tiles of a larger project or image.

10.1 Analyze Single Project Manually

Analyze an open project by executing a process in the **Process Tree** window.

The **Process Tree** window contains the rule set which consists of a number of processes. These processes perform the segmentation of the image into image objects and the subsequent analysis of these image objects.

While developing rule sets, execution of single processes and analysis of single projects allow you to test single processes and whole rule sets iteratively.

Precondition: You must have built a rule set or use an existing rule set file.

1. If not yet displayed, you can load a process into the **Process Tree** window, do one of the following:

→ Load a Process on page 144

   - Right-click inside the **Process Tree** window and choose **Load Rule Set**.
   - Choose **Process > Load Process** or **Load and Merge Process** respectively from the main menu bar.

   The process is loaded and displayed in the **Process Tree** window.

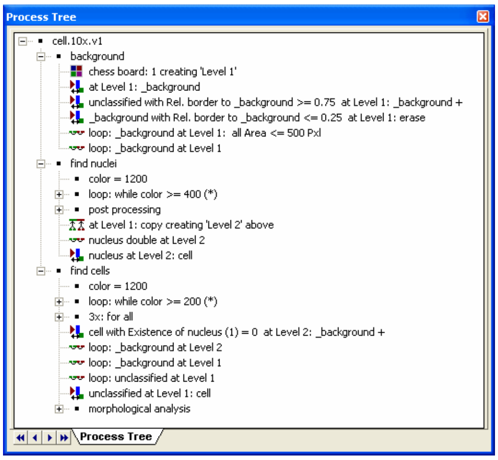


Figure 318: Process Tree window.

2. You can execute the complete rule set or selected processes only. Executing a parent process will also execute all its child processes. To execute a process, do one of the following:

→ Execute Processes for Testing on page 151

   - Select a process and choose **Process > Process Commands > Execute** from the main menu bar.
   - Right-click the (parent) process you want to execute and choose **Execute** from the context menu.
   - Select a process and press F5.

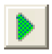
10.2 Execute Solution Manually for Testing

For testing solutions, you can execute an image analysis process manually. You can do this on a currently open project or on multiple projects.

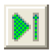
Precondition: You must add and configure at least one action in the **Analysis Builder**.

→ Assemble a Solution from Actions in the Analysis Builder on page 317

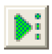
- To execute all assembled actions, click the **Run Solution** button on the **Architect** toolbar. Alternatively, choose **Analysis > Run Solution** on the main menu bar.


Run Solution

- To execute all actions up to a certain stop, select an action and click the **Run Solution Until Selected Action** button on the **Architect** toolbar. Alternatively, choose **Analysis > Run Solution Until Selected Action** on the main menu bar. All actions above and including the selected action will be executed.


Run Solution Until Selected Action

- To execute a single action only, select an action and click **Run Selected Action** button on the **Architect** toolbar.


Run Selected Action

- To execute a configured solution not locally but on the **Definiens eCognition Sever**, select a project in the workspace window. Click the **Run Solution on eCognition Server** button on the **Architect** toolbar. This option is needed if the solution contains actions with workspace automation algorithms.


Run Solution on eCognition Server

You can inspect the state of the analysis in the status bar of the main window. On the left side of the status bar, the currently processed action is displayed. In the middle you see a processing indicator.

10.3 Start Automated Image Analysis

Analyze selected projects. To generate results in batch mode, you can start an automated image analysis job that applies a rule set file to single or multiple projects.

Precondition: You must have built a rule set or use an existing ruleware file, which may be a rule set (**.dcp**) or a solution (**.dax**).

1. Select one or more items in the **Workspace** window. You can select one or more projects from the right pane or an entire folder from the left pane.
2. Do one of the following:
   - Choose **Analysis** > **Analyze** on the main menu bar.
   - Right-click the selected item and choose **Analyze**.

   The **Start Analysis Job** dialog box opens.

Analyze Projects

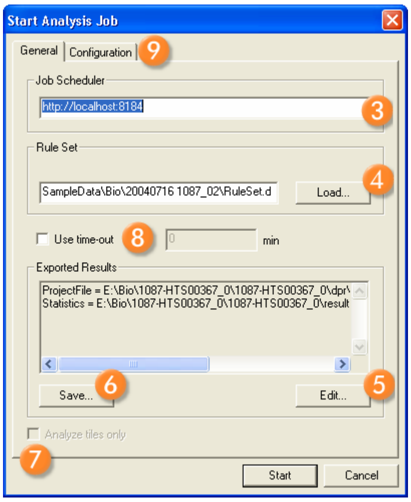


Figure 319: Submit Job dialog box.

3. You can change the **Job Scheduler** 3 which is set to **http://localhost:8184** by default. This is the address of the computer that assigns the analysis to one or more (if applicable) computers.
4. Click **Load** 4 to load a ruleware file for the image analysis. This can be a process file (extension **.dcp**) or a solution file (extension **.dax**).
5. Click **Edit** 5 to configure the exported results and the export paths of the image analysis job in an export template.
6. Click **Save** 6 to store the export template with the process file.
7. If you have created tiles, the **Analyze tiles only** 7 check box is selected by default. Clear it if you prefer to analyze the main project from which the tiles have been created.
8. Select the **Use time-out** 8 check box to set automatic cancellation of image analysis after a period of time that you can define. This may be helpful for batch processing in cases of unexpected image aberrations. When testing rule sets you

→ Configure Exported Results on page 436

→ Tile and Stitch Large Scenes on page 375

can cancel endless loops automatically. The state of projects after this kind of cancellation is **Canceled**.

9. If it is necessary to change the configuration, click the **Configuration** 9 tab.

→ Edit Configuration of the Analysis Engine Software on page 364

10. Click **Start** to start the image analysis.
11. While the image analysis is running, the **State** of the projects displayed in the right pane of the **Workspace** window will change to **Waiting**, then **Processing**, and later to **Processed**.

> **Note**
>
> **Repeat Automated Image Analysis**
>
> If you want to repeat an automated image analysis, for example, when testing, you need to rollback all changes of the analyzed projects to restore the original version. To determine which projects have been analyzed, go to the **Workspace** window and sort the **State** column. Select the **Processed** ones for rollback.
>
> → Rollback All Changes of Projects on page 86

10.3.1 Edit Configuration of the Analysis Engine Software

Use the **Configuration** tab of the **Start Analysis Job** dialog box of the **Start Analysis Job** dialog box to review and alter the configuration for jobs and change versions, plug-ins, drivers, and extensions.

Commonly, you do not need to change the settings.

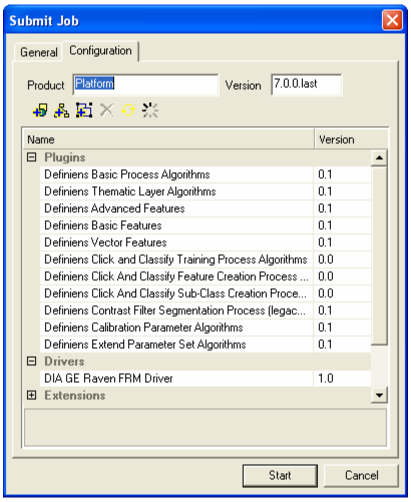


Figure 320: Submit Job dialog box, Configuration tab.

Normally the settings do not need to be altered:

- The **Plug-ins** that display initially are associated with the rule set that has been loaded on the **General** tab.
- The **Drivers** are associated with the project.
- The **Extension** field displays extensions and applications, if any are available.

Change the Engine Version

To alter the engine version, enter the number for the version in the **Version** field at the top of the tab. If the version is available it will be used.

Change a Configuration Item Version

To change the version of a driver or plug-in, you can enter it in the **Version** column. The system will check for the version entered.

Add a Driver

1. To add a driver first load a rule set on the **General** tab to display the associated plug-ins.
2. Load a driver by clicking the **Add Driver** button or using the context menu to open the **Add a Driver** dialog box.

 Add Driver

3. Use the dropdown **Name** list to select a driver and version, if needed.
4. Click **OK** to display the driver in the list.

Add a plug-in

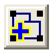
1. To add a plug-in, first load a rule set on the **General** tab to display the associated plug-ins.
2. Load a plug-in by clicking the **Add Plug-in** button or using the context menu to open the **Add a Plug-in** dialog box.

 Add Plug-in

3. Use the dropdown **Name** list to select a plug-in and version, if needed.
4. Click **OK** to display the plug-in in the list.

Add an Extension

1. To add an extension, first load a rule set on the **General** tab.
2. Load an extension by clicking the **Add Extension** button or using the context menu to open the **Add an Extension** dialog box.

 Add Extension

3. Enter the name of the extension in the **Name** field.
4. Click **OK** to display the extension in the list.

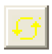
Delete an Item

To delete an item from the list, select the item and click the **Delete Item** button or use the context menu.

 Delete Item

Reset the Configuration

If you have altered the initial configuration, return to the initial state by using the context menu or clicking the **Reset Configuration Info** button.

 Reset Configuration Info

In the initial state, the plug-ins displayed are those associated with the rule set that has been loaded.

Load Client Configuration Information

Load Client Config Info

Click the **Load Client Config Info** button or use the context menu to load the configuration of the current client.

For example, if you are using a rule set developed with an earlier version of the client, you can use this button to display all plug-ins associated with the client.

10.4 Monitor Processing in a Web Browser

If you use **Definiens eCognition Server** for processing, you can control the status of jobs submitted for processing via a web browser.

1. To open display **Definiens Job Scheduler** status page in your web browser, go to the machine that host the Definiens eCognition Grid and choose the **Windows Start** menu > **Start** > **All Programs** > **Definiens Grid** > **Job Scheduler Status Page**.

   Alternatively, you can enter a web address that is identical to the entry in the **Job Scheduler** entry of the **Start Analysis Job** dialog.

   Further, in case a local Job Scheduler is used, you can enter **http://localhost:8184**.

If you use **Definiens eCognition Server** for processing, you can control the status of jobs submitted for processing via a web browser.

2. To open display **Definiens Job Scheduler** status page in your web browser, go to the **Windows Start** menu and choose **Start** > **All Programs** > **Definiens Grid** > **Job Scheduler Status Page**.

   Alternatively, you can enter a web address that is identical to the entry in the **Job Scheduler** entry of the **Start Analysis Job** dialog.

   Further, in case a local Job Scheduler is used, you can enter **http://localhost:8184**.

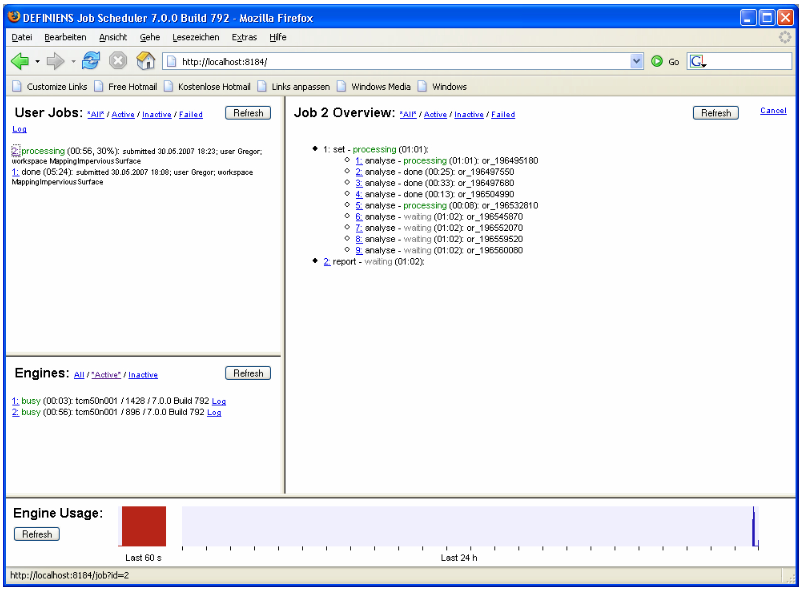


Figure 321: Job scheduler viewed in a web browser.

3. The HTML page is split into four parts, which you can resize the panes by clicking on the dividers and dragging them:

- **User Jobs** lists all jobs on schedule.
- **Engines** lists the Analysis Engine Software instances of participating processing nodes.
- **Engine Usage** displays the capacity utilization of all Analysis Engine Software instances of participating processing nodes of the **Definiens eCognition Grid**.
- **Job** *xy* **Overview**: list all projects of a selected **User Job**.

Review User Jobs

Look in the **User Jobs** section to see all jobs on schedule. There are four options you can use to filter this list.

- **All** is the default.
- **Active** jobs are those currently being processed.
- **Inactive** jobs encompass both successfully completed jobs and those that failed or were cancelled.
- **Failed** lists only those that did not successfully finish. Any filter in use is surrounded by asterisks (this information applies to all filters on the page).

Look at some of the available data in this pane:

1. Click **Active**, to display only jobs currently running.
2. Push the **Refresh** button to reload the site.
3. Click **Log** to see additional information about how the job was processed. The log lists the dates of events, followed by machine and Analysis Engine Software number and the type of event, that are either connecting or shutting down of an instance of Analysis Engine Software.
4. Click on the index number of a job in the User Jobs pane to view its details in the **Job Overview** section.

Review Job Overview

If you wish to stop the current job, click **Cancel** link in the upper right corner.

Click on the item number in front of a job to switch to the **Job Details** pane and review details of the results. Click **1** to view the status of the results, which is one of the following:

- **done**
- **failed**
- **waiting**
- **cancelled**
- **processing**

Information displayed about a specific job includes the start and end times, the version number of your **Definiens Developer** software, the (local) path of the utilized rule set, a list of the image layers submitted for processing and the path of all the output files you specified in the **Configure Exported Results** dialog. Also in case of errors a remarks section will be displayed, providing information about the origin of the error.

View Job Details

In the **User Jobs** section, you can review processed jobs by monitoring the result status. Click on the item number to display details. If a submitted job failed, look in the **Remarks** section of the **Job Details** pane for further information.

Figure 322: Job Details section of the Job Scheduler page displayed in a web browser.

Review Analysis Engine Software Status

In the **Engines** section, the participating Analysis Engine Software instances of participating processing nodes are listed. Filter them by selecting either only the **Active** or **Inactive** instances. The status of an active instances is **idle**. The status of instances whose analysis could not be completed is set to **timeout**. Click on the item number to display details. If an error occurred during processing check the **Remarks** section.

Review Analysis Engine Software Usage

The **Engine Usage** displays two graphs representing capacity utilization of all Analysis Engine Software instances of participating processing nodes of the **Definiens eCognition Grid**. The left graph represents the workload of the last 60 seconds while the right one displays data for the last 24 hours.

10.5 Cancel Image Analysis of an Open Project

You may want to cancel image analysis of a currently open project if the processing seems to take too long or if you want to cancel the job for any other reason.

A processing indicator right of the **Cancel** button in the status bar indicates processing activity by a small oscillating plus sign (**+**).

Figure 323: Cancel button and activity indicator in the status bar.

1. Click the **Cancel** button in the status bar, to cancel a processing image analysis that was manually executed from a currently open project while developing ruleware. The **Cancel Processing** dialog box opens.

→ **Execute Solution Manually for Testing** on page 362

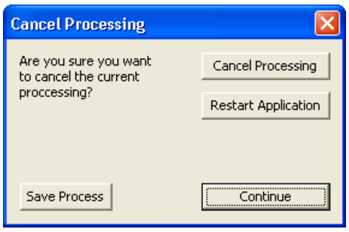


Figure 324: Cancel Processing dialog box.

Processing continues until you make one of the following choices:

2. Click the **Cancel Processing** button to stop the processing image analysis.

> **Note**
>
> for Definiens Developer only
>
> Canceling processing does not mean, that you are automatically returned to the situation before starting the analysis. You might need to delete the existing image object level.
>
> → **Execute a Single Process** on page 141

3. Click the **Restart Application** button to exit and restart **Definiens Developer** if processing cannot be canceled. Before restarting, we highly recommend to save the used ruleware to keep the status of your ruleware development. Click the **Save Process** button to store the rules set as **.dcp** file

4. Click the **Continue** button to close the dialog box without canceling processing.

10.6 Cancel Image Analysis of Waiting and Currently Processing Projects

You may want to cancel image analysis of waiting and currently processing projects if the processing seems to take too long or if you want to cancel the job for any other reason.

Alternatively you can **Use time-out** to set automatic cancellation of image analysis after a period of time that you can define.

→ **Start Automated Image Analysis** on page 363

1. Click into the **Workspace** window.

2. To cancel all projects, make sure that no project is selected. To cancel only particular projects, select the projects to be canceled in the right pane of the **Workspace** window.

3. To cancel the analysis of projects with the state **Waiting** or **Processing**, do one of the following:

- Click the **Cancel Waiting and Processing Projects** button on the **File** toolbar.
- Choose **Analysis > Cancel All Jobs** on the main menu bar.
- Right-click in the right **Workspace** pane and choose **Cancel** from the context menu.

Cancel Waiting or Processing Projects

10.7 About Processing Limitations

Limitations of processing projects are highly dependent on the number and the complexity of created image objects while executing an image analysis.

You have to consider the following processing limitations:

- The number of image objects is theoretically limited to 2^{31} (2 147 483 648). Practically, the limit is lower.
- A related temporary file may allocate too much disk space. The file size depends on the number and the shape complexity of image objects. As a rule of thumb, you can calculate that in case of maximal complexity 1 pixel needs 8 byte disk space.

When analyzing small scenes using regular ruleware, you can ignore processing limitations. However, they will become more distinct with larger images. In such cases there exist different methods for accelerating the image analysis.

→ Maximum Image Size on page 49
→ Process Large Projects on page 370
→ How to Handle Large Data Sets on page 346

10.8 Process Large Projects

If you have to analyze projects with scenes that exceed the processing limitations, you have to consider some preparations.

→ About Processing Limitations on page 370

Projects with scenes within the processing limitations can be processed normally, anyhow some preparation is recommended for accelerating the image analysis or if the system is running out of memory.

The following methods are available for manual and automated image analysis all of them working on some kind of scene subsets:

- Definition of a scene subset
- Working at different scales
- Tiling and stitching of large scenes
- Tiling of large scenes

→ Define a Scene Subset on page 371
→ Work at Different Scales on page 373
→ Tile and Stitch Large Scenes on page 375
→ Tile Large Scenes on page 377

For automated image analysis we recommend developing rule sets that handle the above methods automatically. In the context of workspace automation, subroutines enable you to automate and accelerate the processing specially of large images.

→ Automate the Workspace Processing on page 295

10.8.1 Define a Scene Subset

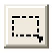
Before starting analysis of a single project, you can define one or more subsets of the scene, which will be represented as new subset projects, also called a subset copy. Defining a subset will accelerate the analysis procedure, as analysis is limited to a selected region of interest (ROI).

As a scene is the combined input data from a project or a workspace, scene copies do not include any results like image objects, classes or variables.

While scene subsets can already be chosen when a project is first created, they can also be selected from an open project through the **File > Modify Open Project** command as long as it has not yet been analyzed. Alternatively, you can define a subset of an existing project as follows:

→ Modify a Project on page 54

1. Having a project open, click the **Select Rectangular Subset** button on the **Tools** toolbar. Go to the project view and drag a rectangle.



Define Rectangular Subset

2. Click the **Save Subset to Workspace** button on the **Tools** toolbar or right-click inside the project view and choose **Save Subset to Workspace** from the context menu. The **Save Subset to Workspace** dialog box opens.

Save Subset to Workspace

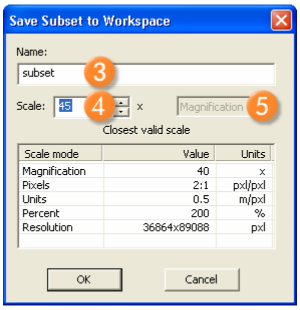


Figure 325: Save Subset to Workspace dialog box.

3. Edit the **Name** 3 of the subset.

4. Keep the **Scale** 4 of the scene for the subset or change it. You can select a different scale compared to the currently open project. That way you can work on the subset at a different magnification/resolution than the currently open project. To consider the current magnification/resolution, check the **Scale** column in right pane of the **Workspace** window. If you enter an invalid **Scale**, it will be changed to the closest valid scale and displayed in the table below.
   Reconfirm with **OK** and in the **Workspace** window a new project item appears within a **subsets** folder that is inside of the folder corresponding to the scale, named e.g. **100%**.

→ About Scales on page 49

5. Optionally, repeat the previous steps to define more subsets.

> **Note**
>
> The display of the current scale mode 5 cannot be modified in this dialog box.

> ***Tip***
>
> *Subsets can be further subdivided into tiles or smaller subsets.*

Tip

Automate Subset Creation by Using Subroutines

For automated image analysis we recommend developing rule sets that handle subset creation automatically.

→ ***Copy of a Scene Subset*** *on page 303*

→ ***Rescaled Copy of a Scene Subset*** *on page 304*

10.8.1.1 Define a Polygonal Scene Subset

Define non-rectangular regions of interest in a scene and store them as separate projects in the workspace.

Compared to defining rectangular scene subsets, polygonal subsets can be more accurate. This may be preferable for manual user selection of regions of interest in scenes.

In addition to the subset project in the workspace, a shapefile (extension **.shp**) is exported to the predefined workspace output folder named **aux_images**.

→ Create a New Workspace on page 57


Define Polygonal Subset

1. Having a project open, click the **Select Polygonal Subset** button on the **Tools** toolbar. Go to the project view and draw a polygon, for example around the regions of interest.Close it by double-clicking or right-clicking and selecting **Close Polygon** from the context menu. Note, that a polygonal subset will be saved as a rectangle which completely covers the selected polygon.


Save Subset to Workspace

2. Click the **Save Subset to Workspace** button on the **Tools** toolbar or right-click inside the project view and choose **Save Subset to Workspace** from the context menu. The **Save Subset to Workspace** dialog box opens.

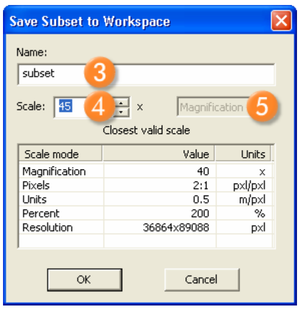


Figure 326: Save Subset to Workspace dialog box.

3. If you do not edit the **Name** 3 of the subset, auto-naming will be applied according to this pattern: **Subset** *Increasing number*.

→ About Scales on page 49

4. Keep the **Scale** 4 of the scene for the subset or change it. You can select a different scale compared to the currently open project. That way you can work on the subset at a different magnification/resolution than the currently open project. To consider the current magnification/resolution, check the **Scale** column in right pane of the **Workspace** window. If you enter an invalid **Scale**, it will be changed to the closest valid scale and displayed in the table below.
Reconfirm with **OK** and in the **Workspace** window a new project item appears within a **subsets** folder that is inside of the folder corresponding to the scale, named for example, **100%**.

In addition to the subset project in the workspace, a shapefile (extension **.shp**) is exported to the predefined workspace output folder named **aux_images**. These shapefiles named automatically and not by the setting of the **Name** 3 text box.

5. Optionally, repeat the previous steps to define more subsets.

Note

The display of the current scale mode 5 cannot be modified in this dialog box.

Tip

Subsets can be further subdivided into tiles or smaller subsets.

10.8.2 Work at Different Scales

Work on different scales, sometimes referred to as magnifications/resolutions, to allow a higher performance when analyzing large projects.

The basic idea is to perform your analysis at a low scale in situations where the image information and detail is sufficient to do so. You can start an image analysis on a copy of the scene at a low scale to identify of structures and regions of interest. All further image analyses can then be done at higher scales. For each region of interest (ROI), a new subset project of the scene is created at a higher scale. The final detailed image analysis, such as structure extraction and quantification takes place on those subset projects with regions of interest at higher scale.

Compared to regular processing, the performance is improved because the image analysis is divided into sections consisting of:

- a rough overview analysis on a smaller image at a lower scale and
- a detailed analysis at high scale, but only on subsets of the entire project.

The multi-scales workflow may follow this scheme:

1. Create a rescaled copy of the scene at a lower scale to perform an overview analysis. This scene copy is administrated as new project.
2. Analyze the rescaled copy of the project at low scale.
3. Look for regions of interest (ROI). To allow a faster detailed image analysis, select regions of interest and create a rescaled subset copy of the scene at a higher scale.
4. Analyze the rescaled subset of the scene at high scale.

→ **Create a Rescaled Scene Copy at Another Magnification/Resolution** on page 374

→ **Define a Scene Subset** on page 371

Tip

Automate Rescaling by Using Subroutines

For automated image analysis we recommend developing rule sets that handle rescaling automatically.

→ ***Rescaled Scene Copy*** *on page 303*

→ ***Rescaled Copy of a Scene Subset*** *on page 304*

10.8.2.1 Create a Rescaled Scene Copy at Another Magnification/Resolution

Create a rescaled copy of a scene at another magnification/resolution which will be represented as a new project, also called a scene copy. If you select a lower scale, this improves the speed of image analysis.

As a scene is the combined input data from a project or a workspace, scene copies do not include any results like image objects, classes or variables.

1. To create a copy of a scene at another scale, select a project in the right pane of the **Workspace** window.

2. Right-click it and on the context menu select **Create Copy with Scale** on the context menu. The **Create Scene Copy with Scale** dialog box opens.

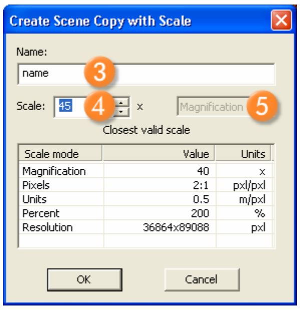


Figure 327: Create Scene Copy with Scale dialog box.

3. Edit the **Name** 3 of the subset. The default name is the same as the selected project name.

4. Keep the **Scale** 4 of the scene for the copy or change it. You can select a different scale compared to the currently selected project. That way you can work on the scene copy at a different magnification/resolution. To consider the current magnification/resolution, check the **Scale** column in right pane of the **Workspace** window. If you enter an invalid **Scale** factor, it will be changed to the closest valid scale and displayed in the table below.
   Reconfirm with **OK** and in the **Workspace** window a new project item appears within the folder corresponding to the scale, named e.g. **100%**.

→ About Scales on page 49

Note

The display of the current scale mode 5 cannot be modified in this dialog box.

Option

5. Click the **Image View or Project Pixel View** button on the **View Settings** toolbar to display of the scene at the original scale. Switch between the display of the scene at the original scale (button activated) and the rescaled magnification/resolution (button released).

Image View or Project Pixel View

Tip

Automate Creation of Scene Copies by Using Subroutines

For automated image analysis we recommend developing rule sets that handle the creation of scene copies automatically.

→ ***Rescaled Scene Copy*** *on page 303*

10.8.3 Tile and Stitch Large Scenes

Use tiling and stitching with images within the size limit (2^{31} or 46.340 x 46.340 pixels). The tiles will later be stitched back together to provide you with the results of the entire scene project.

Tiling and stitching large projects follows this sequence:

1. Creating tiles splits a scene into multiple tiles of the same size which will be administrated as new tile projects, also called tiled copy.
2. The tile projects are analyzed separately.
3. The results of the tile projects are stitched together to form the result of the entire project. Subsequent postprocessing is recommended.

→ **Create Tiles** on page 376

→ **Start Automated Image Analysis** on page 363

→ **Stitch Tile Results** on page 376

Note

Processing Performance and Tiling

Keep in mind that processing of 100 000 tiles and more will be much slower than 50 000 tiles or less. Therefore, if you tend to improve the performance, it makes no sense to have lots of very small tiles.

Find Out More

Postprocessing After Stitching

For processing large scenes using the tiling and stitching method we recommend using two specific rule sets. The first rule set is started after creating tiles. The second rule set, called ***Postprocessing****, can be started directly after stitching the analyzed tile results. It compensates border effects which may occur in the tiling and stitching method.*

The first rule set should be used to classify small objects, which rarely spread across tiles. The second rule set will usually be used to classify bigger objects of interest, which most likely spread over several tiles, and to export statistics about the complete image.

Tip

Automate Tiling and Stitching by Using Subroutines

For automated image analysis we recommend developing rule sets that handle tiling and stitching automatically.

→ ***Tiled Copy*** *on page 304*

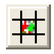
10.8.3.1 Create Tiles

Define the size of tiles that you are going to create from scenes.

As a scene is the combined input data from a project or a workspace, tiles do not include any results like image objects, classes or variables. Each tile will be stored in a separate tile project.

1. Go to the right pane of the **Workspace** window. Do one of the following:
   - Select a single **Project** that you want to tile.
   - Select a **Folder** that contains projects with scenes you want to tile.
2. To open the **Create Tiles** dialog box, do one of the following:
   - Choose **Analysis** > **Create Tiles** on the main menu bar.
   - Right-click and select **Create Tiles** on the context menu.

   The **Create Tiles** dialog box opens.


Create Tiles

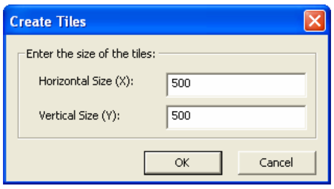


Figure 328: Create Tiles dialog box.

3. Enter the size of the tiles to be created from the project. The unit of the values is the display unit of the project.
4. Click **OK** to create tiles. For each project to be tiled a new **tiles** folder will be created. It contains the created tile projects named **tile***number*.

> **Note**
>
> Tiles can be further subdivided into subsets but not into smaller tiles.
>
> You cannot tile a scene twice. To create a second set of tiles from one scene, you have to create a scene copy before.

10.8.3.2 Analyze Tiles

Analyze the individual tiles with your rule set.

For details, see the **Start Image Analysis** chapter.

→ **Start Automated Image Analysis** on page 363

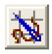
10.8.3.3 Stitch Tile Results

The **Stitch Tile Results** dialog box is used to stitch together processed tile results of one or multiple large scenes to create results of the projects with the complete scenes. Keep in mind that you cannot stitch tiles created from scenes beyond the size limit (2^{31} or 46.340 x 46.340 pixels).

1. Go to the **Workspace** window. Select a project with a scene from which you created tiles. Make sure that these tiles have been analyzed; that means they are in the state **Processed**.

2. To open the **Stitch Tile Results** dialog box, choose **Analysis > Stitch Projects** on the main menu bar.

   The **Stitch Tile Results** dialog box opens.


Stitch Tiles

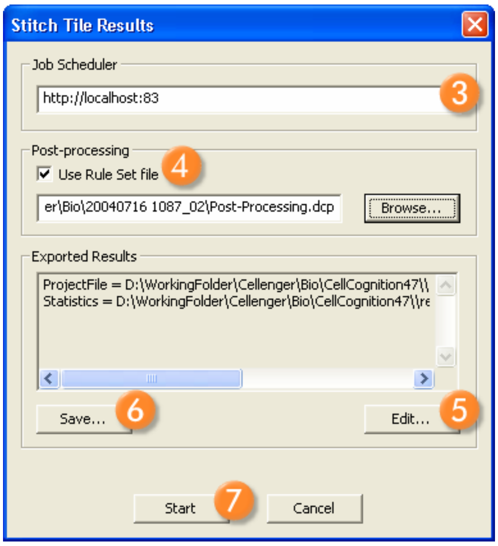


Figure 329: Stitch Tile Results dialog box.

3. You can change the **Job Scheduler** 3 which is set to **http://localhost:8184** by default.
4. Select **Postprocessing** 4 to run a second rule set immediately after stitching. This rule set is recommended for compensating border effects which may occur in the tiling and stitching method. Click **Browse** to load a rule set file. This can be a process file, which you can identify by the file name extension **.dcp**.

→ Tile and Stitch Large Scenes on page 375

5. Press **Edit** 5 to configure the exported results and the export paths of the image analysis job in an export template.

→ Configure Exported Results on page 436

6. Click **Save** 6 to store the export template with the process file.
7. Click **Start** 7 to stitch the tiles together and, if selected, to postprocess the analysis result.
8. While the image analysis is running, the **State** of the main project from which the tiles have been created will change to **stitching** and finally to **Stitched (postprocessed)**.

10.8.4 Tile Large Scenes

Use tiling if the image size is beyond the limit for stitching which is 2^{31} or 46.340 x 46.340 pixels. You can load an image of arbitrary size into your workspace and then define the

→ Create Tiles on page 376

size of tiles that you are going to create. The generated tiles can be used for image analysis, however due to the size of the original image the tiles results cannot be stitched together to form the result of the project with the complete scene.

Tip

Automate Tiling by Using Subroutines

For automated image analysis we recommend developing rule sets that handle tiling automatically.

→ ***Tiled Copy*** *on page 304*

11 Examine Results

Contents in This Chapter

After processing image data you can examine the results, thus allowing you to evaluate the used rule set.

Various methods enable you to explore and evaluate the image analysis results. In addition, you may want to use manual editing tools to influence the results by hand.

11.1 Explore the Image Analysis Results

View and investigate the results of an image analysis.

Results of an image analysis can be both numerical and visual. Numerical results are statistics and result values. Visual results are represented by the classification of image objects.

You can view statistics and compare numerical result values of projects. Depending on the analysis rule set, certain export functionalities enable the calculation of result values in relation to both single projects and large data sets of a workspaces.

Additionally, you can open a project to view and investigate it. A number of viewing options enable various visualizations of analyzed projects. Various display options enable in-depth exploration of both the image object hierarchy and image objects.

11.1.1 View Result Values in the Workspace

View and compare numerical result values of projects.

Precondition: To view results, you need a workspace containing at least one analyzed project.

1. Go to the right pane of the **Workspace** window
2. Next to every listed project, you can see the project **State**, the **Time** and the exported result values. The **Time** column shows the processing time for each project. Numerical result values, which depend on the used rule set, are listed in their respective columns. The **Remarks** column denotes processing information or malfunction.

→ Change the View in the Workspace on page 80

3. By default, exported numerical result values are saved in comma-separated files (extension **.csv**) into a **results** folder that could be found in the same directory as the workspace file. You can open **.csv** files with common spreadsheet software like **Microsoft® Excel**.

   The export settings can be modified.

→ Configure Exported Results on page 436

11.1.2 Open a Project to View the Classification

Open a project to view and visually investigate the classification of image objects. There are different commands for opening projects, depending on the way a project is stored.

→ View a Project on page 84

11.1.3 View Legend of Classes

The **Legend** displays all classes of the currently displayed project and their respective color.

To open the **Legend** window, do one of the following:

- Choose **Classification** > **Legend** from the menu bar
- Choose **View** > **Window** > **Legend** from the menu bar

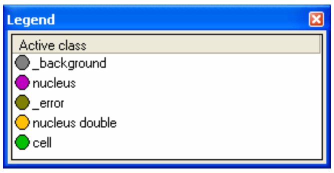


Figure 330: Legend window listing classes.

If necessary, switch to the **View Classification** mode.


View Classification

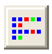
→ View Classification on page 109

11.1.4 View Thumbnails in the Heat Map Window

View thumbnails of workspace scenes and tiles in the **Heat Map** window.

Available only for Definiens eCognition **Life** Server

Thumbnails display by default when a workspace scene or a tiles is opened and the **Heat Map** window is open.


Heat Map

1. Choose **View** > **Windows** > **Heat Map** on the main menu bar.
2. If scenes are not already displayed in the **Heat Map** window, click on a folder in the **Workspace** to open it and view the corresponding scenes in the **Heat Map** window.
   - The folder name displays in the **Heat Map** window along with the number of groups and the number of total scenes in the groups. The subfolders display as separate groups.
   - The folder name displays in the **Heat Map** window along with the number of groups and the number of total scenes in the groups. The subfolders display as separate groups. The thumbnails display in pixel view, with the number of scenes and groups displayed at the top of the window.
   - If the images have been analyzed, statistics can be viewed in the form of a heat map. If they have not been analyzed, the heat map will be gray.

3. To change the display of the thumbnails, select **Thumbnail Settings** in the context menu or go to the main menu bar and select **View > Windows > Thumbnail Settings.**

→ **Edit Image Layer Mixing for Thumbnails** on page 98

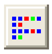
4. Click a thumbnail to display identifying information in the **Selected Image** field at the bottom of the window or move the cursor over a thumbnail to view an identifying tooltip. The corresponding project will be highlighted in the right-hand pane of the **Workspace** window.
5. Double-click a thumbnail to open it in the **Project View.**
6. To change the size of the thumbnails, use the drop-down list at the bottom of the **Heat Map** window to select **Small, Medium** or **Large Squares.**
7. To save an image of the thumbnails as a file, click the **Export** button at the bottom of the **Heat Map** window, select a file format and location and click **Save.**
8. If the selected scene includes tiles, you can click the **Show Tiles** button to display all tiles in the selected scene in the **Heat Map** window. The **Show Tiles** button is not available if there are no tiles in the selected scene.
   - Click a tile thumbnail to display the tile name in the **Selected Image** field and highlight the corresponding tile in the right side of the **Workspace** window.
   - Double-click a tile thumbnail to open it in the **Project View** and highlight it in the right side of the **Workspace** window.
   - Click the **Show Group** button to return to a display of the scene thumbnails.
9. Use the **Display** drop-down list to view scene or tile statistics.

11.1.4.1 Compare Scene Statistics in the Group Heat Map

Use the **Heat Map** window to compare scene statistics.

You must have an analyzed workspace open in order to use the **Heat Map** window.

Available only for Definiens eCognition **Life** Server


Heat Map

1. To view a heat map for result values of groups of scenes choose **View > Windows > Heat Map** on the main menu bar.
2. Click the drop-down arrow next to the **Display** field and select a result.
   Result values will display as colors in the **Heat Map** window for each scene in the group. The features displayed are the features that have been configured for export. The range of colors is from blue to red, with blue being the smallest value and red the highest.

→ **Configure Exported Results** on page 436

3. To change the range of values for the heat map, enter numbers in the fields at either end of the color range at the bottom of the window or use the up and down arrows.
4. Select another result and the heat map will update automatically.
5. If the selected scene is tiled, you can click the **Show Tiles** button to display a tiled view all tiles and also view result values for tiles in the heat map.

11.1.4.2 Compare Tile Statistics in the Heat Map

View tiles in the **Heat Map** window and compare result statistics among tiles.

Available only for Definiens eCognition **Life** Server

Precondition: You must have an analyzed workspace open in order to use the **Heat Map** window.

Heat Map

1. To open the **Heat Map** window, choose **View** > **Windows** > **Heat Map** on the main menu bar.
2. Click the **Show Tiles** button to open tiles in the **Heat Map** window. If there are no tiles in the project, the **Show Tiles** button is not available.
   The folder name displays in the **Heat Map** window along with the number of tiles
   The **Show Tiles** button becomes the **Show Group** button.
3. Use the **Display** drop-down list to select a result to display in the heat map across the entire tile display.

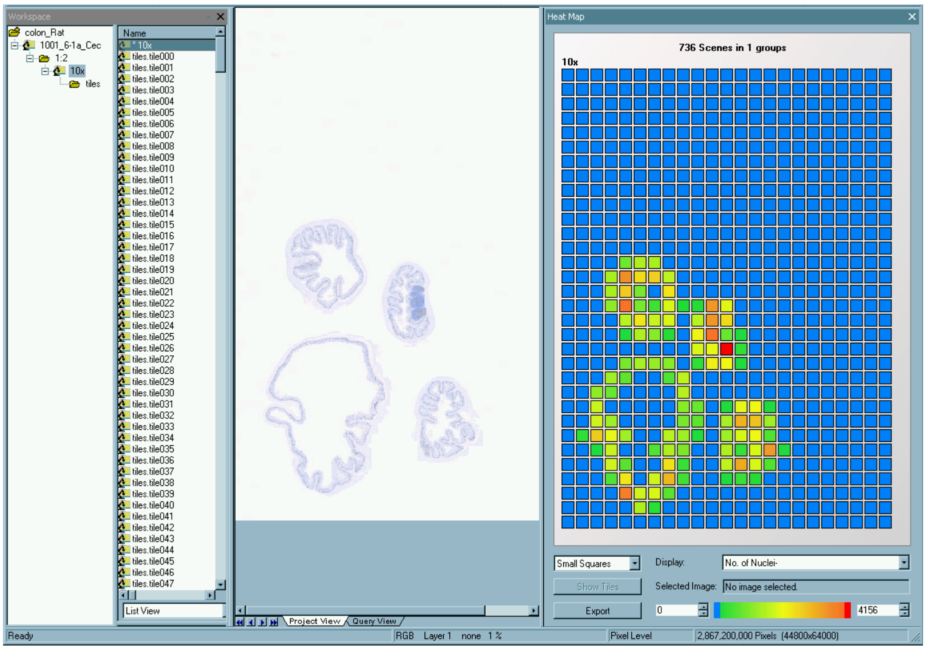


Figure 331: Heat Map window with heat map displayed across tiles, with the corresponding project view and workspace list view. (Image data courtesy Novartis Pharma AG; Pathology/Toxicology EU, E. Persohn.)

4. To adjust the distribution of the colors, enter the desired values for minimum or maximum at either side of the scale or use the arrow buttons to count up or down.
5. Click the **Show Group** button to return display scenes in groups.

11.1.5 Access Results Via the Plate View

Available only for Definiens eCognition **Life** Server

For automated image analysis in cellular biology, you commonly use plates. Definiens Developer provides specific controls and tools to develop ruleware for cellular image analysis.

For easy-to-use image analysis, we recommend to use the **Definiens Architect** in combination with **Definiens Cellenger** application.

Plate View

To open the **Plate View** window, choose **View** > **Windows** > **Plate View** on the main menu bar.

11.1.5.1 View Results in the Plate View

Available only for Definiens eCognition **Life** Server

→ **Manage Data in the Plate View** on page 90

The **Results** tab of **Plate View** window provides graphical information about the results of the analysis for each well. Statistical distribution is indicated by colors. The range varies from blue to red with blue indicating a low value for the selected parameter and red indicating a high value.

- Blue = Minimum value
- Green = Lower half
- Yellow = Mean value
- Orange = Upper half
- Red = Maximum value.

Results tab

To open the **Results** tab of the **Plate View** window, click the **Results** button.

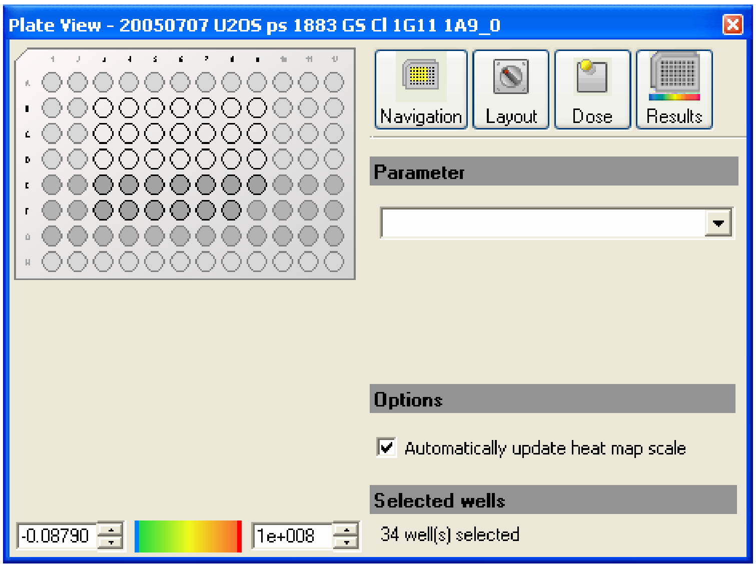


Figure 332: Analysis Results in the Plate View window with automatic scaling option activated.

1. Select a parameter from the **Parameter** list. The list contains parameters that were processed during the analysis.
2. The heat map scale marks the minimum (left, blue color) and maximum value (right, red color) of the selected parameter.
3. Select **Automatically update heat map scale** to automatically adjust the minimum and maximum value to the distribution of the selected parameter.

4. To adjust the distribution of the colors manually, enter the desired values for minimum or maximum at either side of the scale or use the arrow buttons to count up or down.

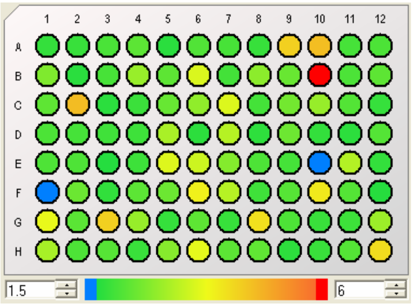


Figure 333: Analysis Results in the Plate View window with manual scaling settings.

On a broader scale, the same data brings different color coding for the distribution.

5. To open the image of a specific well double-click it.

Diagrams such as the **Graph** window or the **2D Scatter Plot** window provide an alternative view on the statistical distribution.

You can use the **Dose Response**e window in conjunction with the **Plate View** window to generate a dose response curve.

→ View Results in a Dose Response Curve on page 385

11.1.5.2 View Result Values in the Graph

The **Graph** window is an alternative view on the statistical values of a certain feature for selected wells. It gives you information across projects.

Available only for Definiens eCognition **Life** Server

Prerequisite: In the **Navigation** or **Result** tab of the **Plate View**, select the plate and site of the plate. In the **Results** tab, select a parameter.


Graph

1. To open the **Graph** window choose **View** > **Windows** > **Graph** from the main menu. The **Graph** window opens with an empty diagram frame.

2. In the **Plate View**, select those wells of the plate map you want to see in the diagram.

→ Manage Data in the Plate View on page 90

3. Change to the **Results** tab and select the parameter.

The graph now displays the values of the parameter for the selected wells on the y-axis. The x-axis indicates the coordinates of the selected wells.

→ Navigate Through a Plate on page 90

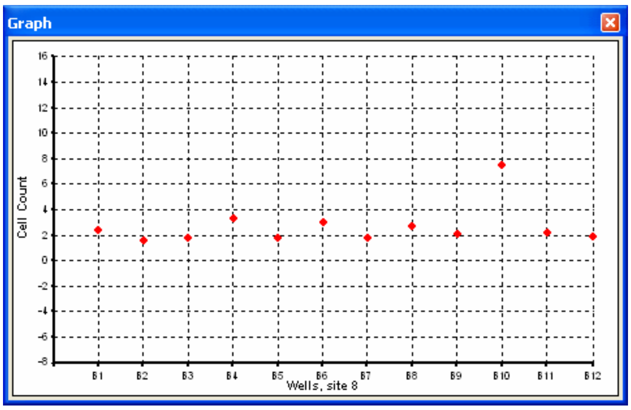


Figure 334: Graph window for selected wells.

You can change the selection of the wells or the parameters as you wish. The **Graph** simultaneously reflects your selection.

11.1.5.3 View Results in a Dose Response Curve

Use the **Dose Response** window to determine the IC50 or EC50 of a test compound. You can add compounds to plate rows and then graph the results and fit them to a curve.

Available only for Definiens eCognition **Life** Server

→ **About Dose Response Experiments and Curve Fitting** on page 389

The dose response curve can only be modeled for a 96-well plate. Dosages or concentrations are modeled across rows only and not up and down columns.

1. To open the **Dose Response** window do one of the following:
   - Choose **View** > **Windows** > **Dose Response** from the main menu.

     Dose Response

   - Select the **Result View** preset layout. Choose **View** > **Result View** from the main menu.

     Result View

   The **Graph** window opens with an empty diagram frame.
2. Input selected compounds to plate rows in a selected dosage.
3. Select a compound for which you want to see the dose response curve.
4. Select a feature for which a dose response curve is needed.
   The dose response data points are automatically placed on the graph.
5. The fitting model is provided.
6. As part of this process, you may choose to alter the initial values of the model's parameters.
7. After an initial fitting you may want to deal with outlying data points and then refit the curve.
8. You can export both the image of the curve and the related data.

Input Compounds to Plate Rows

Use the **Plate View** to add compounds to the plate rows. You can add up to eight compounds, one per row.

Available only for Definiens eCognition **Life** Server

→ **View Results in the Plate View** on page 383

→ **Define Plate Layout** on page 91

1. To add compounds to plate rows, set positive and negative controls and select any inactive rows on the **Layout** tab.
2. Click the **Dose** button to display the **Dose** tab.
   The **Dose** tab is not active unless a 96-well plate is loaded.

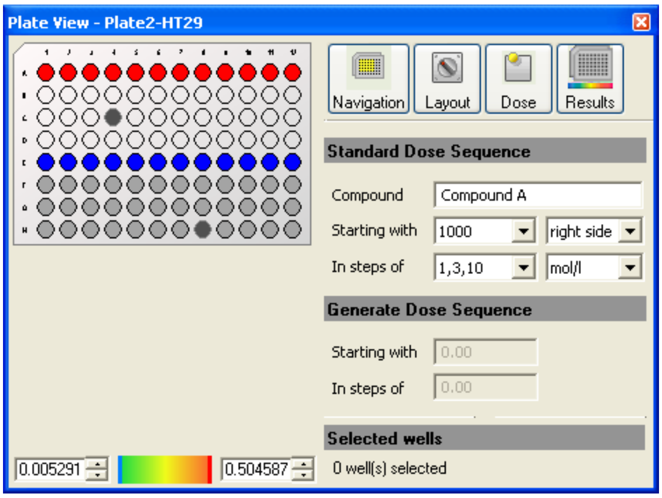


Figure 335: Plate View - Dose tab.

3. Go to **View** in the main menu bar and navigate to **Windows > Dose Response Curve** to open the **Dose Response** window, with the **Results** tab displayed.

→ Fit a Dose Response Curve on page 387

4. Select the rows in the **Plate View** to which you want to apply compounds.
   The number of wells selected displays under **Selected wells.**
   You can select up to eight adjacent rows.
5. In the **Standard Dose Sequence** group box of the **Dose** tab, the value **Compound A** is the starting default value in the **Compound** field.
   You can overwrite the defaults names.
6. Use the drop-down arrows next to **Starting with** to choose a starting concentration and a unit; **mol/l, mmol/l** or **ng/ml.**
   You can either choose a concentration or type in your own choice. This will be the highest concentration applied.
   Use the right-hand drop-down list to choose to begin dilution from the **right** side or the **left** side of the plate.

7. Use the drop-down arrow for the **In steps of** field to choose logarithms of the displayed values for concentration steps.
   - **...1, 3, 10, ...**
   - **...1, 2, 5, 10, ...**
8. Choose **generate** to activate the **Dose Sequence** group box and enter a sequence of your choice.
   - Enter a starting dosage in the **Starting with** field.
   - Enter the step size in the **In steps of** field.
9. You can use the **Save** button on the **Layout** tab to save your dose sequence to file and the **Load** button to use a previously saved dose sequence.
10. Click the **Add** button to begin adding the selected compound to the selected rows. Each compound is automatically displayed in the **Compound** field on the **Parameters** tab of the **Dose Response** window as it is added.

Display Dose Response Data Points

Select a feature and a compound for the dose response to display data points on the graph in the **Dose Response** window.

Available only for Definiens eCognition **Life** Server

Once you display the data points, you can prepare to fit a dose response curve.

→ Fit a Dose Response Curve on page 387

→ Input Compounds to Plate Rows on page 385

1. To select a compound for which you want to see a dosage response curve, select a compound in drop-down list of the **Compound** field of the **Dose Response** window.
   If the window is not already open, open it by selecting **View > Windows > Dose Response Curve** in the main menu bar.
   The compounds available have already been added to plate rows.
2. Select a feature for which you want to display a dose response by clicking the **Results** button in the **Plate View** and making a selection from the drop-down list for the **Parameter** field.
   The data points for the rows where the compound was added display on the graph in the **Dose Response** window.

→ View Results in the Plate View on page 383

Fit a Dose Response Curve

Select a fitting model and assay type for the dose response curve, and then fit the data to the curve.

Available only for Definiens eCognition **Life** Server

As options, you can choose initial values for the curve and/or handle outlying values before fitting the curve, or at any later time, and then fit the curve again.

To fit a dose response curve to the data points:

1. The provided model in the **Model** field of the **Dose Response** window is **4- parameter logistic.**
2. Select the type of assay:
   - **EC50** (concentration of half-maximal effect) has a curve with a positive slope.
   - **IC50** (concentration of half-maximal inhibition) has a curve with a negative slope.

→ About Dose Response Experiments and Curve Fitting on page 389

3. The dose response curve can now be fit to the data by clicking the **Fit** button in the **Parameters** tab of the **Dose Response** window.

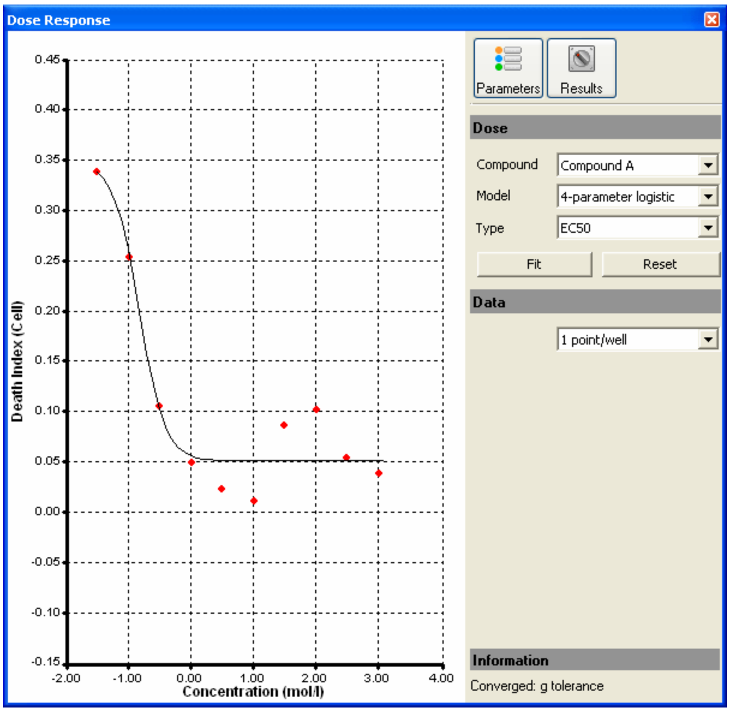
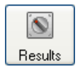


Figure 336: Dose Repsonse Window - Parameters Tab.

The results include the following:

- The dose response curve displays in the graph of the **Dose Response** window. The system recognizes a suspicious fit and displays the curve in red if the slope is greater than or equal to 3 or less than or equal to -3.
- Estimates of the parameters and error estimates display on the **Results** tab of the **Dose Response** window.



Results

4. To view or change the parameters, go to the **Results** tab of the **Dose Response** window. You can change the following values by entering them in the respective fields.

- **EC50/IC50**: The result returned is the concentration that produces the median response..
- Slope: The result returned is the slope. You can enter a value to change it.
  The default value is **+ 1.00** or - **-1.00** depending on the value selected in the **Type** field.
  In general, the default should be used for the first curve fitting.
  The slope should be in the range of -5.00 to +5.00.
- **Max:** The result returned is the highest Y value. You can enter a value to change it.
- **Min:** The result returned is the lowest Y value. You can enter a value to change it.

- After changing values, you can click the **Parameters** button and click **Fit** again to refit the curve.

- To restore the default values, click the **Reset** button on the **Parameters** tab.

5. Use the **Data** drop-down list in the **Parameters** tab to adjust the way data displays. You can display either one point per well or one point per column.

6. There may be data points in the graph that represent invalid values. These data points can be inactivated so that they do not influence the curve.

   - To inactivate points, click them in the graph.
     The inactive points will display as gray points.

   - To reactivate the points, click them again.

7. You can use the context menu to copy an image of the dose response curve to the clipboard.

8. To export the values of the curve as a **.csv** file., click the **Results** button in the **Dose Response** window and click the **Export** button.
   Data exported includes the parameters discussed in step 4 above and also Pearson's correlation coefficient. **(R_Squ)** and the sum of the squared error **(SSE).**

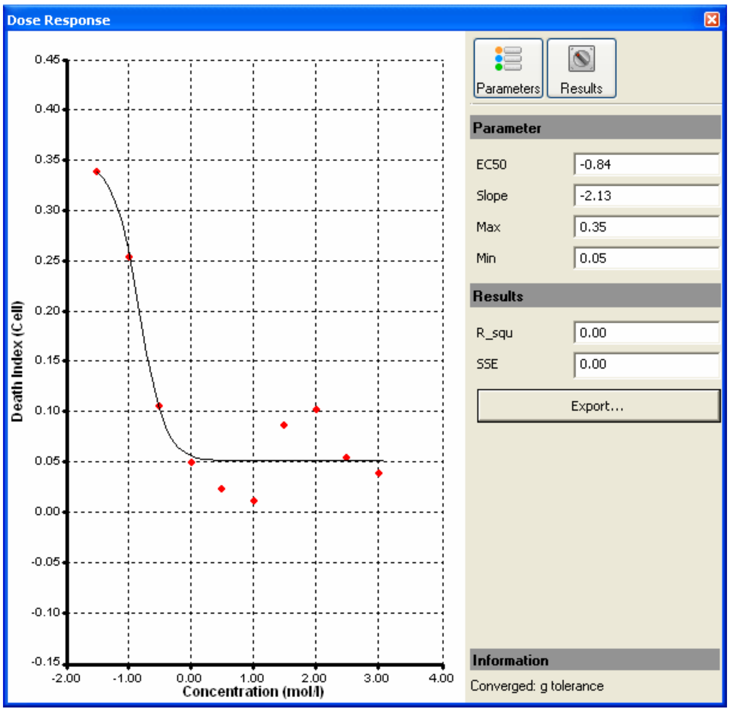


Figure 337: Dose Response Curve Window - Results Tab.

About Dose Response Experiments and Curve Fitting

A dose response curve (DRC) or concentration-response curve describes the relationship between the amount or concentration of a drug or toxin administered to an organism, and the response of the organism to that drug. A dose response curve (DRC) or concentration-response curve describes the relationship between the amount or concentration of a drug or toxin administered to an organism, and the response of the organism to that drug.

Available only for Definiens eCognition **Life** Server

Many biological systems in which a univariate (one parameter of interest) response can be measured show a typical relationship between concentration of the agonist or antagonist and the magnitude of the response.

This concentration response, or dose response relationship, derives from the law of mass action. At sufficiently low concentration, no effect is seen. As the concentration is increased above a threshold concentration, an increasing effect is observed. Finally, at high concentrations (orders of magnitude higher than the threshold concentration), the response saturates; it does not increase any further. When the response is plotted on a linear vertical scale versus the concentration on a logarithmic horizontal scale, a sigmoidal (S-shaped) relationship is apparent.

The characteristic value of a sigmoidal relationship is the half-maximal response, in other words, the point at which the sigmoidal curve is halfway from the lowest value to the highest value. This also corresponds to the inflection point of the curve. For concentration-response curves, this point corresponds to the concentration at which the response is halfway between baseline and saturation.

For inhibitory systems, in which the response decreases with increasing concentration of the test substance, this point is referred to as the IC_{50} (concentration of half-maximal inhibition). For systems in which the response increases with increasing concentration, this point is referred to as the EC_{50} (concentration of half-maximal effect). Because this concentration is a characteristic of the compound for the biological system being tested, the goal of such an experiment is to determine the best estimate of this value.

The most common model that is fitted to such data is the four-parameter logistic curve, or 4PL model. The model has four parameters:

- Minimum response.
- Maximum response.
- Concentration at half-maximal response (halfway from minimum to maximum).
- Slope of the curve at the half-maximal concentration.
- Fitting the model to the curve, in other words, minimizing the sum of squared errors between the model and the data points, results in the best estimate of the four parameters. In general, the only parameter of interest is the concentration at half-maximal response (IC50 or EC50).

The formula for the four-parameter logistic curve follows, expressed in two slightly different ways. X is the log of the concentration.

$$
\begin{aligned}
Y &= Bottom + \frac{(Top - Bottom)}{1 + \left(\frac{10^{LogEC_{50}}}{10^{x}}\right)^{Slope}} \\
&= Bottom + \frac{(Top - Bottom)}{1 + 10^{(LogEC_{50} - X)Slope}}
\end{aligned}
$$

11.1.6 Accept and Reject Results

To record your evaluation of a project, you can mark it as **Accepted** or **Rejected**.

To mark a project, select the **Accept Project** button or the **Reject Project** button on the **File** toolbar.

Accept Project

Reject Project

To reset a project to a previous state (e.g. **Processed**), use the **Roll Back** function in the project **History**.

→ Inspect the History of a Project on page 85

11.2 Manual Editing

Manual editing of image objects and thematic objects allows you to manually influence the result of an image analysis. The main manual editing tools are **Merge Objects Manually**, **Classify Image Objects Manually** and **Cut an Object Manually**.

While manual editing is not commonly used in automated image analysis, it can be applied to highlight or reclassify certain objects or to quickly improve the analysis result without adjusting the applied rule set.

To open the **Manual Editing** toolbar choose **View** > **Toolbars** > **Manual Editing** on the main menu.

11.2.1 Change Editing Mode

The **Change Editing Mode** drop-down list on the **Manual Editing** toolbar is set to **Image Object Editing** by default. If you work with thematic layers and want to edit them by hand, choose **Thematic editing** from the drop-down list.

→ Manage Thematic Objects on page 396
→ Working with Thematic Layers on page 283
→ View Thematic Layer Attributes on page 286

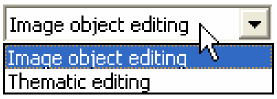


Figure 338: Change Editing Mode dropdown list.

11.2.2 Selection Tools

Objects to be fused or classified can be selected from the **Manual Editing** toolbar in one of the following ways:

- **Single Selection Mode** selects one object.
  Select the object with a single click.


Single Selection Mode

→ Select and Deselect Image Objects on page 115

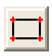
- **Polygon Selection** selects all objects that lie within or touch the border of a polygon.
  Set vertices of the polygon with a single click. Right-click and choose **Close Polygon** to close the polygon.


Polygon Selection

- **Line Selection** selects all objects along a line.
  Set vertices of the line with a single click. A line can also be closed to form a polygon by right-clicking and choosing **Close Polygon**. All objects that touch the line are selected.


Line Selection

- **Rectangle Selection** selects all objects within or touching the border of a rectangle.
  Drag a rectangle to select the image objects.


Rectangle Selection

To select only those objects that belong to certain classes, open the **Edit Classification Filter** dialog box by pressing the **Filter Classes for Multiple Image Object Selection** button. Choose the classes of interest. Any selection you now make with a selection tool will highlight only the image objects of your selected classes.

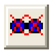

Filter Classes for Multiple Image Object Selection

All manual editing buttons can be deactivated in one of the following ways:

- Toggle the button on or off.
- Deselect the respective button under **Tools > Manual Editing** on the menu bar.
- Click any other manual editing button to activate a different manual editing mode.
- Choose **Tools > Manual Editing > Single Selection** from the main menu bar.
- Click the **Single Selection Mode** button on the **Manual Editing** toolbar.

To clear a selection, do one of the following:

- Press any selection tool button.
- Press the **Clear Selection for Manual Object Merging** button.
- Make a new selection.
- Press the Esc button to clear a rectangle or line selection.


Clear Selection for Manual Object Merging

11.2.3 Merge Objects Manually

The manual editing tool **Merge Objects** is used to manually merge selected neighboring image or thematic objects.


Merge Objects Manually

> **Note**
>
> Manual object merging operates only on the current image object level.

If you want to merge neighboring objects into a new single object, choose **Tools > Manual Editing > Merge Objects** from the main menu bar or press the **Merge Objects Manually** button on the **Manual Editing** toolbar to activate the input mode.

Select the neighboring objects to be combined with single mouse-clicks. Selected objects are displayed in the selection color which is red by default. The selection color can be changed under **View > Display Mode > Edit Highlight Colors**.

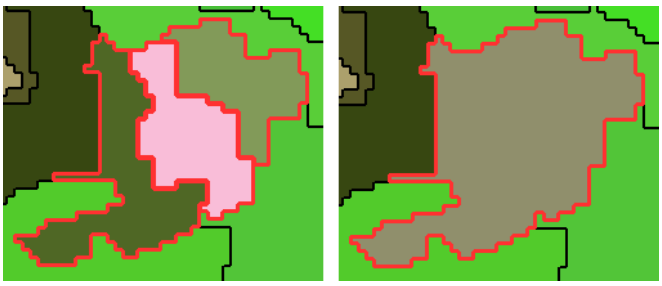
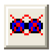


Figure 339: Left: selected image objects. Right: merged image objects.

If an object cannot be merged with the already selected one it cannot be activated because it belongs to a different superobject.

> **Note**
>
> Due to the hierarchical organization of the image objects, an object cannot have two superobjects. This also limits the possibilities for manual object merging, since two neighboring objects cannot be merged if they belong to two different superobjects.

- To clear a selection, click the **Clear Selection for Manual Object Merging** button or deselect individual objects with a single mouse-click.


Clear Selection for Manual Object Merging

There are two options to merge selected objects:

- Click the **Merge Selected Objects** button on the **Manual Editing** toolbar.
- Right-click in the project view and choose **Merge Selection** on the context menu.


Merge Selected Objects


Single Selection Mode

11.2.3.1 Merge Thematic Object Based on Image Object

The **Merge Thematic Object Based on Image Object** tool allows you to fuse a thematic object with an image object.


Merge Thematic Object Based on Image Object

Activate the **Merge Thematic Object Based on Image Object** button. Select a thematic object and an adjoining image object. Right-click and choose **Merge to Polygon** to create a new polygon with the outlines of the thematic and the image object as border.

The image object itself will remain unchanged.

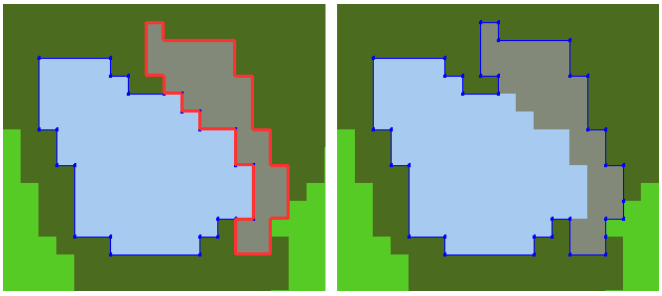


Figure 340: Merge thematic object based on image object.
Left: A thematic object (blue outlines) and a neighboring image object (red outlines) have been selected.
Right: Result of merging the two objects. A new thematic polygon has been created; the image object has not changed.

11.2.4 Classify Image Objects Manually

The manual editing tool **Classify Image Objects** allows easy class assignment of selected image objects.

Manual image object classification can be used for the following purposes:

- Manual correction of previous classification results including classification of previously unclassified objects.
- Classification without rule sets (in case the creation of an appropriate rule set is more time-consuming), using the initial segmentation run for automated digitizing.

Precondition: To classify image objects manually, the project has to contain at least one image object level and one class in the **Class Hierarchy**.

To perform a manual classification, do one of the following:

- Choose **Tools** > **Manual Editing** > **Classify Image Objects** from the menu bar.
- Click the **Classify Image Objects** button on the **Manual Editing** toolbar to


Classify Image Objects

activate the manual classification input mode.

Select Class for Manual Classification

In the **Select Class for Manual Classification** drop-down list box, select the class to which you want to manually assign objects. Note that selecting a class in the **Legend** window or in the **Class Hierarchy** window (if available) will not determine the class for manual editing; the class has to be selected from the before-mentioned drop-down list.

Now objects can be classified manually with a single mouse-click. To classify objects, do one of the following:

- Select the **Classify Image Objects** button and the **Class for Manual Classification**. Click the image objects to be classified.
- Select the image object(s) you want to classify first. Select the **Class for Manual Classification** and press the **Classify Image Objects** button to classify all selected objects.
- Select one or more image objects, right-click into the image object(s) and select **Classify Selection** from the context menu.

When the object is classified, it is painted in the color of the respective class.

If no class is selected, a mouse-click deletes the previous class assignment; the image object becomes unclassified.

To undo a manual classification on a previously unclassified object, simply click the object a second time. If the object was previously classified, then clicking again does not restore the former classification; instead, the object becomes unclassified.

11.2.4.1 Delete Manually Classified Image Objects

1. To delete the complete manual classification result, choose **Classification** > **Class Hierarchy** > **Delete Classification** from the main menu bar. The **Delete Classification** dialog box opens.

→ Delete a Class Hierarchy on page 255

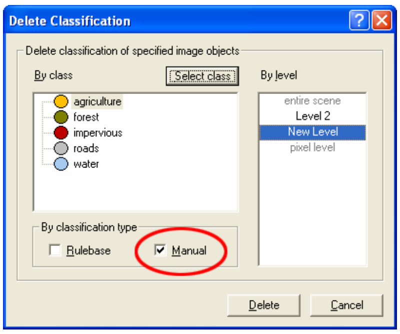


Figure 341: Delete Classification dialog box. Checking Manual deletes the Manual classification.

2. Set options to delete a classification by class, level, or classification type. Make sure that the **Manual** check box is selected in the **By classification type** group box.

> **Note**
>
> Manual classification operates only on the current image object level.
>
> Manual classification is a method for quick interactive classification without having to use a rule set. If a rule set has been processed before, a manual classification will override the existing automatic class assignment. This also means that deleting a manual classification will result in unclassified image objects.

11.2.5 Cut an Object Manually

The manual editing tool **Cut Objects** is used to cut a single image object or thematic object manually.

> **Note**
>
> The cut line needs to cut or touch the **outer** object border at least twice.
>
> **Note**
>
> Due to the hierarchical organization of the image objects, **Cut Objects Manually** cuts both the selected image object and its subobjects on lower levels.

Choose **Tools** > **Manual Editing** > **Cut Objects** from the main menu bar to activate the manual cutting input mode.


Cut Object Manually

Depending on the object's shape, the cut line can touch or cross the object's outline several times, and two or more new objects will be created. To cut an object, activate the object to be split by clicking it. Draw the cut line which can consist of several sections. To perform the cut, right-click and choose one of the following from the context menu:

- **Perform Split** to cut the object.
- **Close and Split** to close the cut line before cutting.

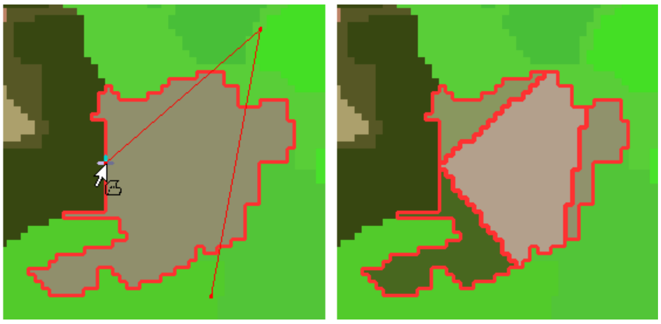


Figure 342: Cut an object manually.
Left: Choosing **Perform Split** will cut the object into three new objects, while **Close and Split** will cause the line to cross the object border once more, thus creating four new objects. Right: Result of the **Close and Split** operation.

The **Snapping Tolerance** is set in pixels. When using **Manual Cutting**, snapping facilitates touching object borders similar to magnetism.

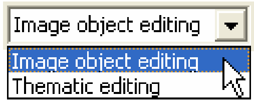
Snapping Tolerance [pxl]

11.2.6 Manage Thematic Objects

A thematic object is the basic element of a thematic layer. A thematic object can be a polygon, a line or a point. It represents positional data of a single object in the form of coordinates and describes the object by attributes.

Managing thematic objects is used to define regions of interest before image analysis. Further it is used for the verification of classifications.

1. Before managing thematic objects, go to the **Change Editing Mode** drop-down list box on the **Manual Editing** toolbar and change the editing mode to **Thematic editing**.



Change Editing Mode

→ Change Editing Mode on page 391

2. Now, you need to do one of the following:

   - Select an existing thematic layer from the **Select Thematic Layer** drop-down list box.

Select Thematic Layer

   - If the **Select Thematic Layer** drop-down list box is set on **-New Layer-**, the **Create New Thematic Layer** dialog box opens to create a new thematic layer. Select the **Type** of thematic layers:
     • **Polygon Layer**
     • **Line Layer**
     • **Point Layer**

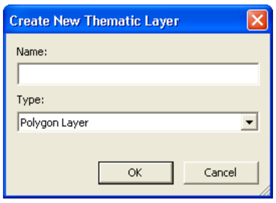


Figure 343: Create New Thematic Layer dialog box.

11.2.6.1 Generate Thematic Objects by Drawing

Create a new thematic object independent of existing image objects.

Generate Thematic Objects

Use the **Generate Thematic Objects** button to draw a new thematic point, line or polygon object. This object can touch or cross any existing image objects.

From the **Change Editing Mode** drop-down list, **Thematic editing** has to be selected.

→ Change Editing Mode on page 391

Generate Point Objects

Generate point objects on a thematic point layer in one of the following ways:

- Enter the point's X and Y coordinates in the **Generate Point** dialog box. Click **Add Point** to generate the point.
- Click in the thematic layer. The point's coordinates are displayed in the **Generate Point** dialog box.

Press **Delete Point** to delete the point whose coordinates are displayed in the **Generate Point** dialog box.

Generate Line Objects

Generate line objects on a thematic line layer by clicking in the thematic layer to set vertices. Right-click and choose **Finish line** to stop drawing.

Generate Polygon Objects

Generate polygon objects on a thematic polygon layer by right-clicking in the thematic layer to set vertices. Right-click and choose **Close polygon** to close the polygon.

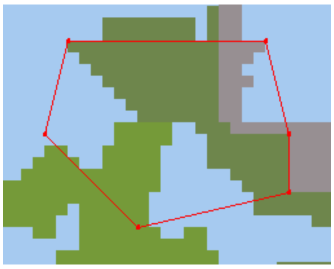


Figure 344: New thematic polygon object. The polygon borders are independent of existing image object borders.

11.2.6.2 Generate Thematic Object Based on Image Object

Generate Thematic Object Based on Image Object

With the **Generate Thematic Object Based on Image Object** button, you can add to a thematic layer a thematic object with the same outlines as an image object.

Manual thematic object generation is useful to improve a thematic layer. New thematic objects are added to the **Thematic Layer Attribute Table**. Their attributes are initially set to 0.

To generate a new thematic object, select an image object and right-click it. From the context menu, choose **Generate Polygon** to add the new object to the thematic layer.

11.2.6.3 Merge Thematic Object Based on Image Object

For details, see the related instructional section.

Merge Thematic Object Based on Image Object

→ Merge Thematic Object Based on Image Object on page 393

11.2.6.4 Delete Thematic Objects

To delete one or more thematic objects, select them and press the **Delete Selected Thematic Objects** button.

Delete Selected Thematic Objects

11.2.6.5 Save Thematic Objects in Thematic Layer

Save thematic objects in a thematic layer saved as a vector shapefile.

Save Thematic Layer As

1. Press the **Save Thematic Layer As** button on the **Manual Editing** toolbar.
2. Click **Save** to export the thematic layer as a shape file with file extension **.shp**.

Alternatively, you can use the **Export Results** dialog box to export thematic objects as a shapefile.

→ Export Results as Shape File on page 404

11.3 Accuracy Assessment

Use accuracy assessment methods to produce statistical outputs that can be used to check the quality of the classification results. Tables from statistical assessments can be saved as **.txt** files, while graphical results can be exported in raster format.

1. Open the **Accuracy Assessment** dialog box by selecting the menu item **Tools > Accuracy Assessment** from the menu bar.

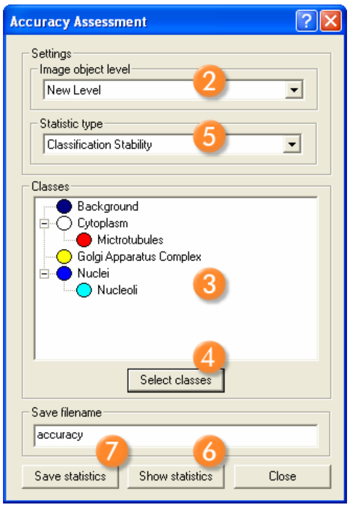
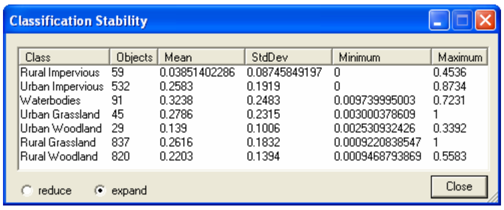


Figure 345: Accuracy Assessment dialog box.

2. A project can contain different classifications on different image object levels. Specify the image object level of interest by using the **Image object level** drop-down menu.
3. In the **Classes** window, all classes and their inheritance structure are displayed.
4. To select the classes you want to assess, click the **Select Classes** button and make a new selection in the **Select Classes for Statistic** dialog box. By default all available classes are selected. You can deselect classes through a double-click in the right frame.
5. In the **Statistic type** drop-down list, select one of the following methods for accuracy assessment:
   - Classification Stability
   - Best Classification Result
   - Error Matrix based on TTA Mask
   - Error Matrix based on Samples
6. To view the accuracy assessment results, click **Show statistics**.
7. To export the statistical output, click **Save statistics**. You can enter a file name of your choice in the **Save filename** text field. The table is saved in comma-separated ASCII **.txt** format; the extension **.txt** is attached automatically.

11.3.1 Classification Stability

The **Classification Stability** dialog box displays a statistic type used for accuracy assessment.



Class	Objects	Mean	StdDev	Minimum	Maximum
Rural Impervious	59	0.03851402286	0.08745849197	0	0.4536
Urban Impervious	532	0.2583	0.1919	0	0.8734
Waterbodies	91	0.3238	0.2483	0.009739995003	0.7231
Urban Grassland	45	0.2786	0.2315	0.003000378609	1
Urban Woodland	29	0.139	0.1006	0.002530932426	0.3392
Rural Grassland	837	0.2616	0.1832	0.0009220838547	1
Rural Woodland	820	0.2203	0.1394	0.0009468793869	0.5583

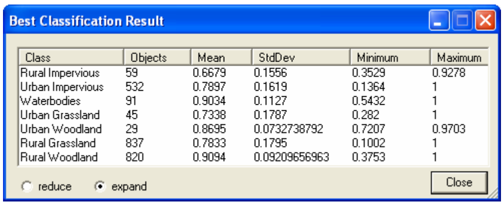
Figure 346: Output of the Classification Stability statistics.

The difference between the best and the second best class assignment is calculated as a percentage. The statistical output displays basic statistical operations (number of image objects, mean, standard deviation, minimum value and maximum value) performed on the best-to-second values per class.

> ***Tip***
>
> *To display the comparable graphical output, go to the* ***View Settings*** *window and select* ***Mode*** *>* ***Classification Stability****.*

11.3.2 Best Classification Result

The **Best Classification Result** dialog box displays a statistic type used for accuracy assessment.



Class	Objects	Mean	StdDev	Minimum	Maximum
Rural Impervious	59	0.6679	0.1556	0.3529	0.9278
Urban Impervious	532	0.7897	0.1619	0.1364	1
Waterbodies	91	0.9034	0.1127	0.5432	1
Urban Grassland	45	0.7338	0.1787	0.282	1
Urban Woodland	29	0.8695	0.0732738792	0.7207	0.9703
Rural Grassland	837	0.7833	0.1795	0.1002	1
Rural Woodland	820	0.9094	0.09209656963	0.3753	1

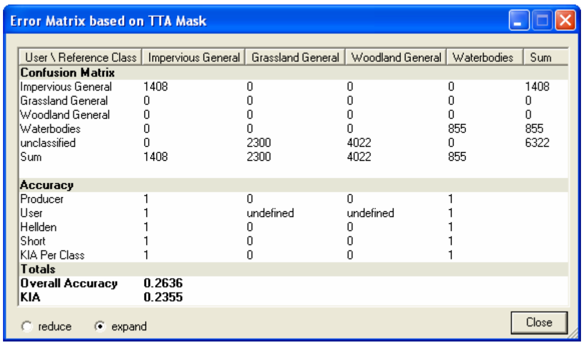
Figure 347: Output of the Best Classification Result statistics.

The statistical output for the best classification result is evaluated per class. Basic statistical operations are performed on the best classification result of the image objects assigned to a class (number of image objects, mean, standard deviation, minimum value and maximum value).

> ***Tip***
>
> *To display the comparable graphical output, go to the* ***View Settings*** *window and select* ***Best Classification Result****.*

11.3.3 Error Matrix Based on TTA (Training and Test Area) Mask

The **Error Matrix Based on TTA Mask** dialog box displays a statistic type used for accuracy assessment.


Error Matrix based on TTA Mask

User \ Reference Class	Impervious General	Grassland General	Woodland General	Waterbodies	Sum
Confusion Matrix					
Impervious General	1408	0	0	0	1408
Grassland General	0	0	0	0	0
Woodland General	0	0	0	0	0
Waterbodies	0	0	0	855	855
unclassified	0	2300	4022	0	6322
Sum	1408	2300	4022	855	
Accuracy					
Producer	1	0	0	1	
User	1	undefined	undefined	1	
Hellden	1	0	0	1	
Short	1	0	0	1	
KIA Per Class	1	0	0	1	
Totals					
Overall Accuracy	**0.2636**				
KIA	**0.2355**				

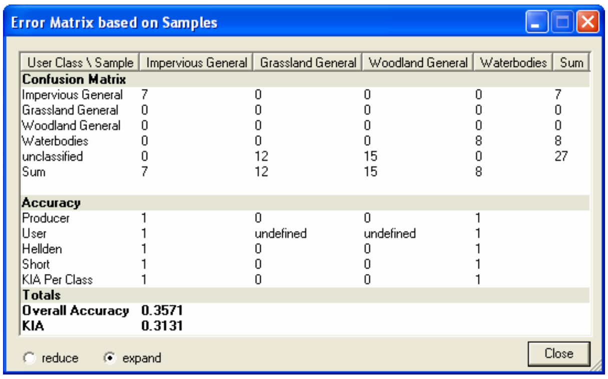
reduce expand Close

Figure 348: Output of the Error Matrix based on TTA Mask statistics.

Test areas are used as a reference to check classification quality by comparing the classification with ground truth based on pixels.

11.3.4 Error Matrix Based on Samples

The **Error Matrix Based on Samples** dialog box displays a statistic type used for accuracy assessment.


Error Matrix based on Samples

User Class \ Sample	Impervious General	Grassland General	Woodland General	Waterbodies	Sum
Confusion Matrix					
Impervious General	7	0	0	0	7
Grassland General	0	0	0	0	0
Woodland General	0	0	0	0	0
Waterbodies	0	0	0	8	8
unclassified	0	12	15	0	27
Sum	7	12	15	8	
Accuracy					
Producer	1	0	0	1	
User	1	undefined	undefined	1	
Hellden	1	0	0	1	
Short	1	0	0	1	
KIA Per Class	1	0	0	1	
Totals					
Overall Accuracy	**0.3571**				
KIA	**0.3131**				

reduce expand Close

Figure 349: Output of the Error Matrix based on Samples statistics.

This is similar to **Error Matrix Based on TTA Mask** but considers samples (not pixels) derived from manual sample inputs. The match between the sample objects and the classification is expressed in parts of class samples.

12 Export Data

Contents in This Chapter

You have several options for exporting data from the system in order to use it in other applications. Analysis result data can be exported as graphic information in vector or raster format. Additionally, statistical information can be created.

There are different basic methods to export analysis results:

- Data export triggered by a rule set that specifies where and how data is exported.
- Data export triggered by an action that specifies where and how data is exported.
- Data export initiated by **Export** menu commands. This is used to report data on a currently open project.

12.1 About Automated Data Export

Data export triggered by rule sets is executed automatically when the image analysis takes place. Only those items are exported that have been defined in the rule set by certain export algorithms.

You can modify where and how the data is exported.

→ Configure Exported Results on page 436

12.2 Report Data on a Single Project

Data export initiated by various **Export** menu commands applies only to single projects.

The **Export Current View** dialog box is used to export the current project view to a file.

→ Export Current View on page 408

Copy the current project view to clipboard. Choose **Export > Copy Current View to Clipboard** on the main menu bar.

→ Copy Current View to Clipboard on page 408

Class, object or scene **Statistics** can be viewed and exported. They are calculated from values of image object features.

→ Export Results as Statistics on page 405

Image Objects can be exported as a thematic raster layer together with an attribute table providing detailed parameter values.
The **Classification** of current image object level can be exported as an image file with an attribute table providing detailed parameter values.

→ Export Results as Raster File on page 402

Polygons, lines or points of selected classes can be exported as **Shapefiles**.

→ Export Results as Shape File on page 404

The **Generate Report** dialog box creates an HTML page about image objects, each specified by image object features, and optionally a thumbnail image.

→ Generate Report on page 406

Note

Most export functions automatically generate **.csv** files containing attribute information. To obtain correct export results, make sure the decimal separator for **.csv** file export matches the regional settings of your operating system. In **Definiens Developer**, these settings can be changed under **Tools > Options.**

Note

If georeferencing information of supported coordinate systems has been provided when creating a project, it should be exported along with the classification results and additional information if you choose **Export image objects** or **Export classification**.

12.2.1 Export Results as Raster File

The export type **Raster file** allows you to export image objects or classifications as raster layers together with attribute tables in **.csv** format containing parameter values.

Image objects or classifications can be exported together with their attributes. Each image object has a unique object or class ID and the information is stored in an attached attribute table that is linked to the image layer. The georeferencing information as provided when creating a project will be exported as well.

There are two possible locations for saving exported files:

- If a new project has been created but not yet saved, the exported files are saved to the folder where the image data are stored.
- If the project has been saved (recommended), the exported files are saved in the folder where the project has been saved.

Note

The thematic raster layer is saved as a 32-bit image file. But 32-bit image files cannot be opened by some image viewers. To view the file in **Definiens Developer**, add the 32-bit image file to a current project or create a new project and import the file.

To export image objects or classifications, open the Export Results dialog box by choosing Export > Export Results from the main menu bar.

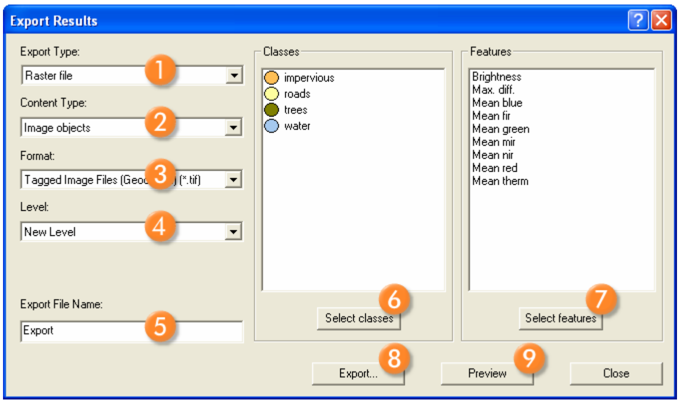


Figure 350: Exporting image objects with the Export Results dialog box.

1. Choose **Raster file** from the **Export Type** drop-down list.
2. From the **Content Type** drop-down list, choose one of the following:
   - **Image objects** to export all image objects with individual object IDs and their attributes.
   - **Classification** to export only the classification of image objects. The attached attribute table contains the class ID, color coding (RGB values) and class name by default. However, with this export type, adjacent image objects belonging to the same class can no longer be distinguished.
3. From the **Format** drop-down list, select the file format for the export file. Supported formats are:

Import File format	File Extension
ESRI ASCII GRID File	**.asc**
Erdas Imagine Image	**.img**
Tagged Image File (Geocoded)	**.tif**
JPEG JFIF	**.jpg**
JPEG 2000	**.jp2**
Portable Network Graphic	**.png**
Windows or OS/2 Bitmap	**.bmp**
PCIDSK	**.pix**
Tagged Image File	**.tif**
8 bit Portable Network Graphic	**.png**

4. Select the image object **Level** for which you want to export results.
5. Change the default file name in the **Export File Name** text box if desired.
6. Click the **Select classes** button to open the **Select Classes for Shape Export** dialog box where you can add or remove classes to be exported.
7. Click the **Select features** button to open the **Select Features for Export as Attributes** dialog box where you can add or remove features to be exported.
8. To save the file, press **Export**. An attribute table in **.csvq** file format is automatically created.
9. To view a preview of the attribute table that will be exported, press the **Preview** button.

12.2.2 Export Results as Shape File

Export shape files together with their attributes. You can export polygons, lines, or points of selected classes.

As with the **Export Raster File** option, image objects can be exported together with their attributes and classifications.

The georeferencing information as provided when creating a project is exported as well. The main difference to exporting image objects is that the export is not confined to polygons based on the image objects.

Generally, it is possible to choose between three basic shape formats:

- **Points**
- **Lines**
- **Polygons**

To export results as shapes, open the **Export Results** dialog box by choosing **Export > Export Results** from the main menu bar.

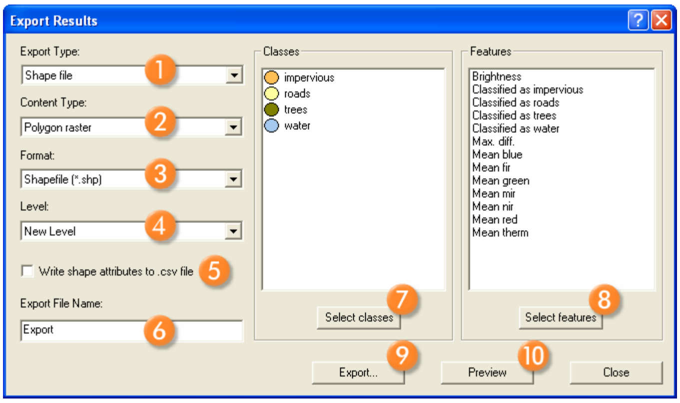


Figure 351: Exporting a shape file with the Export Results dialog box.

1. Choose **Shape file** from the **Export Type** drop-down list 1.
2. From the **Content Type** drop-down list 2, choose from different shape formats:
    - **Polygon raster** to export non-overlapping polygons following the raster outline. The exported shape file describes the border of the image objects along the pixel raster.
    - **Polygon smoothed (not overlapping)** to export non-overlapping polygons following the smoothed outline as defined by the polygonization.
    - **Polygon smoothed (individual)** to export the shape polygons following the smoothed outline as defined by the polygonization. Here, overlaps are possible.
    - **Polygon smoothed (with auto abstraction)**: The exported shape file describes the border of the image objects along abstracted and simplified outlines.
    - **Line skeleton** is based on all lines of a skeleton of each image object.
    - **Line main line** is based on the main line only of the skeleton of each image object.

→ About Skeletons on page 293

- **Point center of gravity** is the result of the calculation of the center of gravity for each image object.
- **Point center of main line** is the result of the calculation of the center of the main line for each image object.

3. The **Format** must be **Shapefile (*.shp)** 3.
4. Select the image object **Level** 4 for which you want to export results.
5. Select the **Write shape attributes to .csv file** check box 5 to store the shape attributes as statistics.
6. Change the default file name in the **Export File Name** text field 6 if desired.
7. Click the **Select classes** button to open the **Select Classes for Shape Export** dialog box 7 where you can add or remove classes to be exported.
8. Click the **Select features** button to open the **Select Features for Export as Attributes** dialog box 8 where you can add or remove features to be exported.

Note

You must be select at least one feature to be able in order to open shape files in third-party software such as **ArcGIS**.

9. To save the shape file to disk, press **Export** 9.
10. To view a preview of the attribute table that will be exported, press the **Preview** button 10.

    The export results in a **.dbf** file, an **.shp** file and an **.shx** file. The **.dbf** file supports string, int and double formats and the columns are formatted automatically according to the data type. The column width is adjustable up to 255 characters.

Note

The class names and class colors are not exported automatically. Thus, if you want to export shapes for more than one class and you want to distinguish the exported features by class, you should also export the feature **Class name**. You can use the **Class color** feature to export the RGB values for the colors you have assigned to your classes.

12.2.3 Export Results as Statistics

Use the **Statistics** Export Type in the **Export Results** dialog box to calculate and to export class statistics. The statistics are calculated from values of image object features.

To export statistics, open the **Export Results** dialog box by choosing **Export > Export Results** from the main menu bar.

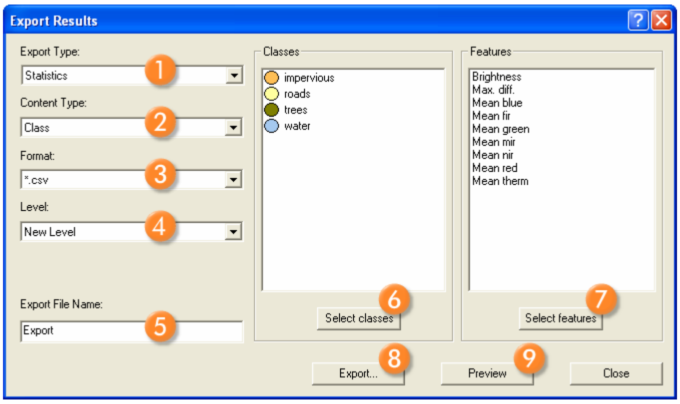


Figure 352: Exporting statistics using the Export Results dialog box.

1. Choose **Statistics** from the **Export Type** drop-down list.
2. From the **Content Type** drop-down list, choose to export statistics for:
   - **Classes**
   - **Objects**
   - **Scenes**
3. The **Format** has to be ***.csv.**
4. Select the image object **Level** for which you want to export results. If **Scene** has been selected as **Content Type,** this option is not available.
5. Change the default file name in the **Export File Name** text field if desired.
6. Click the **Select classes** button to open the **Select Classes for Shape Export** dialog box where you can add or remove classes to be exported. This button is only active when choosing **Class** from the Content Type drop-down list.
7. Click the **Select features** button to open the **Select Features for Export as Attributes** dialog box where you can add or remove features to be exported.
8. To save the statistics to disk, press **Export**.
9. To view a preview of the attribute table that will be exported, press the **Preview** button.

12.2.4 Generate Report

The **Generate Report** dialog box creates a HTML page containing information about image object features and optionally a thumbnail image.

To open the **Generate Report** dialog box, choose **Export > Generate Report** on the main menu bar.

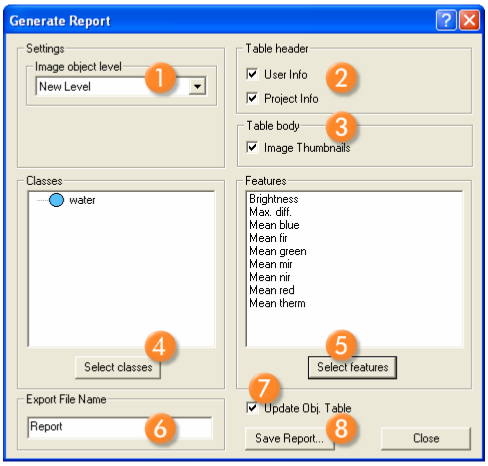


Figure 353: Generate Report dialog box.

1. Select the **Image object level** for which you want to create the report from the drop-down list.
2. The **Table header** group box allows you to choose from the following options:
   - **User Info** to include information about the user/author of the project
   - **Project Info** to include coordinate information, resolution and units of the project.
3. From the **Table body** group box, choose whether or not to include thumbnails of the image objects in **.jpg** format.
4. Click the **Select classes** button to open the **Select Classes for Report** dialog box where you can add or remove classes to be included in the report.
5. Click the **Select features** button to open the **Select Features for Report** dialog box where you can add or remove features to be included in the report.
6. Change the default file name in the **Export File Name** text field if desired.
7. Deselect the **Update Obj. Table** check box if you don't want to update your object table when saving the report.
8. To save the report to disk, press **Save Report**.

ID	class	Brightness	Mean blue	Mean green	Mean red	Max. diff	Image
0	water	36.2472	61.5169	21.2472	15.9438	3.3748	
1	water	43.4460	61.3734	21.8816	18.6576	2.8524	
2	water	55.5092	74.6410	29.0513	29.6154	2.6884	
3	water	52.8636	69.6515	26.6061	26.8030	2.8074	
4	water	39.2796	61.4255	21.8298	18.4043	3.1601	
5	water	39.0504	62.2353	22.8824	18.6471	3.2658	

Figure 354: Sample HTML report with thumbnail images.

12.2.5 Export Current View

Exporting the current view is an easy way to save the project view at the current scene scale to file which can be opened and analyzed in other applications.

This export type does not include additional information such as georeferencing, features or class assignments. To reduce the image size, you can rescale it before exporting.

1. To export a current view, choose **Export** > **Current View** from the main menu bar. The **Select Scale** dialog box opens.

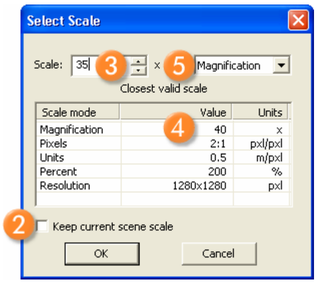


Figure 355: Select Scale dialog box.

2. To export the scene with the displayed scale 3, click **OK**. If you want to keep the original scale of the scene, select the **Keep current scene scale** check box 2.
3. You can select a different **Scale** 3 compared to the current scene scale. That way you can export the current view at a different magnification/resolution.
4. If you enter an invalid **Scale** factor, it will be changed to the closest valid scale as displayed in the table 4.
5. To change the current scale mode, select from the drop-down list box 5.
6. Confirm with **OK** and the **Export Image Layer** dialog box opens.
7. Enter a file name and select the file format from the drop-down list. Click **Save** to confirm. The current view settings are used; however, the zoom settings are ignored.

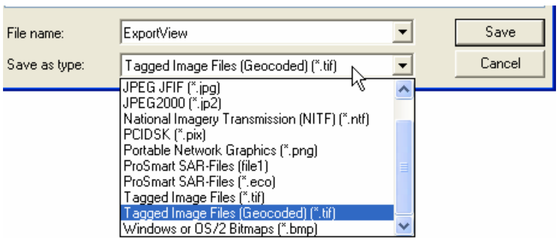


Figure 356: Export Image Layer dialog box (cutout).

> **Note**
>
> Not all formats that can be selected in the drop down list are available for exporting.

12.2.6 Copy Current View to Clipboard

Exporting the current view to clipboard is an easy way to create screenshots which can then be inserted into other applications.

To create a screenshot for use in other applications without saving it to file, there are two options:

- Choose **Export > Copy Current View to Clipboard** from the main menu bar.
- Right-click the **Project view** and choose **Copy Current View to Clipboard** from the context menu.

This export type does not include additional information such as georeferencing, features or class assignments.

12.3 Export Content of a Window

Save the content from a window to a file or to the clipboard.

Many windows which contain lists or tables. You can save them to a file or to the clipboard to be used in other program. Others contain diagrams or images which you can copy to the clipboard.

Right-click to display the context menu and choose:

- **Save to File** allows you to save the table contents as **.csv** or transposed **.csv** (**.tcsv**) file. The data can then be further analyzed in applications like **Microsoft® Excel**.
- **Copy to Clipboard** saves the current view of the window to clipboard. It can then be inserted as a picture into other program e.g. **Microsoft® Office** or an image processing program.

13 User Interface Elements

Contents in This Chapter

The **User Interface Elements** chapter guides you to detailed instructions corresponding to user interface elements like:

- Menus
- Toolbars and command buttons
- Dialog boxes and windows

13.1 Menus

13.1.1 File Menu

File Menu commands	Usage
New Workspace...	Create a new workspace.
Open Workspace...	Open an existing workspace.
Close Workspace	Close the currently open workspace.
Save Workspace	Save the currently open workspace to file.
Save Workspace As...	Save the currently open workspace to another file.
Workspace Properties	Define another output root folder.
Predefined Import...	Import scenes to the selected item of the workspace.
Customized Import...	Import multiple scenes according to a file structure which you can customize to suit your needs.
Import Existing Project...	Import an existing project into a workspace.
New Project... Ctrl + N	Create a new project.
Open Project... Ctrl + O	Open a project file that is not part of a workspace.
Close Project	Close the currently open project.
Save Project Ctrl + S	Save the currently open project to a separate file.
Save Project As...	Save the currently open project to another file.
Modify Open Project	Modify a selected project by exchanging image layers. You can edit image layer aliases, project name and display unit.
User Information...	Set user information to be included in rule sets.
Manage Geodatabase Connections	Create an **ArcSDE** connection. Available only for Definiens Data Management in combination with eCognition **Life** Server.
Exit	Exit the program.

→ Create a New Workspace on page 57
→ Open an Existing Workspace on page 84
→ Close a Workspace on page 83
→ Save a Workspace on page 83
→ See above.
→ Create a New Workspace on page 57
→ Move a Workspace on page 83
→ Import Scenes to a Workspace on page 58
→ Customize the Import of Scenes into a Workspace on page 63
→ Import Existing Project to Workspace on page 87
→ Create a New Project on page 44
→ Open a Separate Project on page 43
→ Close a Project on page 56
→ Save a Project on page 55
→ See above.
→ Modify a Project on page 54
→ User Information on page 463
→ Data Management User Guide

13.1.2 View Menu

View Menu commands	Usage	
View Settings	Open or close the **View Settings** window. Set visualization method for displaying image objects and image layers.	→ View Settings on page 463
Image Layer Mixing...	Define a color composition for display of image layers and set equalizing options.	→ Edit the Image Layer Mixing on page 96
Thumbnail Settings	Set image layer mixing for thumbnails.	→ Edit Image Layer Mixing for Thumbnails on page 98
Cursor Mode ▸	Set the cursor mode for navigating within projects. See sub-menu items below.	→ Navigate Within Projects on page 114
▸ **Normal Cursor**	Switch from zoom mode or pan mode to normal cursor.	→ Normal Cursor on page 103
▸ **Zoom In**	Zoom into point.	→ Zoom In on page 104
▸ **Zoom Out**	Zoom out from point.	→ Zoom Out on page 104
▸ **Area Zoom** Ctrl + U	Drag a rectangle on the display to zoom into an area.	→ Area Zoom on page 103
▸ **Panning** Ctrl + P	Move within project using the hand shaped cursor.	→ Panning on page 105
Display Mode ▸	Navigate within projects. See sub-menu items below.	→ Navigate Within Projects on page 114
▸ **Zoom In Center** +	Zoom into current project view center.	→ Zoom In Center on page 104
▸ **Zoom Out Center** −	Zoom out from current project view center.	→ Zoom Out Center on page 104
▸ **Zoom 100%**	Set zoom value to 100%.	→ Zoom 100% on page 105
▸ **Zoom to Window**	Zoom the project to fit into the current window.	→ Zoom to Window on page 105
▸ **Edit Highlight Colors...**	Set highlight colors and display color of unclassified image objects.	→ Edit Highlight Colors on page 444
Scale Bar	Display a scale bar in the project view. See sub-menu items below.	→ View a Scale Bar on page 107
Visible	Display the scale bar.	→ See above.
Upper Left Corner	Display the scale bar in the upper left corner.	→ See above.
Upper Right Corner	Display the scale bar in the upper right corner.	→ See above.
Lower Left Corner	Display the scale bar in the lower left corner.	→ See above.
Lower Right Corner	Display the scale bar in the lower right corner.	→ See above.
Manage Data View	Open or close windows for loading and managing data.	→ Use Workflow Views on page 40
Analysis Builder View	Open or close windows for assembling and configuring actions.	→ See above.
View Results View	Open or close windows for analyzing the results of data processing.	→ See above.
Develop Rule Set View	Open or close windows for developing rule sets.	→ See above.
Save Current View	Save a modified workflow view.	→ See above.

Restore Default	Go back to the default settings of the workflow views of the user interface.	→ See above.
Windows ▸	Open or close windows. See sub-menu items below.	→ **Windows** on page 39
▸ **Workspace**	Manage your image analysis task and administrate all relevant data.	→ **Manage Projects in a Workspace** on page 56
▸ **Plate View**	Manage data and access results. Available only for Definiens eCognition Life Server	→ **Manage Data in the Plate View** on page 90
▸ **Pan Window** Ctrl + Shift + P	Move to different regions of the image.	→ **Pan Window** on page 106
▸ **Magnifier**	Display an image in detail in a separate window.	→ **Magnify the Image in a Separate Window** on page 106
▸ **Legend**	View the legend of the currently displayed classification.	→ **View Legend of Classes** on page 380
▸ **Image Object Table**	Display a table of selected feature values of all image objects of selected classes.	→ **Compare Multiple Image Objects by Using the Image Object Table** on page 178
▸ **Image Object Information**	Display detailed information about features and classification of image objects.	→ **Investigate Single Image Objects by Using the Image Object Information** on page 174
▸ **Thematic Layer Attribute Table**	View the numeric attributes of thematic objects.	→ **View Thematic Layer Attributes** on page 286
▸ **2D Scatter Plot**	Compares two features of a project. Available only for Definiens eCognition Life Server	→ **Compare Two Features by Using the 2D Scatter Plot** on page 186
▸ **Graph**	View statistical values of a certain feature for selected wells. Available only for Definiens eCognition Life Server	→ **View Result Values in the Graph** on page 384
▸ **Heat Map**	View thumbnails and results in a heat map. Available only for Definiens eCognition Life Server	→ **View Thumbnails in the Heat Map Window** on page 380
▸ **Dose Response Curve**	Determine the IC50 or EC50 of a test compound. Available only for Definiens eCognition Life Server	→ **View Results in a Dose Response Curve** on page 385
▸ **Analysis Builder**	Assemble actions to a build solution.	→ **Open the Analysis Builder** on page 315
▸ **Process Tree**	Edit processes and create rule sets to perform image analysis.	→ **Organize a Process Sequence** on page 141
▸ **Process Profiler**	Measure the execution time of processes and optimize your processes for performance.	→ **Improve the Processing Performance** on page 152
▸ **Snippets**	Save process snippets to a library and reuse them.	→ **Save and Reuse Process Snippets** on page 148
▸ **Find and Replace**	Search within a rule set for rule set items and edit or replace them.	→ **Find and Replace Rule Set Items** on page 147

▸ **Class Hierarchy**	Create and to edit a class hierarchy.	→ Manage Classes in the Class Hierarchy on page 250
▸ **Feature View**	Compare features values (image object attributes).	→ Compare Feature Values by Using the Feature View on page 179
▸ **Rule Set Documentation**	View, store, and edit documentation for a rule set.	→ Document a Rule Set on page 153
▸ **Sample Editor**	View a histogram of feature values of samples.	→ Work with the Sample Editor on page 226
▸ **Sample Selection Information**	Assess the quality of a sample.	→ Assess the Quality of Samples on page 230
Toolbars ▸	View or hide toolbars. See sub-menu items below.	→ Toolbars on page 38
▸ **File**	Use tools for handling data.	→ File Toolbar on page 424
▸ **View Settings**	Use tools for displaying data.	→ View Settings Toolbar on page 425
▸ **View Navigate**	Use tools for navigating within the image object hierarchy of a project.	→ NavigateToolbar on page 426
▸ **Zoom Functions**	Use tools for navigating within a project by zooming and panning.	→ Zoom Toolbar on page 426
▸ **Manual Editing**	Use tools for manually editing of image objects and thematic objects.	→ Manual Editing Toolbar on page 427
▸ **Architect**	Use tools for building analysis solutions by choosing from a set of predefined actions.	→ Architect Toolbar on page 429
▸ **Sample**	Use tools for working with samples.	→ Samples Toolbar on page 430
▸ **Tools**	Use various commands for developing rule sets.	→ Tools Toolbar on page 430
Appearance	Change the appearance of **Definiens Developer**.	→ Customize the User Interface on page 42
Office 2000 **Office XP** **Office 2003** **Windows XP** **Visual Studio 2005**		
Customize...	Customize toolbars, keyboard shortcuts and menus to your personal preferences.	→ Customize on page 439
Message Console	Open or close the **Message Console** window.	→ Message Console on page 451
Status Bar	View or hide an information bar at the bottom line of the main window.	→ Status Bar on page 39

13.1.3 Image Objects Menu

Image Objects Menu commands	Usage
Image Object Information	Open or close the **Image Object Information** window. Display detailed information about features and classification of image objects.
Image Object Table	Open or close the **Image Object Table** window. Display a table of selected feature values of all image objects of selected classes.
Edit Level Names...	Rename image object levels, define image object aliases and create level variables.
Delete Levels...	Select one or more image object levels from the image object hierarchy for deleting.
Copy Current Level	Duplicate an image object level and insert the copy above the currently active one.

→ Investigate Single Image Objects by Using the Image Object Information on page 174
→ Compare Multiple Image Objects by Using the Image Object Table on page 178
→ Edit an Image Object Level or Level Variable on page 196
→ Delete Image Object Level on page 197
→ Duplicate Image Object Level on page 195

13.1.4 Analysis Menu

Analysis Menu commands	Usage
Analysis Builder	Open or close the **Analysis Builder** window. Add actions to a solution.
Analyze...	Analyze an automated on selected projects.
Cancel All Jobs	Cancel analysis of waiting or processing projects.
Create Tiles...	Create Tiles from a project.
Stitch Projects...	Stitch together processed tile results to create results of the entire project.
Run Solution	Run a solution in the **Analysis Builder** window.
Run Until Selected Action	Run a solution until the selected action.
Rollback All	Rollback all changes to restore one or multiple projects to the first version.

→ Assemble a Solution from Actions in the Analysis Builder on page 317
→ Start Automated Image Analysis on page 363
→ Cancel Image Analysis of Waiting and Currently Processing Projects on page 369
→ Create Tiles on page 376
→ Stitch Tile Results on page 376
→ Execute Solution Manually for Testing on page 362
→ See above.
→ Rollback All Changes of Projects on page 86

13.1.5 Library Menu

Library Menu commands	Usage
Load Calibration...	Load general settings of a solution stored in a calibration **.psf** file.
Save Calibration...	Saves general settings of a solution to file.
Load Solution...	Load an existing solution to the **Analysis Builder** window.
Save Solution...	Save the currently open solution to a separate file.
Save Solution Read-Only...	Save the currently open solution as a read-only file.
Edit Action Library	Activate the action library editing mode.
New Action Library...	Create a new action library.
Open Action Library...	Load an existing action library.
Close Action Library	Close the current action library.
Save Action Library	Save an action library.
Import Action...	Load an additional action from file to an action library.
Export Action...	Export an action definition to file.

→ Load Calibration Corresponding to Image Reader Settings on page 320
→ Create Calibration Parameter Set File on page 323
→ Load Solution on page 321
→ Save Analysis Settings as Solution on page 320
→ See above.
→ Assemble and Edit an Action Library on page 309
→ Create a New Action Library on page 309
→ Open Action Library on page 316
→ Close Action Library on page 316
→ Assemble and Edit an Action Library on page 309
→ Import Action Definition from File on page 323
→ Export Action Definition to File on page 315

13.1.6 Classification Menu

Classification Menu commands	Usage
Class Hierarchy	Open or close the **Class Hierarchy** window. Create and view classes and show their groups and inheritance affiliation.
Legend	Open or close the **Legend** window. View the legend of the currently displayed classification.
Class Hierarchy ▸	See sub-menu below.
Nearest Neighbor ▸	See sub-menu below.
Samples ▸	See sub-menu below.
Advanced Settings ▸	See sub-menu below.

→ Manage Classes in the Class Hierarchy on page 250
→ View Legend of Classes on page 380

13.1.6.1 Class Hierarchy (Classification Submenu)

Classification: Class Hierarchy Menu commands	Usage	
▸ **Delete Classification** Ctrl+D	Open the **Delete Classification** dialog box. Delete the classification of specified image objects.	→ Delete Classification on page 442
▸ **Edit Classes** ▸	See sub-menu items below	
▸ **Edit Class Description**	Open the **Class Description** dialog box. Edit the currently selected class.	→ Edit the Class Description on page 246
▸ **Insert Class**	Open the **Edit Class Description** dialog box. Add a new class to the class hierarchy.	→ Create and Edit a Class on page 241
▸ **Insert Class Variable**	Open **Create Class Variable** dialog box. Add a new class variable to the hierarchy.	→ Create a Class Variable on page 259
▸ **Duplicate Class** Ctrl+C	Create a copy of the currently selected class.	→ Duplicate a Class on page 243
▸ **Delete Classification**	Delete the classification of all image objects of the selected class on the current image object level.	→ Delete Classification on page 442
▸ **Select Color...**	Open the **Colors** dialog box. Change the color of the selected class.	→ Create and Edit a Class on page 241
▸ **Active**	Set to inactive to exclude a class from the classification process.	→ See above.
▸ **Abstract**	Set a class to abstract so they are not applied to image objects directly.	→ See above.
▸ **Mark As Shared**	Mark class as shared to use it in several rule sets.	→ See above.
▸ **Delete Samples**	Delete samples of the currently selected class.	→ Delete Samples on page 232
▸ **Delete Class**	Delete the currently selected class.	→ See above.
▸ **Sort Classes** ▸	See sub-menu items below.	
▸ **Alphabetically**	Display classes in alphabetical order.	
▸ **Manually**	Display classes in user-defined order.	
▸ **Enable Sort Mode**	Enable manual sorting mode of the class hierarchy.	
▸ **Load Class Hierarchy...**	Open the **Load Class Hierarchy** dialog box. Load an existing class hierarchy.	→ Load and Save Class Hierarchies on page 254
▸ **Save Class Hierarchy...**	Open the **Save Class Hierarchy** dialog box. Save your class hierarchy as **.dkb** file.	→ See above.
▸ **Export Class Hierarchy...**	Open the **Export Class Hierarchy** dialog box. Export your class hierarchy as **.xml** file.	→ Export a Class Hierarchy on page 255
▸ **Delete Class Hierarchy...**	Delete the existing class hierarchy and processes.	→ Delete a Class Hierarchy on page 255

13.1.6.2 Nearest Neighbor (Classification Submenu)

Classification: Nearest Neighbor Menu commands	Usage	
Edit Standard NN Feature Space...	Edit the feature space.	→ **Define the Feature Space with Nearest Neighbor Expressions** on page 221
Feature Space Optimization...	Select classes and features for optimization.	→ **Optimize the Feature Space** on page 237
Apply Standard NN to Classes...	Select classes and apply the Standard NN to them.	→ **Apply the Standard Nearest Neighbor Classifier** on page 222
Set NN Function Slope...	Enter a value and set the function slope.	→ **Set Nearest Neighbor Function Slope** on page 237

13.1.6.3 Samples (Classification Submenu)

Classification: Samples Menu commands	Usage	
Select Samples	Switch to sample selection mode. Double-click on an image to assign it as a sample.	→ **Select Samples** on page 228
Select Samples with Brush	Select samples with the Sample Brush.	→ **Select Samples with the Sample Brush** on page 230
Sample Editor	Open the **Sample Editor** window. View a histogram of feature values of samples.	→ **Work with the Sample Editor** on page 226
Sample Selection Information	Open the **Sample Selection Information** window and assess the quality of a samples.	→ **Assess the Quality of Samples** on page 230
Sample Editor Options ▸	See sub-menu items below.	→ **Work with the Sample Editor** on page 226
Select Features to Display...	Select the features visible as histograms in the **Sample Editor**.	
Display Standard Nearest Neighbor Features	Display the nearest neighbor features visible as histograms in the **Sample Editor** window.	
Membership Functions ▸	See sub-menu items below.	
Compute	Automatically compute a membership function.	→ **Generate Membership Functions Automatically** on page 216
Edit/Insert	Edit or insert a membership function.	
Delete	Delete the selected membership function.	
Parameters...	Set membership function parameters.	→ **Edit Membership Function Parameters** on page 217
Display Membership Functions	Show or hide membership functions over the feature histograms.	
Display Entire Feature Range	Toggle the display range between maximal feature range and used feature range.	
Display Axis Labels	Show or hide axis labels for feature histograms.	
Display Samples from Inherited Classes	Display samples from inherited classes in the **Sample Editor** window.	
Activate Sample Navigation	Activate or deactivate the sample navigation.	→ **Navigate Samples** on page 231

Delete Samples of Classes	Ctrl + Alt + M	Select classes for which to delete samples.	→ **Delete Samples** on page 232
Delete All Samples	Shift + M	Delete all samples of all classes.	→ **Delete Samples** on page 232
Create Samples from TTA Mask...		Automatically create samples using an existing TTA mask.	→ **Load and Apply TTA Mask** on page 233
Create TTA Mask from Samples...		Create a TTA mask from a selected level.	→ **Create and Save a TTA Mask** on page 233
Load TTA Mask...		Load training and test areas as TTA mask layer with a conversion table.	→ **Load and Apply TTA Mask** on page 233
Save TTA Mask...		Choose a name and location for the file.	→ **Create and Save a TTA Mask** on page 233
Edit Conversion Table...		Display and edit the conversion table.	→ **Edit Conversion Table** on page 234

13.1.6.4 Advanced Settings (Classification Submenu)

Classification: Advanced Settings Menu commands	Usage	
Edit Minimum Membership Value...	Define the minimum value for a classification. All lower values will be defined as unclassified.	→ **Edit Minimum Membership Value** on page 218
Select Image Layers for Brightness...	Select the image layers to be included in the calculation of the **Brightness** feature.	→ **Reference Book**: Feature **Brightness**

13.1.7 Process Menu

Process Menu commands	Usage	
Process Tree	Open or close the **Process Tree** window. Create, modify and execute processes in order to classify image objects. Organize a process sequence.	→ **Use Processes to Build Rule Sets** on page 130
Process Profiler	Open or close the **Process Profiler** window. Measure the execution time of processes and optimize your processes for performance.	→ **Improve the Processing Performance** on page 152
Snippets	Open or close the **Snippets** window. Save process snippets to a snippets library and reuse them.	→ **Save and Reuse Process Snippets** on page 148
Find and Replace	Search within a rule set for rule set items and edit or replace them.	→ **Find and Replace Rule Set Items** on page 147
Rule Set Documentation	View, store, and edit documentation for a rule set.	→ **Document a Rule Set** on page 153

Load a Rule Set	Load a rule set from file.	→ Load a Process on page 144
Load and Append Rule Set	Load a process from file and add it to existing processes.	→ See above.
Save Rule Set	Save the rule set to file.	→ Save a Process or Rule Set on page 145
Delete Rule Set	Delete the complete rule set.	→ Delete a Process or Rule Set on page 146
Encrypt Rule Set	Encipher a rule set.	
Decrypt Rule Set	Remove the encryption of a rule set.	
Process Commands ▸	See sub-menu items below.	
▸ **Edit Process** Ctrl + Enter	Edit the selected process.	→ Edit a Single Process on page 131
▸ **Execute** F5	Execute the selected process.	→ Execute a Single Process on page 141
▸ **Execute on Selected Object**	Execute the process on the selected object only.	→ Execute a Single Process on page 141
▸ **Append New Process** Ctrl + A	Append a new process after the selected process.	→ Create a Single Process on page 132
▸ **Insert Child** Ctrl + I	Insert a new process as a child process of the selected process.	→ See above.
▸ **Copy**	Duplicate the selected process.	→ Duplicate Process on page 146
▸ **Paste**	Paste a process into the Process Tree	
▸ **Delete Process**	Delete the selected process.	→ Delete a Process or Rule Set on page 146
▸ **Decrypt Process**	Remove the encryption of a process.	
▸ **Create Customized Algorithm...**	Create a customized algorithm based on a process sequence.	→ Create a Customized Algorithm on page 278
▸ **Edit Customized Algorithm...**	Edit the customized algorithm.	→ See above.
▸ **Edit Customized Algorithm Default Parameters...**	Edit the customized algorithm default parameters.	→ See above.
▸ **Active**	Set to inactive to exclude a process from execution.	→ Execute Processes for Testing on page 151
▸ **Breakpoint** F9	Insert a breakpoint for the selected process. Stop the execution of processes at this process.	→ See above.
▸ **Update View**	Update the project view while processing.	→ See above.
Edit Aliases ▸	See sub-menu items below.	→ About Layer Aliases on page 54
▸ **Image Layer Aliases...**	Edit and assign image layer aliases.	→ Manage Layer Aliases on page 450
▸ **Thematic Layer Aliases...**	Edit and assign thematic layer aliases.	→ See above.
▸ **Attribute Table Columns...**	Modify a thematic layer attribute table by adding, editing or deleting table columns or editing table rows.	→ Modify Thematic Layer Attribute Table on page 287

V=	**Manage Variables...**	Edit, add, or delete variables.	→ **Manage Variables** on page 261
P=	**Manage Parameter Set...**	Manage parameter Sets.	→ **Manage Parameter Sets** on page 268
	Manage Metadata Conversion...	Provide a selection of the available metadata to the feature tree.	→ **Convert Metadata to Provide it to the Feature Tree** on page 52

13.1.8 Tools Menu

Tools Menu commands		Usage	
Feature View	Ctrl + F	Open or close the **Feature View** window. Compare image object attributes.	→ **Compare Feature Values by Using the Feature View** on page 179
Manage Customized Features...		Manage features that are adapted to your needs.	→ **Use Customized Features** on page 269
Load Customized Features...		Load customized features from **.duf** file.	→ See above.
Save Customized Features...		Save customized features as **.duf** file.	→ See above.
Image Layer Histograms...	Ctrl + H	View each layer's histogram.	→ **Check Image Layer Histograms** on page 101
2D Feature Space Plot...		View a 2D plot of selected features.	→ **Analyze the Correlation of Two Features by Using the 2D Feature Space Plot** on page 188
Feature Space Optimization...		Optimize the feature space.	→ **Optimize the Feature Space** on page 237
Accuracy Assessment...		Produce statistical outputs which can be used to check the quality of the classification results.	→ **Accuracy Assessment** on page 398
Thematic Layer Attribute Table		View the numeric attributes of thematic objects.	→ **View Thematic Layer Attributes** on page 286

Manual Editing ▸	See sub-menu items below.	→ Manual Editing on page 391
▸ **Single Selection**	Select a single object.	→ Selection Tools on page 391
▸ **Polygon Selection**	Select all objects within a polygon.	→ See above.
▸ **Line Selection**	Select all objects along a line.	→ See above.
▸ **Rectangle Selection**	Select all objects within a rectangle.	→ See above.
▸ **Cut Objects**	Cut a single image object or thematic object manually.	→ Cut an Object Manually on page 395
▸ **Merge Objects**	Manually merge selected neighboring image or thematic objects.	→ Merge Objects Manually on page 392
▸ **Filter Classes for Multiple Image Object Selection...**	Select only those objects that belong to certain classes.	→ Selection Tools on page 391
▸ **Classify Image Objects**	Manually classify image objects on the current image object level.	→ Classify Image Objects Manually on page 393
▸ **Generate Thematic Objects**	Generate new thematic objects by drawing.	→ Generate Thematic Objects by Drawing on page 396
▸ **Delete Selected Objects**	Delete one or more selected thematic objects.	→ Delete Thematic Objects on page 397
▸ **Save Thematic Layer As...**	Save a thematic layer as **.shp** file.	→ Save Thematic Objects in Thematic Layer on page 397
Options...	Change various optional settings.	→ Options on page 452

13.1.9 Export Menu

Export Menu commands	Usage	
Export Results…	View and export your analysis results as raster files, vector files or statistics.	→ Report Data on a Single Project on page 401
Generate Report…	Create a HTML page about image objects, each specified by image object features and optionally a thumbnail image.	→ Generate Report on page 406
Current View…	Export the view of the project to file.	→ Export Current View on page 408
Copy Current View to Clipboard	Copy the current project view to the clipboard.	→ Copy Current View to Clipboard on page 408

13.1.10 Window Menu

Window Menu commands	Usage	
Split	Split the project view horizontally and vertically.	→ Use Different Viewing Modes on page 116
Split Horizontally	Split the project view horizontally.	→ See above.
Split Vertically	Split the project view vertically.	→ See above.
Independent View	Unlink project view to adjust view settings separately.	→ Use Independent Project Views on page 118
Side by Side View	Synchronize the display area of other project views with the active window.	→ Use Linked Side-by-Side Project Views on page 119
Swipe View	Apply the view settings for one section of a divided project view across the project view by dragging and resizing sections.	→ Use Flexible Visualization with Swipe View on page 119
Enable Docking	Enable or disable magnetism and dynamic resizing of windows within the main window.	→ Handle Docking Windows on page 41

13.1.11 Help Menu

Help Menu commands	Usage	
System Info...	Display system configuration and state information.	→ System Info on page 462
Definiens Developer Help F1	Open the **User Guide**.	→ Access to this User Guide on page 467
Reference Book	Open the Reference Book.	
Help Keyboard...	Short descriptions and shortcuts of all commands.	→ Help Keyboard on page 446
About...	See information about **Definiens Developer** and Definiens AG.	

13.2 Toolbars and Command Buttons

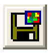
Note

Depending on the used workflow view, some buttons might not be displayed. If you like to display all available buttons, you can reset toolbars in the **Customize** dialog box.

→ **Customize** on page 439

13.2.1 File Toolbar

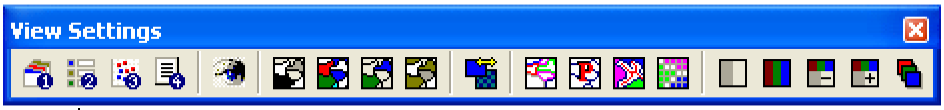
Use tools for handling data.

	**Create New Project** Ctrl + N Create a new project.
	**Open Project** Ctrl + O Open a project file that is not part of a workspace.
	**Save Project** Ctrl + S Save the currently open project to a separate file.
	**Create New Workspace** Create a new workspace.
	**Open Workspace** Open an existing workspace.
	**Save Workspace** Save the currently open workspace to file.
	**Import Scenes** Import scenes to the selected item of the workspace.

→ Load and Manage Data on page 43
→ Create a New Project on page 44
→ Open a Separate Project on page 43
→ Save a Project on page 55
→ Create a New Workspace on page 57
→ Open an Existing Workspace on page 84
→ Save a Workspace on page 83
→ Import Scenes to a Workspace on page 58

13.2.2 View Settings Toolbar

Use tools for displaying data.



Manage Data View Open or close windows for loading and managing data.	→ View Data in Projects on page 93
Analysis Builder View Open or close windows for assembling and configuring actions.	→ Use Workflow Views on page 40 → See above.
View Results View Open or close windows for analyzing the results of data processing.	→ See above.
Develop Rule Set View Open or close windows for developing rule sets.	→ See above.
View Settings Open **View Settings** window. Set visualization method for displaying image objects and image layers.	→ View Settings on page 463
View Layer Display current level using mixed image layer values of pixel.	→ View Layer on page 109
View Classification Display classification of image objects.	→ View Classification on page 109
View Samples Show selected samples on current image object level.	→ View Samples on page 112
Feature View Show objects with the value of the selected feature in the feature view.	→ Feature View on page 111
Pixel View or Object Mean View Switch between pixel view and object mean view.	→ Pixel View or Object Mean View on page 111
Show or Hide Outlines Show or hide outlines of image objects.	→ View Outlines of Image Objects on page 108
Show or Hide Polygons Show or hide polygons. .	→ View Polygons on page 112
Show or Hide Skeletons Show or hide skeletons.	→ View Skeletons on page 114
Image View or Project Pixel View Display of the scene at the original scene.	→ Compare Rescaled Scene Copy with Original Scene on page 108
Single Layer Grayscale Display one image layer to see it in grayscale mode.	→ Single Layer Grayscale on page 95
Mix Three Layers RGB Display three image layers to see image in RGB mixing.	→ Three Layers RGB on page 95
Show Previous Image Layer In grayscale mode: Display the previous image layer. In RGB mode: Change color composition for the layers to display.	→ Show Previous Image Layer on page 96
Show Next Image Layer In grayscale mode: Display the next image layer. In RGB mode: Change color composition for the layers to display.	→ Show Next Image Layer on page 96
Edit Image Layer Mixing Define a color composition for display of image layers and set equalizing options.	→ Edit the Image Layer Mixing on page 96

13.2.3 NavigateToolbar

Use tools for navigating within and modifying the image object hierarchy of a project.

→ Navigate Within the Image Object Hierarchy on page 114

	Copy Level Duplicate an image object level and insert the copy above the currently active one.	→ Duplicate Image Object Level on page 195
	Delete Level Select one or more image object levels from the image object hierarchy for deleting.	→ Delete Image Object Level on page 197
New1	**Select Level in Object Hierarchy** Select level in image object hierarchy.	→ Navigate Within the Image Object Hierarchy on page 114
	Next Level Down in Object Hierarchy Next level lower in the in hierarchy of image objects.	→ See above.
	Next Level Up in Object Hierarchy Next level higher in the in hierarchy of image objects.	→ See above.

13.2.4 Zoom Toolbar

Use tools for navigating within a project by zooming and panning.

→ Zoom Within Projects on page 101

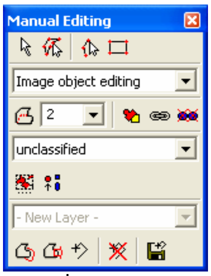
	Normal Cursor Switch from zoom mode or panning mode to normal cursor.	→ Normal Cursor on page 103
	Panning Ctrl + P Move project using the hand shaped cursor.	→ Panning on page 105
	Area Zoom Ctrl + U Drag a rectangle on the display to zoom into an area.	→ Area Zoom on page 103
	Zoom In Center + Zoom into the current project view center.	→ Zoom In Center on page 104
	Zoom Out Center - Zoom out from current project view center.	→ Zoom Out Center on page 104
50%	**Zoom** Select or enter a zoom value to change the display in the project view.	→ Select Zoom Value on page 104
	Zoom to Window Zoom project to fit into the project view.	→ Zoom to Window on page 105

	Zoom In Zoom into point.	→ Zoom In on page 104
	Zoom Out Zoom out from point.	→ Zoom Out on page 104
	Zoom 100% Display the image at its original scale.	→ Zoom 100% on page 105
	Pan Window Move to different regions of the image.	→ Pan Window on page 106
	Magnifier Display an image in detail in a separate window.	→ Magnify the Image in a Separate Window on page 106

13.2.5 Manual Editing Toolbar

Use tools for manually editing of image objects and thematic objects.

→ Manual Editing on page 391



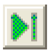
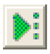
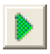
	Single Selection Mode Single selection mode.	→ Selection Tools on page 391
	Line Selection Draw a line to select all image objects touched by the line.	→ See above.
	Polygon Selection Draw a polygon to select all image objects within or touched by the polygon.	→ See above.
	Rectangle Selection Draw a rectangle to select all image objects within or touched by the rectangle.	→ See above.
Image object editing Image object editing Thematic editing	**Change Editing Mode** Choose between **Image Object Editing** and **Thematic Layer Editing** mode.	→ Change Editing Mode on page 391
	Cut Object Manually Cut a single image object or thematic object manually.	→ Cut an Object Manually on page 395
10	**Snapping Tolerance [pxl]** Define the snapping tolerance in pixel. When using Cut Object Manually, snapping facilitates touching object borders similar to magnetism.	→ See above.
	Merge Objects Manually Select manually neighboring image objects or thematic objects for merging.	→ Merge Objects Manually on page 392
	Merge Selected Objects Merge selected image objects or thematic objects to one object.	→ See above.
	Clear Selection for Manual Object Merging Clear all selected image object and do not merge objects.	→ See above.

unclassified	**Select Class for Manual Classification** Select a class that will be assigned to image objects using **Classify Image Objects**.	→ Classify Image Objects Manually on page 393
	Filter Classes for Multiple Image Object Selection Define classes, whose image objects can be selected by **Line Selection**, **Polygon Selection**, or **Rectangle Selection**.	→ Selection Tools on page 391
	Classify Image Objects Classify manually an image object by clicking.	→ Classify Image Objects Manually on page 393
ThematicLayer.shp	**Select Thematic Layer** Select an existing thematic layer or create a new one.	→ Manage Thematic Objects on page 396
	Merge Thematic Object Based on Image Object Select a polygon and an adjacent image object. Right-click and choose **Merge to Polygon** to merge to an extended polygon.	→ Merge Thematic Object Based on Image Object on page 393
	Generate Thematic Object Based on Image Object Select an image object. Right-click and choose **Generate Polygon** to create a polygon.	→ Generate Thematic Object Based on Image Object on page 397
	Generate Thematic Objects Draw nodes for a polygon. Right-click and choose Close Polygon to generate a thematic object.	→ See above.
	Delete Selected Thematic Object Delete a selected thematic object.	→ Delete Thematic Objects on page 397
	Save Thematic Layer As Save a thematic layer as **.shp** file.	→ Save Thematic Objects in Thematic Layer on page 397

13.2.6 Architect Toolbar

Use tools for building analysis solutions by choosing from a set of predefined actions.

→ **Build Analysis Solution** on page 315

	**Analysis Builder** Assemble actions to a build solution.
	**Save Solution as File** Save the currently open solution as a separate file.
	**Save Solution Read-Only** Save the currently open solution as a read-only file.
	**Load Solution From File** Load an existing solution to the **Analysis Builder** window.
	**Load a Calibration** Load general settings of a solution stored in a calibration **.psf** file.
	**Run Solution Until Selected Action** Execute solution until the selected action.
	**Run Selected Action** Execute a selected action.
	**Run Solution** Execute solution in the **Analysis Builder** window.
	**Run Solution on eCognition Server** Test the configured solution available in the **Analysis Builder** window on a selected project.

→ **Open the Analysis Builder** on page 315
→ **Save Analysis Settings as Solution** on page 320
→ See above.
→ See above.
→ **Load Calibration Corresponding to Image Reader Settings** on page 320
→ **Execute Solution Manually for Testing** on page 362
→ See above.
→ See above.

13.2.7 Samples Toolbar

Use tools for working with samples.

→ Use Samples for Nearest Neighbor Classification on page 219

Select Sample
Select Sample in **Sample Editor** window.

→ Select Samples on page 228

Select Samples with Brush
Use the **Sample Brush** to select samples.

→ Select Samples with the Sample Brush on page 230

Sample Editor
Open or close the **Sample Editor** window.

→ Work with the Sample Editor on page 226

Sample Selection Information
Open or close the **Sample Selection Dialog** box.

→ Assess the Quality of Samples on page 230

Activate/Deactivate Sample Navigation
Activate or deactivate **Sample Navigation.**

→ Navigate Samples on page 231

Navigate to Next Sample
Switch to the next sample.

→ See above.

Navigate to Previous Sample
Switch to the previous sample.

→ See above.

Select Sample Navigation Level
Switch between samples.

→ See above.

13.2.8 Tools Toolbar

Use various commands for developing rule sets.

→ Develop Efficient Rule Sets on page 192
→ Process Data on page 361
→ Examine Results on page 379

Workspace
Manage your image analysis task and administrate all relevant data.

→ Manage Projects in a Workspace on page 56

Image Object Information
Open or close the **Image Object Information** window.
Display detailed information about features and classification of image objects.

→ Investigate Single Image Objects by Using the Image Object Information on page 174

Object Table
Open or close the **Image Object Table** window.
Display a table of selected feature values of all image objects of selected classes.

→ Compare Multiple Image Objects by Using the Image Object Table on page 178

Legend
Open or close the **Legend** window.
View the legend of the currently displayed classification.

→ View Legend of Classes on page 380

Create Tiles Create Tiles from a project.	→ Create Tiles on page 376
Stitch Tiles Stitch together processed tile results to create results of the entire project.	→ Stitch Tile Results on page 376
Define Rectangular Subset Draw a rectangle in the scene to define a subset of a project for processing.	→ Define a Scene Subset on page 371
Define Polygonal Subset Draw a polygon in the scene to define a subset of a project for processing.	→ See above.
Save Subset to a Workspace Save a selected subset to the database.	→ See above.
Analyze Projects Analyze selected projects.	→ Start Automated Image Analysis on page 363
Cancel Waiting and Processing Projects Cancel analysis of waiting or processing projects.	→ Cancel Image Analysis of Waiting and Currently Processing Projects on page 369
Class Hierarchy Open or close the **Class Hierarchy** window. Create and to edit a class hierarchy.	→ Manage Classes in the Class Hierarchy on page 250
Process Tree Open or close the **Process Tree** window. Edit processes and create rule sets to perform image analysis.	→ Organize a Process Sequence on page 141
Find and Replace Search for all occurrences of rule set items within a rule set.	→ Find and Replace Rule Set Items on page 147
Rule Set Snippets Organize and save process snippets for reuse in other rule sets.	→ Save and Reuse Process Snippets on page 148
Rule Set Documentation View, store, and edit documentation for a rule set.	→ Document a Rule Set on page 153
Process Profiler Measure the execution time of processes and optimize your processes for performance.	→ Improve the Processing Performance on page 152
Feature View Ctrl + F Open or close the **Feature View** window. Compare image object attributes.	→ Compare Feature Values by Using the Feature View on page 179
Manage Customized Features Manage features that are adapted to your needs.	→ Create Customized Features on page 269
Manage Variables Edit, add, or delete variables.	→ Manage Variables on page 261
Manage Parameter Sets Manage parameters.	→ Manage Parameter Sets on page 268
Undo Undo the last operation.	→ Undo Process Editing Operations on page 139
Undo/Redo Process Editing drop-down list undo (or redo) multiple previous operations	→ See above.
Redo Process Editing Redo the last undone operation.	→ See above.

	Accept Project Mark selected project as accepted.	→ Accept and Reject Results on page 390
	Reject Project Mark selected project as rejected.	→ See above.
	Manual Editing Toolbar View or hide the **Manual Editing** toolbar. Influence the result of an image analysis by manual editing.	→ Manual Editing Toolbar on page 427
	Message Console Open or close the **Message Console** window.	→ Message Console on page 451
	Show/Hide Thematic Layer Attribute Table View the numeric attributes of thematic objects.	→ View Thematic Layer Attributes on page 286
	Feature Space Optimization Optimize the feature space.	→ Optimize the Feature Space on page 237

13.3 Dialog Boxes and Windows

13.3.1 2D Feature Space Plot

See the related instructional section.

→ Analyze the Correlation of Two Features by Using the 2D Feature Space Plot on page 188

13.3.2 2D Scatter Plot

See the related instructional section.

→ Compare Two Features by Using the 2D Scatter Plot on page 186

13.3.3 Accuracy Assessment

See the related instructional section.

→ Accuracy Assessment on page 398

13.3.4 Analysis Builder

See the related instructional section.

→ Build Analysis Solution on page 315

→ Create Action Libraries on page 306

13.3.5 Apply Standard Nearest Neighbor to Classes

See related instructional section.

→ Apply the Standard Nearest Neighbor Classifier on page 222

13.3.6 Assign Image Layer Alias

When loading a rule set or a process, there might occur conflicts concerning the aliases of the image layers. If the image layer aliases of the loaded rule set or process are not identical to those of the currently open project, the **Assign Image Layer Alias** dialog box opens automatically to support you in solving this naming conflict.

→ About Layer Aliases on page 54

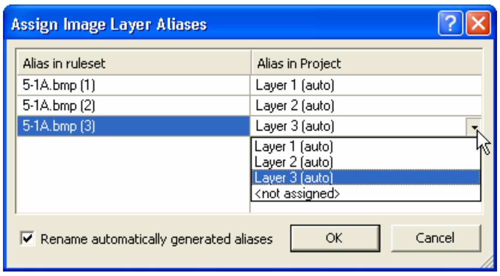


Figure 357: Edit Image Layer Alias dialog box

1. The **Assign Image Layer Alias** dialog box opens automatically in concern of loading rule sets or processes.

→ Load a Process on page 144

2. Select an image layer alias of the loaded rule set or process in the left column.
3. Click the drop-down arrow in the right column and select an image layer of the currently open project to assign it to the selected image layer alias of step 2.
4. Repeat step 2 and 3 for every image layer alias.
5. If you assign image layer aliases of the currently open project that have been edited before, they will not be renamed.

   For automatically generated image layer aliases in the currently open project, you can select:

   - Select **Rename automatically generated aliases** check box to rename them as the loaded image layer aliases as listed in the left column.
   - Clear the **Rename automatically generated aliases** check box to keep the automatically generated image layer aliases in the currently open project as listed in the right column.

6. Click **OK** to confirm. To check or modify the image layer aliases use the **Manage Layer Aliases** dialog box.

→ Manage Layer Aliases on page 450

13.3.7 Assign No-Data-Values

Set the value of those pixels that are not to be analyzed.

> **Note**
>
> No-data-value definitions only can be applied to projects that have not yet been analyzed.

No-data-values can be assigned to image pixel values (or combinations of values) to save processing time. These areas will not be included in the image analysis. Typical examples for no data values are bright or dark background areas.

1. To open the **Assign No Data Value** dialog box, do one of the following:

- Create a new project.

Create New Project

→ Create a New Project on page 44

- Modify an existing project.

→ Modify a Project on page 54

The **Create Project** or the **Modify Project** dialog box opens.

2. After importing image layers press the **No Data** button. The **Assign No Data Values** dialog box opens.

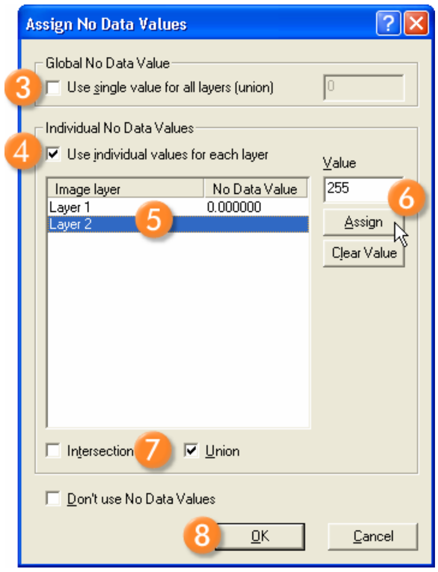


Figure 358: Assign No Data Values dialog box.

3. Global 3: Set a single pixel value for all image layers.

4. Individual 4: Select to set individual pixel values for each image layer.

5. Select one or more image layers 5.

6. Enter a value 6 for those pixels that are not to be analyzed. Click **Assign**. Example: In the figure above, the no data value of **Layer 1** is **0.000000**. This implies that all pixels of the image layer **Layer 1** with the value **0.000000** (i.e. the darkest pixels) are excluded from the analysis. The no data value of **Layer 2** is set to **255** in the **Value** field.

Find Out More

No Data Value

The values which can be selected may vary considerably. They depend on the type of input image.

No data values are useful if large areas of the image consist of areas that do not contain useful information for the analysis (e.g. black margins in satellite images). However this also means that some areas within the image are classified as no data and therefore are not analyzed. Using no data values will generally save processing time. However in calculations that use relations to neighboring objects, it may slow the analysis down.

7. Select one of the check boxes 7:

- Select **Intersection** to include those no data areas only that all images have in common (overlapping no-data areas).
- Select **Union** to include the no data areas of all individual layers for the whole project, i.e. if a no data value is found in one layer, this area is treated as no data in all other layers too.

8. Confirm with **OK** 8 to return to the superordinate dialog box.

13.3.8 Class Description

See the related instructional section.

→ Create and Edit a Class on page 241

13.3.9 Class Hierarchy

See the related instructional section.

→ Manage Classes in the Class Hierarchy on page 250

13.3.10 Colors

This dialog box is used to edit the color information of classes and to define highlight colors.

To open do one of the following:

- Open the **Class Description** dialog. Check the color list item and click **Other**.
- Double-click in the left field of the **View Settings** dialog box. Check one color list item and click **Other**.

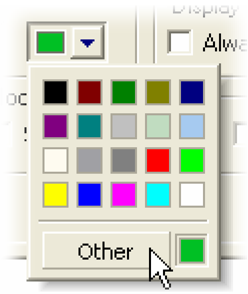


Figure 359: Colors dialog box

The **Colors** dialog box opens and offers different options to set colors.

13.3.11 Configure Analysis

The **Configure Analysis** dialog box is used to configure the image analysis of both single projects and multiple projects that are embedded in folders of the **Workspace** window.

1. Right-click the root item in the tree view on the left pane of the **Workspace** window and choose **Configure Analysis**. The **Configure Analysis** dialog box opens.
2. For more details, see the **Start Analysis Job** dialog box, which is similar.

→ Start Automated Image Analysis on page 363

13.3.12 Configure Exported Results

In the **Configure Exported Results** dialog box you can specify where and how the results of image analysis and related data are exported.

You can change the default export path and define settings like file format.

1. To open the **Configure Exported Results** dialog box, do one of the following:

   - Open the **Start Analysis Job** dialog box and select **Edit**.
   - Open the **Configure Analysis** dialog box and select **Edit**.

Analyze Projects

→ Start Automated Image Analysis on page 363
→ Configure Analysis on page 435

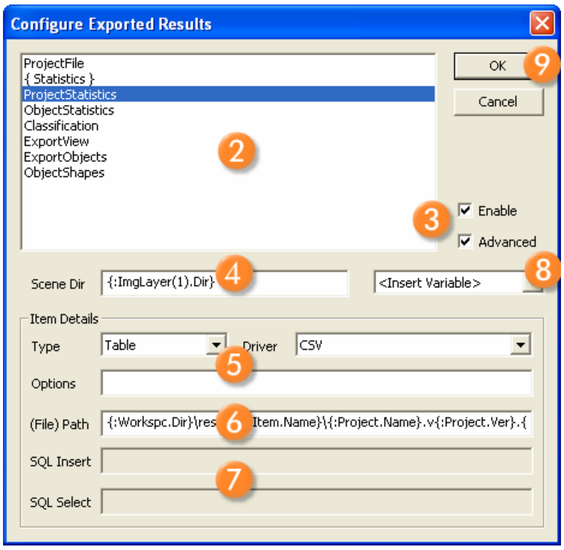


Figure 360: Configure Exported Results dialog box.

2. Display of exported items 2 which will be exported when the analysis is performed. Apart from the project file which is exported automatically, only those items are exported that have been specified in the rule set.

   - ProjectFile exports each project file (**.dpr**) that is created in the image analysis.
   - Any statistics the process specifies to be exported. In the figure above, the specified analysis process contains an export statistics algorithm called **Statistics** (which has been disabled in this example).
   - **ProjectStatistics** creates a **.csv** file by default. This presumes that the used rule set includes the **Export project statistics algorithm** defining the file name **ProjectStatistics.**
   - **ObjectStatistics** creates a **.csv** file by default. This presumes that the rule set includes the **Export object statistics algorithm** defining the file name **ObjectStatistics.**

- **Classification produces a .tif raster file** by default**.** This presumes that the rule set includes the **Export classification view algorithm** defining the file name **Classification.**
- **ExportView produces a .jpg raster file** by default**.** This presumes that the rule set includes the **Export current view algorithm** defining the file name **ExportView.**
- **ExportObjects produces a geocoded .tif raster file** by default**.** This presumes that the rule set includes the **Export thematic raster files algorithm** defining the file name **ExportObjects.**
- **ExportShapes produces** a **.shp** vector file and additional auxiliary files. This presumes that the rule set includes the **Export vector layers** algorithm defining the file name **ExportShapes**.

3. Select an item from the list.
   Enable 3 or disable the exporting functionality. Disabled items are displayed in braces: **{item}** (like the {Statistics} item in the figure above).
   Check **Advanced** 3 to view further configuration options.
4. **Scene Dir** 4 specifies the path which will be used in the **{:ImgLayer(1).Dir}** variable inside the **(File) Path** template. If this variable is not used in the template, then the **Scene Dir** text box is ignored.
5. You can edit the **Type** 5 and an appropriate **Driver** for exported items.

   For **ProjectFile**, you can choose between the following drivers:

   - **DPR -Analyst** exports image objects and classes only.
   - **DPR - Developer** additionally exports rule sets, variables, and features. Use this for further modifications of the rule set.

   For **Statistics** (**ProjectStatistics**, **ObjectStatistics**) you can select one of the following drivers:

   - Select **CSV** and the **(File) Path** field is automatically filled in with the name and path of the **.csv** file that will be created. You can change this if you wish. In the **Options** text box, you can write **append** to append the data to those in a pre-existing file.
   - Select **ODBC** and in the (File) Path field, you should fill in **DSN=MS Access Database;DBQ=DIA.mdb** where **DIA.mdb** is the path and name of a pre-existing **Access** database file.
     Classification, **ExportView** and **ExportObject** allow you to choose from a variety of table, raster, or vector file formats, or to specify a path.
6. The **(File) Path** 6 text box is automatically filled in but can be manually edited. The file name and path is made up of a number of variables in **{: }**. These variables are:

   - **Workspc.Name** represents the name of the workspace.
   - **Workspc.Dir** represents the directory in which the workspace file is stored.
   - **Project.Name** represents the name of the project.
   - **Project.Ver** represents the version of the project.
   - **Project.Dir** represents the directory in which the project file is stored.

- **Scene.Dir** represents the directory in which the individual scenes are stored.
- **ImgLayer(n).Dir** represents the folder in which the image layer is stored. You have to change the default value **n** to a number representing the layer in the project, e.g. 1 for the first layer in a project.
  If this variable is used, you can define its contents in the **Scene Dir** 4 text box.
- **Item.Name** represents the name of the exported item which is defined in the process algorithm parameters.
- **Ext** represents the name of the file extension.

Example: A workspace **New Workspace.dpj** was stored in the directory **D:\work\Definiens**, and the name of the Workspace is **New Workspace**. So the (file) path from **{:Workspace.Dir}\{:Workspace.Name}.csv** means that the file path is **D:\work\Definiens\New Workspace.csv.**

Note that special characters like \ have to be manually inserted.

Alternatively to using variables to define the export path, you can also make an absolute reference to your preferred directory. In the above example, the file path **D:\work\Definiens\New Workspace.csv** could be entered directly like that instead of using variables.

7. If ODBC has been selected as Driver under 5, SQL commands should be entered in the **SQL Insert** and the **SQL Select** text boxes 7.
   Example for **SQL Insert**:
   INSERT INTO statistics VALUES (%row%,%feature%,%value%)
   Example for **SQL Select**:
   SELECT * FROM statistics_cross.
8. The **<Insert Variable>** 8 drop down list box can be used for building up the **(File) Path** or the **Scene Dir** path by using internally held variables (e.g. **Workspace.Dir**). To include a variable as part of the **(File) Path**, click in **(File) Path** and select a variable from **<Insert Variable>**. The variable will be added in the **(File) Path**.
9. Click **OK** 9 to close the dialog box.

13.3.13 Configure Image Object Table

See related instructional section.

→ Compare Multiple Image Objects by Using the Image Object Table on page 178

13.3.14 Conversion Table

See related instructional section.

→ Edit Conversion Table on page 234

13.3.15 Create New Action Library

See related instructional section.

→ Create a New Action Library on page 309

13.3.16 Create New Workspace

See related instructional section.

→ Create a New Workspace on page 57

13.3.17 Create Project

See related instructional section.

→ Create a New Project on page 44

13.3.18 Create Samples from TTA Mask

See the related instructional section.

→ Load and Apply TTA Mask on page 233

13.3.19 Create Tiles

See related instructional section.

→ Create Tiles on page 376

13.3.20 Create TTA Mask from Samples

See the related instructional section.

→ Create and Save a TTA Mask on page 233

13.3.21 Create Variable

See the related instructional section.

→ Create a Variable on page 258

13.3.22 Customize

The **Customize** dialog box allows customization and resetting of toolbars, keyboard shortcuts and menus.

To open, choose **View** > **Customize** from the main menu. The **Customize** menu lets you move toolbar buttons through drag-and-drop operations.

Commands Tab

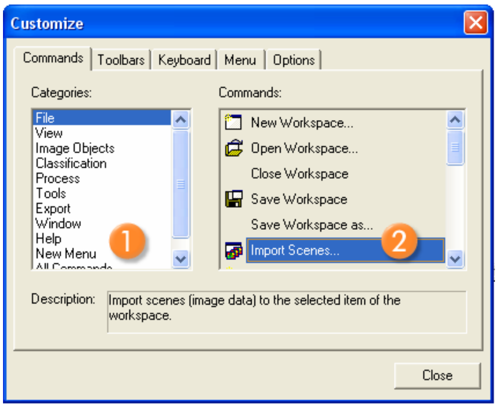


Figure 361: Commands tab of the Customize dialog box

1. From the **Commands** tab, choose the **Category** you want to customize.

2. From the **Commands** list box, drag a command to the appropriate toolbar and drop it at the desired position.

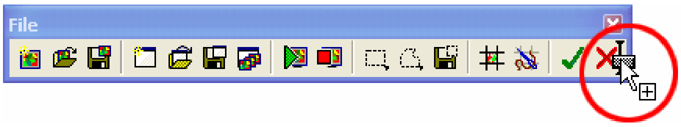


Figure 362: The plus sign indicates that the command button will be placed at the shown position.

Toolbars Tab

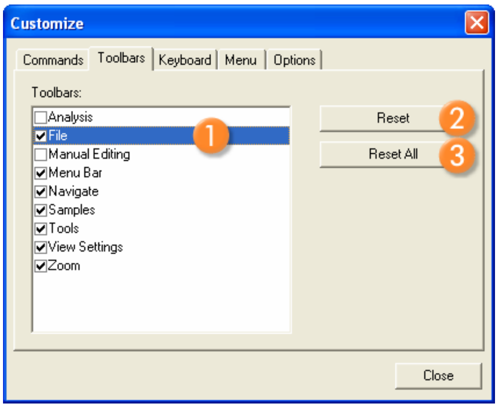


Figure 363: Toolbars tab of the Customize dialog box

1. The **Toolbars** tab allows you to switch toolbars on or off.
2. Click the **Reset** button to display all available commands buttons of the selected toolbar.
3. Click the **Reset All** button to display all available commands buttons of all toolbars.

Keyboard Tab

Customize keyboard shortcuts to your preferred settings.

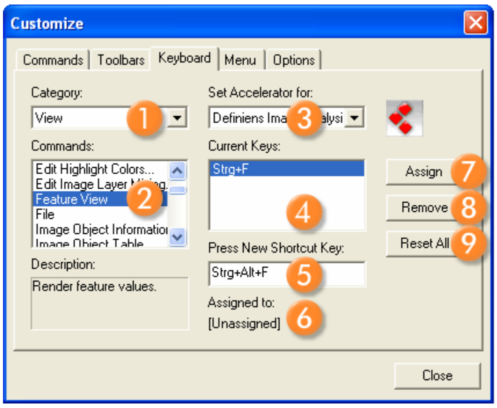


Figure 364: Keyboard tab of the Customize dialog box

1. Select a **Category** from the drop-down list box.
2. Select the **Commands** from the list box. A shortcut key can be assigned for the activated command.
3. The **Set Accelerator for** field needs to be set to Definiens Image Analysis.

4. Display the current keyboard shortcut.
5. Assign a **New Shortcut Key**. To change the shortcut, make sure the cursor is in this text box.
6. Informs you if the newly assigned shortcut is currently assigned to another command. If it is, assign another shortcut or change the existing shortcut.
7. Click **Assign** to confirm the new shortcut key.
8. Click **Remove** to delete a selected shortcut key from the command.
9. Click **Reset All** to assign default values to all shortcut keys.

Menu Tab

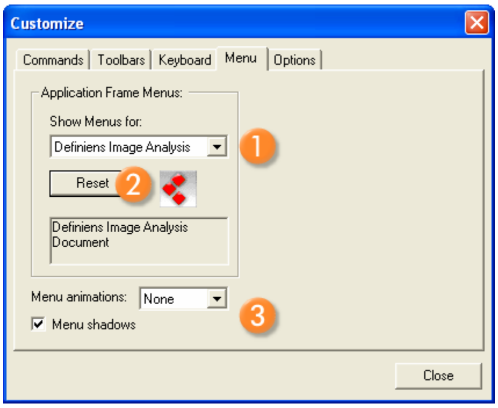


Figure 365: Menu tab of the Customize dialog box

1. Under **Application Frame Menus** set the **Show Menus for** to Definiens Image Analysis.
2. Click **Reset** to reset all main menu bars.
3. Change the look of the menu display according to your personal preferences.

Options Tab

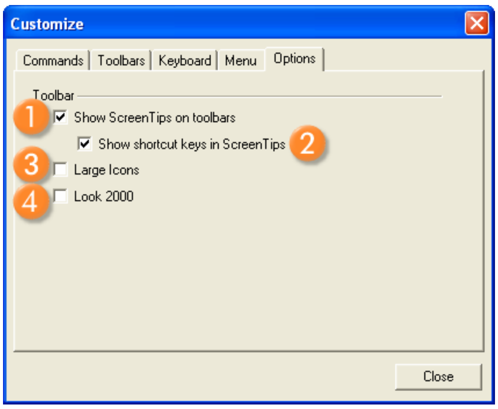


Figure 366: Options tab of the Customize dialog box

1. Select the **Show ScreenTips on toolbars** check box to display tool tips.
2. Select the **Show shortcut keys in ScreenTips** check box to display shortcut keys in tool tips.

3 Select the **Large Icons** check box to display larger toolbar icons.

4 Select the **Look 2000** check box if you prefer a **Windows 2000** look.

13.3.23 Customized Algorithm Properties

See the related instructional section.

→ **Create a Customized Algorithm** on page 278

13.3.24 Customized Import

See the related instructional section.

→ **Perform a Customized Import** on page 63

13.3.25 Define Brightness

See the related instructional section.

→ See feature **Brightness** in **Reference Book**

13.3.26 Delete Classification

You can delete a classification result manually in order to recreate it.

1. To delete the result of a classification, do one of the following:

   - Right-click a class and select **Delete Classification** on the context menu. This deletes the classification of all image objects of the selected class on the current active image object level.

   
   Delete Classification

   - Select **Classification** > **Class Hierarchy** > **Delete Classification** from the main menu bar.
     In the opening **Delete Classification** dialog box, you can delete the complete classification or only parts of it. Choose whether to delete by class, classification type, or image object level.

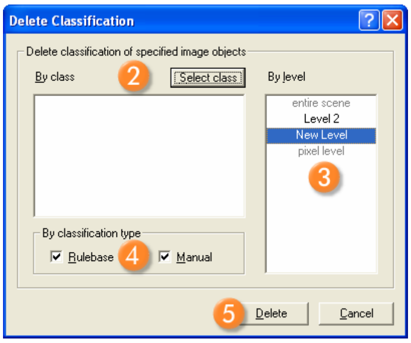


Figure 367: Delete Classification dialog box.

2. Click the **Select class** button 2 to add or remove classes to be deleted.

→ **Select Classes** on page 456

3. Choose an image object level 3 to delete the classification of a selected image object level.
4. Select 4 to delete the manual and/or the classification based on your rule base.
5. Click the **Delete** button 5 to delete the classification with your settings.

13.3.27 Delete Levels

See related instructional sections.

→ Delete Image Object Level on page 197
→ Navigate Within the Image Object Hierarchy on page 114
→ Execute a Single Process on page 141

13.3.28 2D Scatter Plot

See the related instructional section.

→ View Results in a Dose Response Curve on page 385

13.3.29 Edit Classification Filter

Select and edit classes.

1. The **Edit Classification Filter** dialog box is primarily used in the **Edit Process** dialog box when creating or editing a process within the **Process Tree** window. Open the **Edit Classification Filter** dialog box from:

→ Create a Single Process on page 132

   - The **Image Object Domain** group box. Click the broad **All Objects** button (alternatively displaying the names of selected classes).
   - The **Parameter** group box by clicking on the ellipsis button.

   The **Edit Classification Filter** dialog box displays the available classes in hierarchy as a group tree.

... **(ellipsis button)**

Groups

→ Manage Classes in the Class Hierarchy on page 250

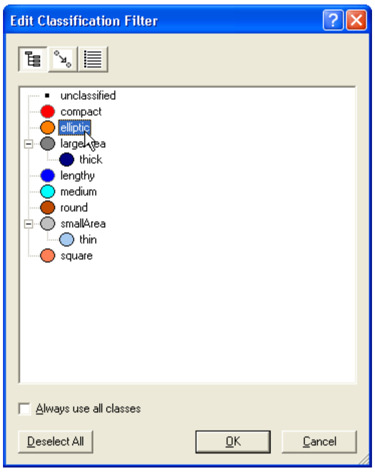


Figure 368: Edit Classification Filter dialog box.

2. Select a desired class.
3. To select multiple classes, click the **List** button. Hold the Ctrl key while selecting multiple classes with the cursor.

List

4. Use all available classes by selecting the **Always use all classes** check box.
5. Deselect all chosen classes by using the **Deselect All** button. Alternatively, use the context menu.
6. Press **OK** to activate the selection.

Options:

7. To display the available classes in hierarchy as an inheritance tree.

Inheritance

→ About Inheritance on page 252

8. To search for a specific class select the **List** mode and type the class name into the **Find** text box.
9. To edit a class, right-click it and choose **Edit Class** on the context menu. The **Class Description** dialog box opens.

→ Create and Edit a Class on page 241

10. To create a new class, right-click any class and choose **Insert Class** on the context menu. The **Class Description** dialog box opens.

The **Edit Classification Filter** dialog box is also utilized here:

- From **Manual Editing** tools
- **Segmentation** dialog box
- **Edit Action Dependencies** dialog box

→ Edit Action Dependencies on page 312

13.3.30 Edit Minimum Membership Value

See the related instructional section.

→ Edit Conversion Table on page 234

13.3.31 Edit Highlight Colors

The **Edit Highlight Colors** dialog box is used to set highlight colors and the display color of non-sample and image objects in the **View Layer** mode.

→ View Layer on page 109

1. To open the **Edit Highlight Colors** dialog box do one of the following:
   - Choose **View** > **Display Mode** > **Edit Highlight Colors** from the main menu.
   - If you have the **View Settings** dialog box open then double-click on the left pane.

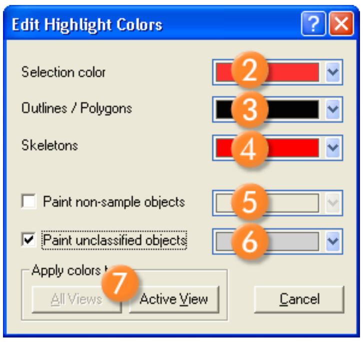


Figure 369: Edit Highlight Colors dialog box.

2 Changes the outline color of selected image objects.

3 Changes the color of displayed outlines and polygons.

4 Changes the color of displayed skeletons.

5 Select the check box if you want to paint non-sample objects displayed in the **Sample View** mode and then select the display color.

6. Select the check box if you want to paint unclassified objects displayed in the **View Classification** mode and then select the display color.

7. In the **Apply colors to** area you can do one of the following:

   - Apply the settings to **All Views** that are currently open.
   - Apply the settings to the **Active View** only.

13.3.32 Edit Level Alias

See related instructional section.

→ **Edit an Image Object Level or Level Variable** on page 196

13.3.33 Edit Minimum Membership Value

See the related instructional section.

→ **Edit Minimum Membership Value** on page 218

13.3.34 Edit Parameter Set

See related instructional section.

→ **Edit Parameter Set** on page 267

13.3.35 Edit Process

See related instructional section.

→ **Create a Single Process** on page 132

13.3.36 Edit Standard Nearest Neighbor Feature Space

See related instructional section.

→ **Select Features** on page 457

13.3.37 Edit Threshold Condition

See related instructional sections.

→ **Define the Image Object Domain** on page 135
→ **Use Thresholds for Classification with Class Descriptions** on page 212

13.3.38 Export Current View

See related instructional section.

→ **Export Current View** on page 408

13.3.39 Export Results

The **Export Results** dialog box allows you to export your analysis results in the following formats:

- **Shape file** → Export Results as Shape File on page 404
- **Raster file** → Export Results as Raster File on page 402
- **Statistics** → Export Results as Statistics on page 405

See related instructional sections.

13.3.40 Feature Space Optimization

See related instructional section.

→ Optimize the Feature Space on page 237

13.3.41 Feature View

See related instructional section.

→ Compare Feature Values by Using the Feature View on page 179

13.3.42 Find and Replace

See the related instructional section.

→ Find and Replace Rule Set Items on page 147

13.3.43 Generate Report

See related instructional section.

→ Generate Report on page 406

13.3.44 Graph

See the related instructional section.

→ View Result Values in the Graph on page 384

13.3.45 Help Keyboard

The **Help Keyboard** dialog box provides brief descriptions of all commands.

1. To open the **Help Keyboard** dialog box, go to the main menu and choose **Help > Help Keyboard**.

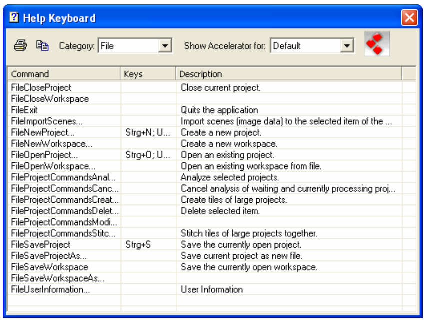


Figure 370: Help Keyboard dialog box.

2. Select a menu from the **Category** drop-down list box.

Options

For printing the currently displayed category, click the **Print** button within the dialog box.

 **Print**

Select one command or select multiple commands while pressing the Ctrl key. To copy to clipboard, click the **Copy** button within the dialog box.

 **Copy to Clipboard**

13.3.46 Image Layer Histograms

See related instructional section.

→ **Check Image Layer Histograms** on page 101

13.3.47 Image Layer Weights

For selected algorithms you can define the weighting of image layers. The weighting of thematic layers works similar.

→ **Image Layer Weighting** on page 338

→ **Use a Thematic Layer for Segmentation** on page 289

1. Open the **Edit Process** dialog box and select a relevant algorithm e.g. **multiresolution segmentation**.

→ **Create a Single Process** on page 132

2. In the **Algorithm parameters** area, expand the **Image Layer weights** list and set the weight of the image layers to be considered by the algorithm. You can use both of the following methods:

   - Select an image layer and edit the weight value placed inside the value field.
   - Select **Image Layer weights** and click the ellipsis button located inside the value field to open the **Image Layer Weights** dialog box.

(**ellipsis button**)

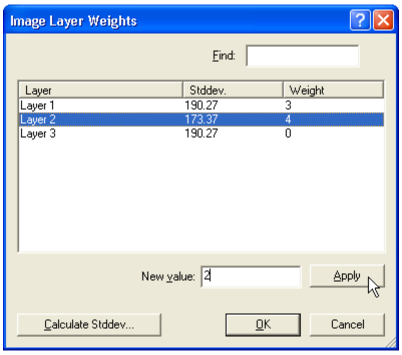


Figure 371: Image Layer Weights dialog box.

3. Select an image layer in the list. To select multiple image layers press Ctrl.
4. Enter a new weight in the **New value** text box and click **Apply**.

Options

5. Click the **Calculate Stddev** button to check the image layer dynamics. The calculated standard deviations of the image layer values for each single image layer are listed in the **Stddev.** column.
6. To search for a specific layer, type the class name into the **Find** text box.

13.3.48 Image Object Information

See related instructional section.

→ Investigate Single Image Objects by Using the Image Object Information on page 174

13.3.49 Image Object Table

See related instructional section.

→ Compare Multiple Image Objects by Using the Image Object Table on page 178

13.3.50 Import Image Layers

Two different **Import Image Layers** dialog boxes can open. Select one or more image files to import their image layers to a project.

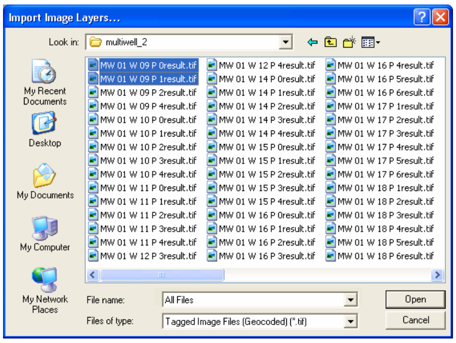


Figure 372: Import Image Layers dialog box.

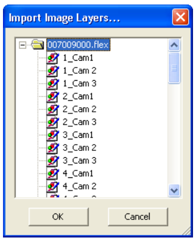


Figure 373: Import Image Layers dialog box for archive files.

See related instructional sections.

→ Create a New Project on page 44
→ Create a New Project Within a Workspace on page 87

13.3.51 Import Scenes

See related instructional section.

→ Import Scenes to a Workspace on page 58

13.3.52 Insert Expression

See related instructional section.

→ Insert an Expression on page 246

13.3.53 Layer Properties

The **Layer Properties** dialog box is used to edit an layer alias for a selected layer.

→ About Layer Aliases on page 54

1. To open the **Layer Properties** dialog box, do one of the following:
   - Create a new project.

→ Create a New Project on page 44

   - Modify an existing project.

→ Modify a Project on page 54

   The **Create Project** or **Modify Project** dialog box opens.

2. After importing layers, double-click or right-click an layer and choose **Edit**. The **Layer Properties** dialog box opens.

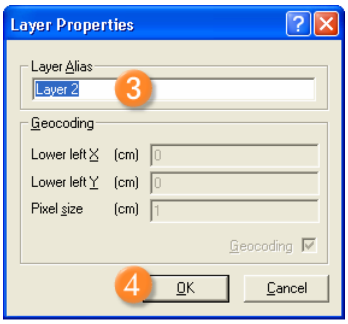


Figure 374: Layer Properties dialog box

3. Edit the **Layer Alias** 3.

   Edit the geocoding information. The following information can be included:

- X-coordinate of the lower left corner of the image
- Y-coordinate of the lower left corner of the image
- Geometric resolution
- Geocoding of the specified layer

4. Confirm with **OK** 4 to return to the superordinate dialog.

13.3.54 Legend

See related instructional section.

→ View Legend of Classes on page 380

13.3.55 Manage Customized Features

See the related instructional section.

→ Create Customized Features on page 269

13.3.56 Manage Layer Aliases

Edit layer aliases to avoid loading conflicts concerning the names of the image layers or thematic layers.

→ About Layer Aliases on page 54
→ Load a Process on page 144

1. To open the **Manage Layer Aliases** dialog box, choose in the main menu bar one of the following:
   - **Process** > **Manage Aliases** > **Image Layers Aliases**
   - **Process** > **Manage Aliases** > **Thematic Layers Aliases**

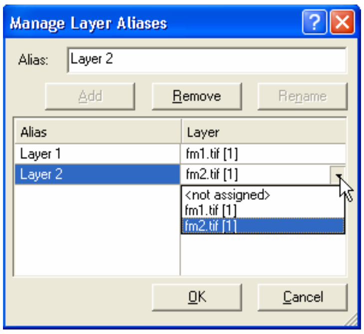


Figure 375: Manage Layer Aliases dialog box.

2. Select an layer alias in the left column. Do one of the following:
   - To add a layer alias, edit the **Alias** text field and press the **Add** button.
   - To remove a layer alias, press the **Remove** button.
   - To rename the layer alias edit the **Alias** text field and press the **Rename** button.
   - Click the drop-down arrow in the right column and select an layer to assign it to the selected layer alias in the left column.
3. To confirm press **OK**.

13.3.57 Manage Parameter Sets

See related instructional section.

→ Create a Parameter Set on page 266

13.3.58 Manage Variables

See the related instructional section.

→ Create a Variable on page 258

13.3.59 Membership Function

See related instructional section.

→ Use Membership Functions for Classification on page 213

13.3.60 Membership Function Parameters

See the related instructional section.

→ Edit Membership Function Parameters on page 217

13.3.61 Message Console

The **Message Console** display box shows warnings or instructive messages.

- The **Message Console** will automatically come up whenever an operation cannot be executed. There are four types of messages:
- Information messages are shown in green.
- Messages that need user interactions are marked in grey.
- Warning messages are shown in yellow.
- Error messages that immediately stop the process execution are shown in red.

1. To open choose **View > Message Console** on the main menu bar.

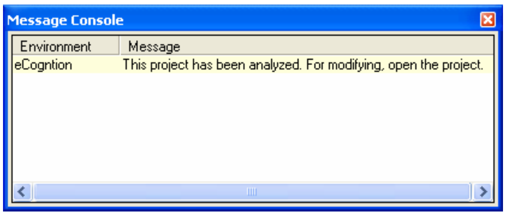


Figure 376: Message Console display box

The left column shows the **Definiens** software environment which prompted the message. The right column displays the output message and - if any - the last user interaction.

2. To clear the **Message Console** right-click the window and choose **Clean Console**.

Message Boxes

If you prefer to see a message box instead of the **Message Console** display box, proceed as follows:

1. Choose **Tools > Options** on the main menu bar.
2. Go to the **General** group and set **Show warnings as message box** to **Yes**.

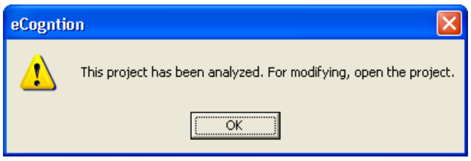


Figure 377: Message box

Messages that are displayed in message boxes are additionally listed in the **Message Console**. Thus you can retrace a sequence of messages.

13.3.62 Metadata Conversion

See related instructional section.

→ Convert Metadata to Provide it to the Feature Tree on page 52

13.3.63 Modify Project

See related instructional section.

→ Modify a Project on page 54

13.3.64 Options

The **Options** dialog box allows several optional settings concerning:

- General
- Display
- Manual Editing
- Output Format
- Developer
- Unit Handling
- Miscellaneous
- Engine
- Project Settings

1. To open choose **Tools > Options** on the menu bar.
2. Click in the right column to change settings. The default values are listed first.
3. Confirm with **OK**.

Tip

The following settings can be changed by using the ***set rule set options*** *algorithm:*

Distance calculation

Resampling Method

Evaluate conditions on undefined Features as 0.

→ ***Reference Book***

Options	Settings
General	
Show warnings as message box	**Yes**: Default. Messages are displayed in a message box and additionally listed in the **Message Console**. **No**: Messages are displayed in the **Message Console** where a sequence of messages can be retraced.
Ask on closing project for saving project or rule set	**Yes**: Default. Opens a message box to prompt saving before closing. **No**: Closes without asking for saving.
Save rule set minimized	**No**: Default. Does not saves features with the rule set. **Yes**: Saves the features used in the **Image Object Information** windows with the rule set.
Automatically reload last project	**No**: Starts with a blank project view when opening **Definiens Developer**. **Yes**: Useful if working with the same project over several sessions.
Display	
Annotation always available	**No**: The **Annotation** feature is not available for all image objects. **Yes**: The **Annotation** feature is available for all image objects.
Default image equalization	Select the default equalization method of the scene display in the project view. The Options dialog box allows several optional settings concerning: **Linear** **None** **Standard deviation** **Gamma correction** **Histogram** **Manual**
Display scale with	Select a type of scaling mode used for displaying scale values and calculating rescaling operations. **Auto**: Automatic setting dependent on the image data. **Unit (m/pxl)**: Resolution expressed in meters per pixel, for example, 40 m/pxl. **Magnification**: Magnification factor used similar as in microscopy, for example, 40x. **Percent**: Relation of the scale to the source scene scale, for example, 40%. **Pixels**: Relation of pixels to the original scene pixels, for example, 1:20 pxl/pxl.
Import magnification if undefined	Select a magnification used for new scenes only in cases where the image data has no default magnification defined. Default: **20x**

→ Message Console on page **451**

→ Edit the Image Layer Mixing on page 96

→ About Scales on page 49

Options	Settings
Display Scale Bar	Choose whether to display the scale bar in the project view by default.
Scale Bar default Position	Choose where to display the scale bar by default.
Instant Render Update on Slider	Choose whether to update the rendering of the Transparency Slider instantly. **No:** The view is updated only after the slider control is released or after it has been inactive for one second. **Yes:** The view is updated instantly as the slider control is moved.
Manual Editing	
Snapping tolerance [pxl]	Set the snapping tolerance for manual image object selection and editing. Default: **2**
Include objects on selection polygon outline	**Yes**: Includes all image objects that touch the selection polygon outline. **No**: Only includes image objects that are completely within the selection polygon.
Output Format	
CSV	
Decimal separator for CSV file export	**.** : Uses period as decimal separator.
Column delimiter for CSV file export	**;** : Uses semicolon as column delimiter.
Reports	
Date format for reports	**DD.MM.YYYY** or **MM/DD/YYYY** Select or edit the notation of dates used for exporting reports.
Developer	
Load Extension Algorithms	**No:** Deactivate algorithms created with the Definiens SDK (Software Development Kit). **Yes:** Activate algorithms created with the Definiens SDK.
Keep Rule set on closing Project	**Yes**: Keeps current rule set when closing a project. Helpful for developing on multiple projects. **No**: Removes current rule set from the Process Tree window when closing a project. **Ask**: Opens a message box when closing.
Process Editing	
Always do profiling	**Yes**: Always uses time measurement of processes execution to control the process performance. **No**: Does not uses time measurement of processes.
Action for double-click on a process	**Edit**: Opens the **Edit Process** dialog box. **Execute**: Executes the process immediately.
Switch to classification view after process execution	**Yes**: Shows the classification result in the project view window after executing a process. **No**: Current project view does not change.
Switch off comments in process tree	**No**: Comments in process tree are active. **Yes**: No comments in process tree.
Ask before deleting current level	**Yes:** Uses **Delete Level** dialog box for deletion of image objects levels. **No:** Delete image objects levels without reconfirmation. (Recommended for advanced users only.)

→ View a Scale Bar on page 107
→ See above.
→ Adjust Classification Transparency on page 110
→ Cut an Object Manually on page 395
→ Manual Editing on page 391
→ View Classification on page 109
→ Name the Process on page 139
→ Delete Image Object Level on page 197

Options	Settings	
Undo		
Enable undo for process editing operations	**Yes**: Enable undo function to go backward or forward in the operation's history. **No**: Disable undo to minimize memory usage.	→ Undo Process Editing Operations on page 139
Min. number of operation items available for undo (priority)	**5** Default. Minimum number of operation items available for undo. Additional items can be deleted if maximum memory is not exceeded as defined in **Max. amount of memory allowed for undo stack (MB)** below.	→ See above.
Max. amount of memory allowed for operation items (MB)	**25** Assign a maximum of memory allowed for undo items. However, a minimum number of operation items will be available as defined in **Min. length of undo stack** above.	→ See above.
Sample Brush		
Replace existing samples	**Yes**: Replace samples that have already been selected when the sample brush is reapplied. **No**: Do not replace samples when the sample brush is reapplied.	
Exclude objects that are already classified as samples	**No**: Applying the sample brush to classified objects will reclassify them according to the current sample brush. **Yes**: Applying the sample brush to classified objects will not reclassify them.	
Unit Handling		
Initialize unit conversion from input files	Used for image files with geocoding information. **Yes:** Use the unit in the image file. **No:** Ignore the unit in the image file and use the last settings selected in the Create Projects dialog box.	→ Create a New Project on page 44
Miscellaneous		
Automatically send crash report to Definiens	**No**: Default. Does not send crash report. **Yes**: Sends crash report supporting Definiens in improving products.	
Engine		
Raster data access	**Direct**: Accesses image data directly where they are located. **Internal copy**: Creates an internal copy of the image and accesses data from there.	
Project Settings		
Distance Calculation	Displays the distance calculation set using the Set Rule Set Options algorithm.	→ **Reference Handbook**
Current resampling method	Displays the resampling method set using the Set Rule Set Options algorithm.	→ Reference Handbook
Evaluate conditions on undefined Features as 0	Displays the value set in the Set Rule Set Options algorithm for the **Evaluate conditions on undefined Features as 0** parameter.	→ Reference Handbook
Polygons Base polygon threshold	Displays the degree of abstraction for the base polygons set in the Set Rule Set Options algorithm.	→ Reference Handbook
Polygons Shape polygon threshold	Displays the degree of abstraction for the shape polygons set in the Set Rule Set Options algorithm.	→ Reference Handbook
Polygons remove slivers	Displays the setting for removal of slivers from the Set Rule Set Options algorithm.	→ Reference Handbook

13.3.65 Pan Window

See related instructional section.

→ Pan Window on page 106

13.3.66 Plate View

See the related instructional section.

→ Manage Data in the Plate View on page 90

→ View Results in the Plate View on page 383

13.3.67 Process Profiler

See related instructional section.

→ Improve the Processing Performance on page 152

13.3.68 Process Tree

See related instructional section.

→ Organize a Process Sequence on page 141

13.3.69 Project History

See related instructional section.

→ Inspect the History of a Project on page 85

13.3.70 Rollback All Changes

See related instructional section.

→ Rollback All Changes of Projects on page 86

13.3.71 Sample Editor

See the related instructional section.

→ Work with the Sample Editor on page 226

13.3.72 Sample Selection Information

See related instructional section.

→ Assess the Quality of Samples on page 230

13.3.73 Select Classes

Select available classes.

This dialog box is used to select specific classes in the context of different dialogs. As this dialog box appears in the context of different dialogs, the window title bar may vary, but functionality remains the same. The window title bar may vary, but functionality is the same.

Similarly:
· Apply Standard Nearest Neighbor to Classes
· Select Classes for Shape Export
· Select Classes for Statistics
· Select Classes to Delete

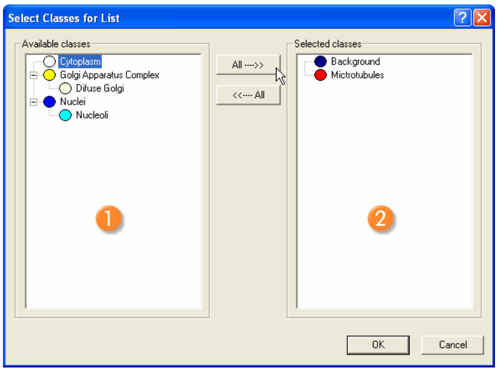


Figure 378: Select Classes for [function] dialog box.

- **Available Classes:** To switch to **Selected Classes** select the classes individually by clicking them or using the move **ALL--->>** button. With a right click it is possible to select a parent class including all child classes.
- **Selected Classes:** To deselect the classes individually double-clicking them or using the respective move **<<---ALL** button. With a right click it is possible to select a parent class including all child classes.

13.3.74 Select Features

Select features from the feature list.

Similarly with:
- Edit Standard Nearest Neighbor Feature Space
- Select Displayed Features
- Select Features to Display
- Select Features for Export
- Select Features for Export as Attributes
- Select Features for Statistics
- Select Variable for Parameter Set

The **Select Displayed Features** dialog box is used to choose features appearing in the **Image Object Information** window. Because this dialog box appears in the context of different dialogs, the window title bar may vary, but functionality remains the same.

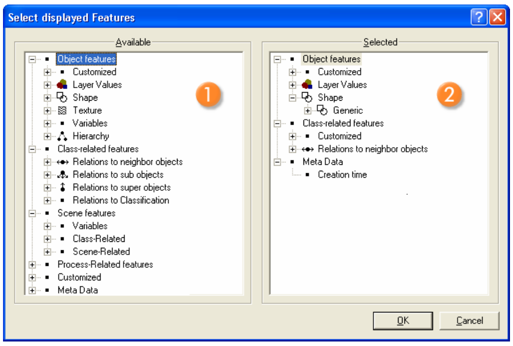


Figure 379: Select Displayed Features dialog box.

1. Display of all selectable features 1. Double-click a feature to select it.
2. Some features named **Create new '***feature***'**, for example, **Create new 'Min. pixel value'** must first be created before they maybe used. They require the setting of parameter values before the feature may be created. The **[feature]** dialog box with the same name as the feature (here **Min. pixel value**) will open.

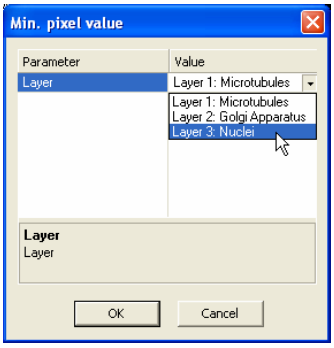


Figure 380: [Feature] dialog box for creating a new feature from the Select Displayed Features dialog box.
In this example for the feature **Min. pixel value**, the parameter **Layer** is set on the value **Nuclei**.

3. Depending on the feature and your project you have to set parameter values. Press **OK** and the new feature is listed in the feature tree. Additionally the newly created feature will be loaded into the **Image Object Information** window.

Context Menu Options

To display available context menu commands, right-click a feature in the **Select Displayed Features** dialog box.

> **Note**
>
> Only the available context menu options applicable to the selected feature will be displayed. They differ according to the selected feature and by different products.

- Display in **Image Object Information:** Deselect to remove the feature.
- **Edit:** Opens the **[feature]** dialog box with the same named as the feature. For example, if a feature from within the **Standard Deviation** feature tree grouping is selected, the **Standard Deviation** editing dialog box will open. It is used for editing parameter values of a feature.
- **Edit Object Value:** Opens **Edit Annotation** dialog box where you may change the selected image object value. The new value is displayed after its value in the feature tree as well as in the project view before the actual value.
- **Find:** Opens **Find and Replace** window. You may then substitute a created feature with another.

→ Find and Replace Rule Set Items on page 147

- **Delete:** Deletes the feature completely.
- **Create:** Opens a create dialog box, which is named after the *[Feature Group]*.
- **Create for All** [*Feature Group*]**:** Creates new features within the Feature View feature tree for all of the [*Feature Group*].
- **Manage Variables:** Opens the **Manage Variables** dialog box.
- **Manage Customized Features:** Opens the **Manage Customized Features** dialog box.
- **Save:** Saves feature as **.duf** file.
- **Load:** Import saved customized features from **.duf** files.

- **Mark as Shared:** Designates feature as **shared** in order to facilitate merging it within multiple rule sets.

> **Note**
>
> Once a feature is shared, it may not be edited.

- **Edit Unit:** Opens the **Select Unit** dialog box. You change the feature unit if available. The unit of a feature is displayed after its value in the feature tree.
- **Save to File:** Saves the dialog box contents as tabular **.csv** or **.tcsv** file.
- **Copy to Clipboard**: Saves the active dialog box to the clipboard.
- **Update Range**: When checkbox is selected, the complete range of the feature variables are displayed within the range boxes 4 5.

→ Investigate Threshold Values of a Feature on page 181

13.3.75 Select Single Feature

Select a single feature from the feature list.

Similarly with:
· Select Feature for...
· Insert Expression

It will be available in different dialog boxes, like: **Edit Process**, **2D Feature Space Plot** or **Class Description**. The window title bar may therefore vary.

→ Define the Image Object Domain on page 135
→ Analyze the Correlation of Two Features by Using the 2D Feature Space Plot on page 188
→ Insert an Expression on page 246

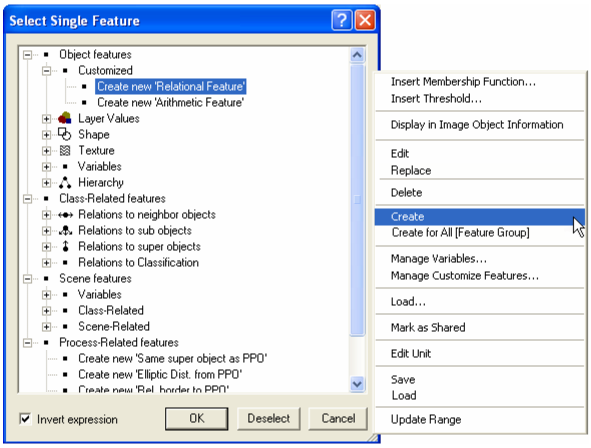


Figure 381: Select Single Feature dialog box with context menu after a right-click.

1. To select a feature from this feature list, double-click it or select it and click **OK**.
2. Depending upon which dialog box you opened, there may be the option of checking the box **Invert expression**. If selected, the inserted **Membership Function** will be added within the **Class Description** dialog box with the prefix **not** before the feature name.

Context Menu Options

To display available context menu commands, right-click a feature.

> **Note**
>
> Only the available context menu options applicable to the selected feature will be displayed. They differ according to the selected feature and by different products.

- **Insert Membership Function:** Opens the **Membership Function** dialog box in order to insert the feature.
- **Insert Threshold:** Opens the **Edit Threshold Condition** dialog box in order to insert the feature.
- **Display in Image Object Information:** Deselect to remove the feature.
- **Edit:** Opens the **[feature]** dialog box with the same named as the feature. For example, if a feature from within the **Standard Deviation** feature tree grouping is selected, the **Standard Deviation** editing dialog box will open. It is used for editing parameter values of a feature.
- **Find:** Opens **Find and Replace** window. You may then substitute a created feature with another.

→ Find and Replace Rule Set Items on page 147

- **Delete:** Deletes the feature completely.
- **Create:** Create a feature in the opening dialog box , which is named after the [*Feature Group*].
- **Create for All** [*Feature Group*]**:** Creates new features within the Feature View feature tree for all of the [*Feature Group*].
- **Manage Variables:** Opens the **Manage Variables** dialog box.
- **Manage Customized Features:** Opens the **Manage Customized Features** dialog box.
- **Mark as Shared:** Designates feature as **shared** in order to facilitate merging it within multiple rule sets.

> **Note**
>
> Once a feature is shared, it may not be edited.

- **Edit Unit:** Opens the **Select Unit** dialog box. You change the feature unit if available. The unit of a feature is displayed after its value in the feature tree.
- **Save:** Saves feature as **.duf** file.
- **Load:** Import saved customized features from **.duf** files.
- **Update Range:** When checkbox is selected, the complete range of the feature variables are displayed within the range boxes 4 5.

→ Investigate Threshold Values of a Feature on page 181

13.3.76 Select Variable for Parameter Set

See related section.

→ Select Features on page 457

13.3.77 Set Nearest Neighbor Function Slope

See the related instructional section.

→ Set Nearest Neighbor Function Slope on page 237

13.3.78 Snippets

See the related instructional section.

→ Save and Reuse Process Snippets on page 148

13.3.79 Start Analysis Job

See related instructional section.

→ Start Automated Image Analysis on page 363

13.3.80 Stitch Tile Results

See related instructional section.

→ Stitch Tile Results on page 376

13.3.81 Subset Selection

The **Subset Selection** dialog box is used to define a subset of the image to be analyzed.

> **Note**
>
> Project subset selections on projects that already have been analyzed does not display the classification results.
>
> Within the **Modify Project** dialog box, subset selection is deactivated on projects that already have been analyzed.

1. To open the **Subset Selection** dialog box, do one of the following:
   - Create a new project.
   - Modify an existing project.
   - Open a workspace, Right-click a project and choose **Open Subset** on the context menu.

   The **Create Project** or the **Modify Project** dialog box opens.
2. After importing image layers press the **Subset Selection** button. The **Subset Selection** dialog box opens.

→ Create a New Project on page 44
→ Modify a Project on page 54
→ Open a Project from a Workspace on page 94

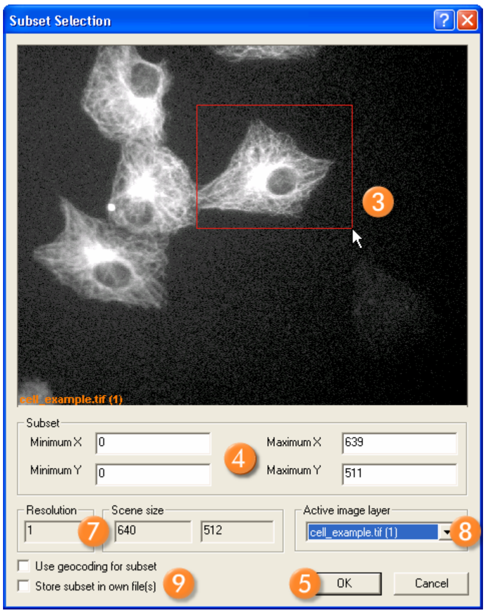


Figure 382: Subset Selection dialog box.

3. Click in the image 3 and drag to select a subset area.
4. Alternatively, you may enter the subset coordinates 4. You can modify the coordinates by typing.
5. Confirm with **OK** 5.

Options

(Only available for creating and modifying projects.)

6. You can clear the subset selection by clicking **Clear Subset** in the superordinate dialog box.

7 Displays the **Resolution** of the active image layer and the **Subset size**:

- **Minimum X** displays the X-coordinate of the left edge of the subset.
- **Maximum X** displays the X-coordinate of the right edge of the subset.
- **Minimum Y** displays the Y-coordinate of the lower edge of the subset.
- **Maximum Y** displays the Y-coordinate of the upper edge of the subset.

8 Change the displayed image layer. If you have inserted different layers, you can select the image layer to display.

9 Select **Store subset in own files** to store the subset to separate files. You can assign a new name to the subset files or accept their default name. If your data is georeferenced, activate the **Use geocoding for subset** to select the bounding coordinates based on the respective geographical coordinate system.

13.3.82 System Info

The **System Info** dialog box provides information about the hardware and software status of the machine as well as the memory usage of **Definiens Developer**.

To open, choose **Help** > **System Info** from the main menu bar.

13.3.83 Thematic Layer Attribute Table

See related instructional section.

→ View Thematic Layer Attributes on page 286

13.3.84 User Information

The **User Information** dialog box is used to include information about the author of the rule set.

1. Choose **File > User Information** on the main menu bar. The **User Information** dialog box opens.
2. Edit name, company, and copyright.
3. Click **OK** to confirm. When a rule set is saved, the user information will be stored as a property of the rule set.

13.3.85 View Settings

The **View Settings** window is used to set the visualization method for displaying image objects and image layers.

You have several options to view and to investigate your **Definiens** projects visually. Depending on the current workflow step of your image analysis task, you can use specific visualization methods enabling you to see what you are searching for.

→ View Data in Projects on page 93

1. To open the **View Settings** window, choose **View > View Settings** from the main menu.
   The **View Settings** window consists of two panes.

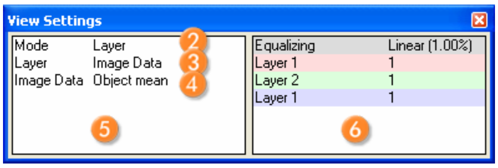


Figure 383: View Settings window.

2. On the left pane, click **Mode** 2 to open a context menu. Several modes of coloring the image objects can be selected; alternatively, you can use the toolbar buttons as indicated:

   - **Layer**:
     In the **Layer** mode, the objects are displayed without a fill color. The layer information itself is visible. To apply this mode, choose **Layer** or the respective toolbar button. When using this setting, the increasing level of abstraction with increasing object size can be observed.

     View Layer

   - **Samples**:
     In the **Samples** mode, the sample objects are displayed in their class color; the remaining objects are colored according to the **Layer** settings.

     View Samples

   - **Classification**:
     To view the classification result, use the **Classification** mode. Unclassified

     View Classification

objects are displayed according to the settings of the **Edit Highlight Colors** dialog box.

- **Classification Stability**:
  Graphical output of the classification stability of image objects of a selected level showing the difference in membership values of the best and second best class. All classified image objects are colored in a smooth transition from red (small differences) via orange to green (high differences).

- **Best Classification Result**:
  Graphical output of the best classification results for image objects of a selected level. All classified image objects are colored in a smooth transition from red (low membership values) via orange to green (high membership values).

- **Classification Membership**:
  Graphical output of the best classification results for image objects of a selected level. All classified image objects are colored in a smooth transition from black (low membership values) via grey to white (high membership values).

3 **Layer**: Fixed setting on **Image Data.** The grayed-out TTA Mask option is not active. If thematic layers are used in the project, they can be selected to be displayed in the project view.

4 **Image Data**: Switch between pixel view and object mean view. When viewing a project, you can switch between two views:

Pixel View or Object Mean View

→ Select and Deselect Image Objects on page 115

- **Pixel View** displays the image pixels. When the View Layer mode is selected, image objects are visible when selected by clicking on them.

- **Object Mean View** displays a fill color, calculated from the mean value of the pixels of the image object.

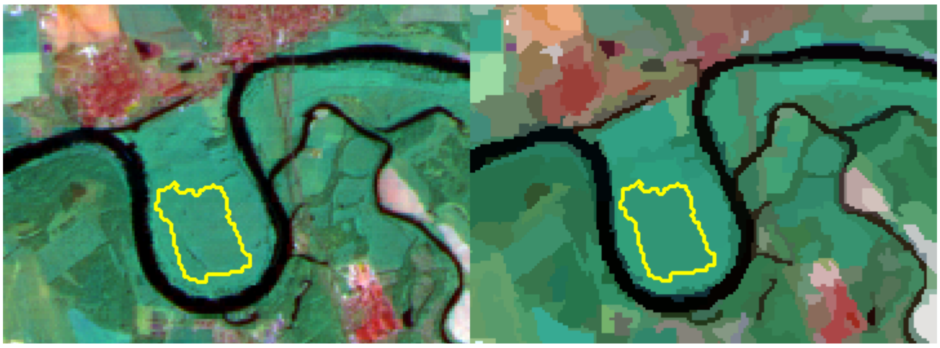


Figure 384: Sample project in Pixel View (left) and Object Mean View (right). The yellow outlines show a selected image object. (Image data courtesy of Ministry of Environmental Affairs of Sachsen-Anhalt, Germany.)

5 Double-click in the left pane of the **View Settings** to open the **Edit Highlight Colors** dialog box. Set the highlight colors and the display color of non-sample and unclassified objects.

→ Edit Highlight Colors on page 444

6 Double-click in the right pane to open the **Edit Image Layer Mixing** dialog box. You can define a color composition for the display of image layers and set equalizing options. Use different visualizations without changing the image data.

→ Edit the Image Layer Mixing on page 96

13.3.86 Workspace Properties

See related instructional sections.

→ Create a New Workspace on page 57
→ Move a Workspace on page 83

13.3.87 Workspace

In the **Workspace** window, you manage your image analysis task and administrate all relevant data. Using the **Workspace** window, you control the workflow of automated image analysis.

For detailed instructions, see the related instructional sections.

→ Create a New Workspace on page 57
→ Import Scenes to a Workspace on page 58
→ Save a Workspace on page 83
→ Manage Projects in a Workspace on page 56
→ View Data in Projects on page 93
→ Start Automated Image Analysis on page 363
→ Inspect the State of a Project on page 85
→ Accept and Reject Results on page 390
→ View Result Values in the Workspace on page 379

14 More Information About this Product

Contents in This Chapter

This chapter outlines information about this **Definiens** product provided by this guide, related documents, the user interface, and **Definiens** support.

14.1 About this Document

14.1.1 Information in this User Guide

Within the **Overview** chapter you can learn how to get a quick start in using **Definiens Developer**. In addition, the new functions of the current product version are listed with cross-references for easy access.

→ Overview on page 11

The **Basic Concepts** chapter introduces important elementary terms.

→ Basic Concepts on page 19

The **Workflow** chapter presents an overview of the basic operational sequence.

→ Workflow on page 32

The instructional chapters describe how to use the main functions to fulfill main tasks of image analysis with **Definiens Developer**. These instructions are sequenced according a basic workflow.

→ Start Definiens Software on page 37
→ Load and Manage Data on page 43
→ View Data in Projects on page 93
→ Three Chapters About Rule Set Development on page 123
→ Process Data on page 361
→ Examine Results on page 379
→ Export Data on page 401

The **User Interface Elements** chapter guides you through brief instructions for the following user interface elements:

→ User Interface Elements on page 410

- Menus
- Toolbars and command buttons
- Dialog boxes and windows

→ More Information About this Product on page 466

→ Glossary

→ . on page 473

This **More Information About this Product** chapter outlines the set of user documents and further available information.

The **Glossary** helps you with brief definitions of key words.

14.1.2 Access to this User Guide

Before the installation, you can access this **UserGuide.pdf** file in the downloaded installation folders. If you have a **Definiens** installation CD, you can find it in the root directory. The path on the CD is **...\UserGuide.pdf**.

After installation, you can choose one of the following:

- Open the **User Guide** from the **Start Menu.** Click
  Start > All Programs > Definiens *Current EII client (for example Developer, Architect, Analyst, Viewer)* **> User Documents > User Guide.**
- Choose **Help > Definiens Developer Help** on the main menu bar.
- Press the function key F1 or choose **Help > Developer Help** on the main menu bar.
- Consult the **UserGuide.pdf** in the installation directory. The default path is
  C:\Program Files\Definiens EII *Client (*for example, *Developer, Architect, Analyst, Viewer), Version number*
  \UserGuides\UserGuide.pdf.

14.1.3 On Screen Reading with Adobe Acrobat Reader

If you display this document on screen using **Adobe Acrobat Reader**, you can benefit from several linked references.

Display a linking table of contents by clicking the navigation tab **Bookmark** on the left side of the **Adobe Acrobat Reader**.

To search for a term of interest, use the **Search** function.

For installation of **Adobe Acrobat Reader**, you can use the **Definiens** installation CD. Alternatively, you may download it for free on **http://www.adobe.com**.

14.1.4 Notation

Warnings

Note or Caution ⚠
A caution warns you to protect against possible damage. A note informs you about restrictions.

Command Buttons

Most software commands are available by using both menu items and command buttons. Beside the description of the access to menu items, the respective command button is depicted in the margin.

Browse For Folder (example)

Keyboard Shortcuts

Default settings of keyboard shortcuts are denoted like this: Ctrl + S.

Placeholders (Variables)

Placeholders used as variables in file paths, or web addresses are noted in italics, for example, **C:\Program Files\Definiens Developer** *Version number***UserGuides\UserGuide.pdf**

Cross References

Cross References are indicated by an arrow.

→ About this Document on page 466

Figure Indicator Numbers

3 Figure indicator numbers are used to connect text passages to specific areas in graphics and vice versa.

Additional Information

Find Out More

Additional information is offered in boxes headed ***Find Out More*** *or* ***Tip****.*

Sample Images

Because **Definiens** software is used in different industries this document contains medical images as well as satellite images as sample data.

Process Symbols

Process symbols are used in the **Process Tree** representing the used algorithm by a dedicated icon.

Process Name

Code

`Code` is denoted as follows:

```
Within code, comments are highlighted green.
A code keyword is highlighted blue and bold.
```

Notation of Rule Sets

Rule set code is denoted as follows:

```
Subroutine 1
    Process 1
    Process 2 (Parent Process)
        Process 2.1 (Child Process)
        Process 2.2 (Child Process)
        Process 2.3 (Child Process)
            Process 2.3.1 (Child Process)
    Process 3
    Process 4  (Parent Process)
        Process 4.1 (Child Process)
Subroutine 2
    Process 1
```

14.2 Related Documents

This **Definiens** product comes with a set of user documents. This section lists and describes them.

14.2.1 Release Notes

Find information about new functions, known issues and late changes in the Release Notes.

Do one of the following:

- Before the installation, you can access **Release Notes** PDF files in the folders containing the installation files.
- After installation, go to the the installation directory. The default path is **C:\Program Files\Definiens**EII *Client or product name, Version number* **\UserGuides\Definiens Release Notes.pdf**.

14.2.2 Installation Guide

In the **Installation Guide** you can find all information about installation of **Definiens Developer** and license handling.

→ Installation Guide

Before the installation, you can access the **InstallationGuide.pdf** file in the downloaded folder containing the installation files. If you have a **Definiens** installation DVD, you can find it in the root directory. The path on the CD is **...\InstallationGuide.pdf**.

After installation, you can choose one of the following:

- Open the **Installation Guide** from the **Start Menu.** Click **Start > All Programs > Definiens** *EII Client or product name* **> User Documents > Installation Guide.**
- Consult the **InstallationGuide.pdf** in the installation directory. The default path is **C:\Program Files\Definiens** *EII Client or product name, Version number***\UserGuides \InstallationGuide.pdf**.

14.2.3 Reference Book

The **Reference Book** lists detailed information about algorithms, features, and gives general reference information. For individual image analysis and rule set development you may keep a printout ready at hand.

→ Reference Book

To access, you can choose one of the following:

- Access the **Reference Book** from the **Start Menu.** Click **Start > All Programs > Definiens Developer > User Documents > Reference Book.**
- Choose **Help > Reference Book** on the main menu bar.
- Alternatively, go to the installation directory. The default path is **C:\Program Files\Definiens Developer** *Version number***\UserGuides \ReferenceBook.pdf**..

14.2.4 Tutorials and Guided Tours

→ Get a Quick Start on page 14

Tutorial Basic Knowledge

To get a brief idea about the workflow of image analysis we recommend the **Tutorial Basic Knowledge 1**. Using sample data, you perform a very simple process sequence.

Go to the folder **Examples**, which is installed in the **Definiens Developer** installation directory. The default path is

C:\Program Files\Definiens Developer *Version number***\Examples\ TutorialBasicKnowledge1**.

Guided Tour Mapping Impervious Surface

The guided tour **Mapping Impervious Surface** presenting a geographic example of image analysis. It includes sample data to follow in practice. This guided tour focuses on the basic steps involved in developing a rule set using **Definiens Developer** and submitting a set of images for batch processing.

Guided tours are installed separately. Consult the **Installation Guide** for details.

→ Installation Guide on page 469

14.2.5 Definiens Application and Extension User Guides

User guides of applications products like **Definiens Cellenger** or extension products like **Definiens Data Management** provide product specific instructions and reference information.

→ Start Definiens Software on page 37

If application or extension functionality are available you can access the corresponding user guide by one of the following:

- Open the corresponding **User Guide** from the **Start Menu.** Click **Start** > **All Programs** > **Definiens** *Current EII client (for example, Developer, Architect, Analyst, Viewer)* > **User Documents** > *Product* **User Guide.**
- Choose **Help** > *Product* **Help** on the main menu bar.
- Consult the *Product* **UserGuide.pdf** in the installation directory. The default path is **C:\Program Files\Definiens** EII *Client (for example Developer, Architect, Analyst, Viewer), Version number***UserGuides**
Product **UserGuide.pdf**.

14.2.6 SDK User Guides and References

The software development kit (SDK) of **Definiens Developer** includes three application programming interfaces (API). Each API comes with a user guide document (**.pdf**) and a reference help file (**.chm**).

→ About the Software Development Kit (SDK) on page 18

After installation of the software development kit (SDK), you find them in the help folder of the corresponding API directory under the default path:

C:\Program Files\Definiens Developer *Version number***SDK**

- Engine API

→ **.\Engine\Help\EngineAPI_UserGuide.pdf**
→ **.\Engine\Help\EngineAPI _Help.chm**

- Data I/O API

→ **.\DataIO\Help\DataIOAPI_UserGuide.pdf**
→ **.\DataIO\Help\DataIOAPI _Help.chm**

- Automation API

→ **.\Automation\Help\AutomationAPI_UserGuide.pdf**
→ **.\Automation\Help\AutomationAPI _Help.chm**

14.3 Help on the User Interface

14.3.1 Tool Tips

The user interface displays tool tips for all command buttons on the tool bars.

Go with the pointer to a command button for an instant and the screen tip will appear.

> **Note**
>
> If no tool tip is highlighted, check the customization options.
>
> → **Customize** on page 439

14.3.2 Context-Sensitive Help

Many dialog boxes can display context-sensitive **Help** providing information about dialog box elements.

Context-sensitive Help

1. If available, click the question mark in the upper right corner of a dialog box. A question mark is added to the pointer.

Help pointer

2. Go with the **Help** pointer to the dialog box element of interest and click it. A pop-up dialog box will display instructional information.

14.3.3 Help Keyboard

Intermediate users may use the **Help Keyboard**, which provides very brief descriptions of all commands. Choose **Help** > **Help Keyboard** on the menu bar.

14.4 Support

Feel free to contact us via internet form on the website **http://www.definiens.com/support/index.htm**.

15 Glossary

.

.

(This glossary section lists file format extensions following the period, for example **.txt** for text documents.)

.Csv File

Comma-separated value file. List file that can be opened with most spreadsheet applications like **Microsoft® Excel**.

.Dax file

A solution file.

→ Solution on page 489

.Dbf File

A dBase file used for storing attribute data of shapefiles.

→ Shapefile on page 489

.Dcp File

A process file.

→ Process on page 485

.Dkb File

A class hierarchy file.

→ Class Hierarchy on page 476

.Dlx file

An action library file.

→ Action on page 474

.Dpj File

A workspace file.

→ Workspace on page 490

.Dpm File

A workspace information file.

→ Workspace on page 490

.Dpr File

A **.dpr** file or project file contains all information about the saved image analysis project beside the image files. Image files are referenced by the **.dpr** file.

→ Project on page 486

.Dps File

A settings file that stores view settings of workspaces, of projects or global.

.Duf File

A **Definiens** feature files.

→ **Feature** on page 479

.Psf file

A parameter set file used for both storing parameter sets and calibrations.

→ **Parameter Set** on page 485
→ **Calibration** on page 475

.Dbf File

A shapefile.

→ **Shapefile** on page 489

A

Abstract Class

An abstract class is an active class which is not used to classify image objects directly. In the internal logic of the inheritance hierarchy, any class that has applicable child classes is defined as an abstract class.

Accuracy Assessment

Methods to assess the quality of a classification.

Action

An action represents a predefined building block of an image analysis solution. Configured actions can perform different tasks like object detection, classification or export of results to file. Actions are sequenced and together they represent a ready-to-use solution accomplishing the image analysis task.

A configured action consists of a set of processes with defined parameters. Standard action definitions, which are just unconfigured actions, are provided in action libraries.

The file name extension of action libraries is **.dlx**.

Active Class

If a class is set inactive, it will be ignored during the classification process.

Additive Color

See **RGB**.

Alias

An alias is a pseudonym which can be assigned to an object like an image layer, an thematic layer or an image object level to replace its standard name. Aliases are used to simplify identification and working with named objects.

→ **Layer Alias** on page 482

Analysis Engine Software

Software component of the of the **Definiens eCognition Node Software** executing rule sets on projects and retrieves results.

B

Brightness

Sum of the mean values of the layers containing spectral information divided by their quantity computed for an image object.

C

Calibration

A calibration stores **General Settings** properties of a solution as Definiens **Parameter Set** file with the extension **.psf** . Thus you can save and provide common settings for for example, common image readers as part of an application.

Candidate

Image objects some algorithm use during execution for modification of the **seed** image objects.

→ Seed on page 488

Child Class

A child class is a class with a superior class, either in the inheritance or in the groups' hierarchy or both. Furthermore, it is used to differentiate a superior class, a so-called parent class. In the inheritance hierarchy, a child class inherits the class description from its parent class, whereas in the groups hierarchy child classes are grouped to a parent class of superior semantic meaning.

Child Domain

An image object domain defined by using one of the four local processing options:

→ Image Object Domain on page 481
→ Child Process on page 475

- Current Image Object
- Neighbour objects
- Subobjects
- Superobject.

A process that is set do run on a child domain is called a child process.

Sometimes referred to as **subdomain**, but avoid this term because it might be misleading.

Child Process

A process grouped on a level beneath a parent process within a process hierarchy. Its image object domain may be defined as a child domain.

→ Parent Process on page 485
→ Child Domain on page 475

Class Description

The class description is an essential part of each class. In the class description, the characteristics of image objects are defined as a member of the respective class. This feature description is adjusted by means of membership expressions. In addition, general information like a class name or a class color is specified in the appropriate dialog.

Class

A class is a category of image objects. It can both be used to simply label image objects or to describe its semantic meaning.

Class Hierarchy

The class hierarchy is the knowledge base for the classification of image objects. It contains the sum of all classes with their specific class descriptions. The classes can be structured hierarchically and displayed in an inheritance hierarchy and a groups hierarchy.

Class hierarchies can be saved to file. The file name extension of action libraries is **.dkb**.

Class-Related Features

Features that refer to the classification of other image objects. They take the actual classification of networked image objects into account in their evaluation process, e.g., relations to neighbor objects, to subobjects and superobjects.

Classification

Classification is a procedure that associates image objects with an appropriate class.

Through the process of classification, each image object is assigned to a certain (or no) class and thus connected with the class hierarchy. The result of the classification is a network of classified image objects with concrete attributes, concrete relations to each other and concrete relations to the classes in the class hierarchy..

Classification Stability

The difference between the best and the second best class membership of an image object.

Client

See related entry.

→ **Definiens Enterprise Image Intelligence Client** on page 478

CNL

see **Cognition Network Language**.

Cognition Network Language

Definiens software provides a unique language for developing advanced image analysis algorithms. This Cognition Network Language (CNL) can be accessed easily by a well-structured, easy to use click-and-drop user interface.

The development of image analysis solutions is carried out interactively. The impact on the results of each development step can be immediately visualized and traced in the displayed image.

The developed rule sets transparently document the applied analysis strategy to ensure a reliable assessment of the strategy. This ensures a rapid development and optimization of the Cognition Network Language for powerful, robust and reliable solutions for your image analysis tasks.

The developed rule sets can be saved for automated image analysis both in stand-alone and distributed environments using **Definiens eCognition Server**.

Color Criterion

The homogeneity criterion used by **Definiens's** multiresolution segmentation is comprised partly by homogeneity concerning the color (color criterion) and homogeneity concerning the shape (shape criterion) of the image objects to be merged. The color criterion minimizes the standard deviation calculated from the spectral values of an image object.

→ **Digital Value** on page 478
→ **Pixel Value** on page 485

Compactness Criterion

The shape criterion in **Definiens's** multiresolution segmentation is itself comprised of a smoothness criterion and a compactness criterion. In a raster the ideal compact form of an object is a square. The compactness criterion minimizes the deviation from the ideal compact form.

Connector

A piece of software enabling **Definiens** software to transfer specific types of data from or to the environment. A connector describes file driver, data format and structure required for data transmissions like import or export. A connector provides an pre-defined import template that can be selected when importing scenes.

Context Menu

A menu opening after a right-click.

Conversion Table

The conversion table defines the linkage between classes in the class hierarchy and classes from a TTA mask. It can be displayed using the **Conversion Table** dialog box.

Coordinates

Numbers defining positioning marks in images.

→ **Geocoding** on page 480

Customized Features

Feature that can be suited to the needs of a specific image analysis task. New arithmetic and relational features can be created by the user.

D

Data Fusion

See **Multisource Data Fusion**.

Definiens Application

A **Definiens** application can extend each of **Definiens Enterprise Image Intelligence** platform products providing industry- or user-specific ruleware and functions.

Definiens applications enlarge the capabilities of both the **Definiens Enterprise Image Intelligence (EII)** Clients (**Definiens Developer, Architect, Analyst, Viewer)** and the processing environment **Definiens eCognition Server**.

Definiens applications enable users of clients like **Definiens Architect** to create ready-to-use solutions for their specific image analysis problem. Started with one of the **Definiens Image Intelligence Clients**, an application completes the client functionalities by using particular ruleware, controls, and workflows needed for industry- or user-specific tasks.

Definiens eCognition™ Server

A **Definiens** software package that provides a scalable high-performance computing environment for distributed high-throughput image analysis. It is the core component of the **Definiens Enterprise Image Intelligence Suite**.

The **Definiens eCognition™ Server** software package commonly is set up on multiple computing facilities. This setup includes the **Definiens License Server Software** and a so-called **Definiens eCognition Grid**.

The package of all **Definiens** server side software that is used within the **Definiens eCognition Grid** including the **Definiens License Server Software** is called **Definiens eCognition Server**.

Definiens Enterprise Image Intelligence Client

A piece of software that provides the different role based user interfaces for the **Definiens eCognition Server**, which is the core component of the **Definiens Enterprise Image Intelligence Suite**. As a client, it accesses remote services from another piece of software, called a server.

→ Definiens eCognition™ Server on page 478

Diagonal Pixel Neighborhood

One of two definitions of neighborhoods: plane 4-neighborhood and diagonal 8-neighborhood on a pixel raster. The diagonal neighborhood considers all 8 adjacent pixels surrounding a specific pixel to be neighbors.

Digital Value

Information contained within imported image object layers is processed and stored as digital information. The nature of these maybe visible color, IR/UV radiation, radar, or spatial. Each pixel or digital value is assigned an intensity, normally an integer.

→ Color Criterion on page 477

Grayscale images have a pixel value of 8-bit data value (range 0-255) or of 16-bit data value (range 0-655535).

Color images may have 8-bit, 16-bit, 24-bit, or 30-bit colors. 24-bit colors are known as true colors which consist of three 8-bit pixels, representing red, green, and blue intensity.

Domain

See related entry.

→ Image Object Domain on page 481

Driver

A driver is a software component responsible for reading and writing data.

The most common type of driver is the file driver, which is used to read and write a specific file format. For example, the FRM file driver supports the Incell 3000 proprietary image format. File drivers work in tandem with connectors to implement the required image import function on batches of data.

→ Connector on page 477

Another type of driver is the metadata driver. Metadata drivers, can read and exchange metadata information from an external source and pass this through the engine for analysis. This metadata can be passed through and written to the analysis results. For example, a bar code of a cell biology plate that is used to organize or report on data can be passed through from the import to export steps.

E

EII

Enterprise Image Intelligence.

→ Definiens Application on page 478

Expression

Expressions are used to build fuzzy or crisp class descriptions. They are distinguished as to features and terms, and serve different purposes such as the formulation of concepts and knowledge, acting as classifiers or performing logical operations on other expressions.

F

Feature

An attribute, that is a piece of information concerning an object, for example measurements, attached data or values. In **Definiens** software, such an attribute is called feature.

Features are used as source of information to define the inclusion-or-exclusion parameters when classifying image objects. Features are only readable, not writable.

There are two major types of features:

- **Object features** are related to image objects. Object features describe spectral, form, hierarchical, or other properties of an image object, for example its area.
- **Global features** are not related to an individual image object, for example the number of image objects of a certain class.

A feature can be saved as **.duf** file.

Feature View

Feature View is a display function that visualizes a feature over the entire image view. It gives a visual overview of the properties of a specific feature.

Floating Point Number

An approximate representation of a real number. It has been developed for the numerical calculations with a computer.

Form Feature

Form features characterize image objects by their shape.

Fuzzy Logic

Fuzzy logic is a mathematical approach to quantifying uncertain statements. In **Definiens**, fuzzy logic is the formal basis of classification. The basic idea is to replace the two strict logical statements **0 ~ is not member of** and **1 ~ is member of** by the continuous membership range of [0...1] where 0 means does definitely not belong to and 1 means belongs fully to. By transferring each feature value into a fuzzy value between 0 and 1, fuzzy logic allows the transfer of features of very different dimensions into a uniform evaluation system. One important advantage of fuzzy logic classification is the possibility of including concepts and knowledge in class descriptions. Another is the full transparency of decisions: each single step of a class evaluation can be retraced in detail.

G

Geocoding

Geocoding is the assignment of positioning marks in images by coordinates. The position marks serve as geographic identifiers. but geocoding is helpful for life sciences image analysis too. Typical examples include working with subsets, at multiple magnifications, or with thematic layers for transferring image analysis results.

Geocoding information is automatically detected by **Definiens Developer** or can be entered manually. Images without geocodes create automatically a virtual coordinate system with 0/0 upper left and the unit of 1 pixel. For such images, geocoding represents the pixel coordinates instead of geographic coordinates.

GIS

Geographic information system is an integrated accumulation of information technology and geographic data for capturing, managing, analyzing, and displaying geographically referenced information.

→ Thematic Layer on page 490

Group Box

A main area in a dialog box bordered by a thin line.

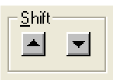


Figure 385: Group box, here containing two Shift buttons.

Groups

The semantic groups' hierarchy is a hierarchy of classes, which is determined by semantic membership of classes. It can be edited in the **Groups** register tab of the **Class Hierarchy** window.

H

Hierarchy Features

An image object can be characterized by its embedding in the image object hierarchy using hierarchy features. See **Class-Related Features**.

Homogeneity Criterion

In **Definiens's** multiresolution segmentation the homogeneity criterion is used to determine which heterogeneity attributes of image objects are to be minimized as a result of a segmentation run. Three criteria are used to describe image object heterogeneities: color criterion, smoothness criterion and compactness criterion. The smoothness and the compactness criteria are additionally summarized to the shape criterion. The composition of the entire homogeneity criterion based on the specific criteria can easily be defined by assigning weights to each of the specific criteria.

I

If–Then Rules

Fuzzy rules are **if**–**then** rules.If a condition is fulfilled, an action takes place. The following rule could be defined: **If** a feature x is low, **then** the image object should be assigned to **class a**. In fuzzy terminology this would be written: If feature x is a member of fuzzy set **low**, then the image object is a member of **class a**.

Image Layer

Definiens Developer allows the import of image raster layers and thematic raster layers. In contrast to thematic layers, image layers contain continuous information: similar spectral intensity stands for similar meaning. Besides spectral information, image layers can also contain a wide range of other information, e.g., geographical elevation models.

Image Layer Mix

For visualization purposes, **Definiens Developer** offers a variety of combinations in which image layers can be mixed. These can be defined in the **Edit Layer Mixing** dialog. By switching between the image layers in the image layer stack the chosen default mixings can be applied to different sets of layers.

Image Object Domain

The image object domain defines the region of interest on which an algorithm should be performed. The image object domain is defined by a structural description of the corresponding subset. Examples for image object domains are the entire image, an image object level or all image objects of a given class.

A image object domain is a set of image objects which can be unclassified or classified. Before any classification, the image object domain is a set of unclassified image objects.

Some algorithms need a second image object domain to define a second set of objects. See **Seed** for further information.

Image Object

A group of connected pixels in a scene. Each image object represents a definite region in an image. Image objects can provide information about this definite image region.

Image Object Hierarchy

Two or more image object levels of an project build an image object hierarchy.

The entirety of image objects is organized into a hierarchical network of image objects. Such a network consists of one or more image object levels, from fine resolution on the lowest image object level to coarse resolution on the highest image object level.

Within the image object hierarchy, each image object is linked to its neighbors, its superobject, and its subobjects.

On its superlevel, every image object has only one image object, the superobject. On the other hand an image object may have—but is not required to have—multiple subobjects.

Image Object Level

A layer of image objects within the image object hierarchy. It serves as an internal working area of the image analysis.

Image Object Primitive

A maximum sized group of neighbored pixels of defined characteristics. As the result of a segmentation image object primitives are basic unclassified image objects. Concerning further processing, image object primitives serve as both building blocks and information carrier.

Image Reader

A camera, instrument, image acquisition facility or other hardware facility that produces image data.

Inheritance

Inheritance is a general concept extensively used in object oriented software languages. In **Definiens**, inheritance plays a significant role in passing down class descriptions from parent classes to child classes. It can be edited in the **Inheritance** tab of the **Class Hierarchy** window.

K

Knowledge Base

A collection of characteristic attributes (image object features) and rules (class hierarchy, classifications, and algorithms) utilized to perform an image analysis task.

L

Layer

See **Image Layer** or **Thematic Layer**.

Layer Alias

A layer alias is a name for an image layer or thematic layer that you can choose to replace the standard layer name. You can use a descriptive name for easier identification of an image layer or thematic layer.

The basic function of layer aliases is to make the whole workflow in **Definiens** more transparent and independent from the initial input data. Thus, when assigning and using layer aliases, all steps of your analysis become more transferable, since all layer-sensitive operations and features can optionally refer to the layer aliases or to the layers themselves. Hence, processes, classes and customized features become more independent and flexible.

Layer Values

Layer values evaluate the first and second statistical moments (mean and standard deviation) of an image object's pixel values and relationships to other image objects' pixel values in each image layer. Use them to describe image objects in terms of information derived from their spectral properties.

Level

See related entry.

→ Image Object on page 481

Logical Term

Logical terms are used to combine and evaluate two or more expressions in a class description in terms of fuzzy logic. The following logical fuzzy operators are available:

- **and (min)**
- **or (max)**
- **and (*)**
- **mean (arithm.)**
- **mean (geo)**

M

Magnification

A specific scale mode representing a scaling factor of image enlargement as used in microscopy, for example 40x. Scenes of different magnification scales have different pixel resolutions.

→ Scale, Scaling on page 488

In contrast, the magnification zoom value represents a zoom factor used for display within the project view. The magnification zoom value does not affect the scale.

Membership Function

Membership functions are a simple method to translate an arbitrary feature value into a membership degree between [0 ... 1], indicating the membership of a class. Multidimensional membership functions are used in the nearest neighbor classifier. Membership functions are especially suited to introduce existing knowledge or concepts into the classification.

Metadata Conversion

Preparation of a selection of available metadata provided by the feature tree. This allows the usage of metadata in ruleware development.

Multiresolution Segmentation

A basic procedure for largely knowledge-free and unsupervised segmentation of homogeneous image object primitives in any chosen resolution. It was developed to work even on highly textured image data.

Multisource Data Fusion

Data of different sources can be synchronized and brought into a meaningful relationship.
Objects resulting from a segmentation of one layer can be evaluated using information from another or several other image layers.

N

Nearest Neighbor

Nearest neighbor (NN) classifies image objects in a given feature space based on given samples for the classes concerned. The distance in the feature space to the nearest sample object of each class is calculated for each image object. The image object is assigned to the class represented by the closest sample object.

NN

See related entry.

→ Nearest Neighbor on page 484

No Data Value

Exclude certain areas of the image from being analyzed by defining a pixel value. Each pixel with this value will be **No Data Value**.

O

Object

See related entry.

→ Image Object on page 481
→ Thematic Object on page 490

Object Feature

See related entry.

→ Feature on page 479

Object Level

See related entry.

→ Image Object on page 481

Outline

The borders of an image object based on raster objects.

P

Parameter Set

A parameter set is a superset of variables which allows you to save and reload the specific parameter configuration of a process tree independently of the rule set. This enables the adoption of an algorithm to a specific problem without changing the entire rule set. Parameter sets are used for developing action definitions.

Parent Class

A superordinate class which is further differentiated by child classes in the class hierarchy. In the inheritance hierarchy, parent classes pass on their class description to their child classes. In the groups hierarchy, parent classes summarize child classes to a class of superior meaning.

→ Class Hierarchy on page 476

Parent Process

A parent process is used for grouping child processes together on a hierarchy level beneath the parent process within a process hierarchy. The typical algorithm of the parent process is the **Execute child processes** algorithm.

→ Child Process on page 475

Parent Process Object (PPO)

A parent process object (PPO) is an image object defined in a parent object that a child process refers to.

→ Child Process on page 475

Pixel

A pixel is the smallest unit of an image. The term pixel is a contraction of **picture element**.

Pixel Level

The lowest level in the image object hierarchy, whereby information is represented in single pixels. The information derived from the pixel level is only used for segmentation of image objects in the first level of the image object hierarchy. All other procedures work directly on image objects.

Pixel Value

See **Digital Value.**

Polygon

The borders of vector objects are called polygons.

PPO

See **Parent Process Object**.

Process

An individual operation of an image analysis routine.

A single process executes an algorithm on an image object domain. Thus, it is the main working tool for developing rule sets. A single process is the elementary unit of a rule set providing a solution to a specific image analysis problem. Processes are the main working tools for developing rule sets.

> **Note**
>
> In **Definiens Developer,** the term **Process** is used for both a single process and a process sequence.

Processes can be sequenced and grouped together. Processes can be saved as **.dcp** file.

Project

A wrapper for all information that is related to one scene (e.g. images, rule-sets). It stores references to at least one image layer and related result information from image analysis expressed by classified image objects. A scene holds metadata like layer aliases or unit information. Optionally, a project can enclose thematic layers.

In **Definiens** software, a project represents a scene. A project is saved as a **.dpr** project file.

R

Raster Data

Image information expressed as a matrix of cells or pixels.

Image data can only be imported as a raster layer. Thematic layers can be raster layers or vector layers, which become rasterized during the project creation process and are also raster layers internally.

Raster data can be stored in raster files.

Resolution

Specific scale mode of raster image data defining the actual distance in reality per pixel, e.g. 10 meter per pixel.

→ Scale, Scaling on page 488

Result

Results of an image analysis can be both numerical and visual. Numerical results are statistics and result values. Visual results are represented by the classification of image objects.

Result Value

Numerical information extracted by image analysis from one or multiple image objects.

RGB

Red, Green, Blue (RGB) are the three basic colors of the light, which are mixed additively to produce any other color, called additive color. Colored images are frequently stored as a layered sequence of RGB triplets. An RGB image file contains three image layers, which are Red, Green, and Blue.

In **Definiens Developer**, the RGB colors are assigned to the layers of a multilayered image. They are displayed together as additively mixed colors. Thus, the color of an image area informs the viewer about the layer concerned. Depending on the imported image data, the displayed colors might not represent the coloring of the image area in reality.

Rule Set

A set of at least one process solving a defined image analysis task. If available, a rule set includes the used class hierarchy, the used parameter sets and the used meta data conversion.

Rule Set Item

A secondary component of a rule set. Rule set items are:

- Classes
- Features
- Image object levels
- Variables
- Image layers
- Thematic layers

Ruleware

A piece of software used for image analysis. Ruleware is based on the Cognition Network Language (CNL) and consists of one or multiple rule sets and optionally additional functionalities.

Examples:

- A set of processes → Process on page 485
- An action → Action on page 474
- An action library
- A solution → Solution on page 489

Using the different **Definiens Enterprise Image Intelligence** products, ruleware can be created, configured and executed.

S

Sample

See **Sample Object**.

Sample Object

A sample object is an image object which is assigned to be a typical representative of a certain class. Sample objects can be used as anchors for nearest neighbor classification or to display and compare the feature distribution of certain classes in the **Sample Editor** window.

Sample View

The sample view mode is a display mode in the **View Settings** window, which displays assigned sample objects with the color of their respective class. All other image objects are displayed in the actual color mix of the layer mean mode or in the non-samples highlight color.

Scale Parameter

An important parameter of the **Multiresolution Segmentation** algorithm, used to determine the upper limit for a permitted change of heterogeneity throughout the segmentation process. The scale parameter determines the average image object size.

Scale, Scaling

A value describing the ratio of depicted distance to the actual distance in the reality.

→ **Resolution** on page 486
→ **Magnification** on page 483

Scaling means the modification of a scale with the consequence of another magnification/resolution. Images at a smaller scale have a smaller file size.

Unlike zooming, scaling images, does not affect only the display of an image e.g. in the project view; moreover, scaling changes the number of pixels and consequently the information quality of image data.

Scales can be expressed by using different scaling modes, for example 20m/pxl (resolution), 20x (magnification), 20% (percent in relation to the source scene), or 1:20 pxl/pxl (pixel relation to the source scene). Note that different scaling modes cause different results.

Scene

Combined input image data from an **Definiens** project or workspace. Each scene representing one set of data used in **Definiens** software is managed in a project.

A scene consists of at least one image layer of one image file. When working with a combination of different data, a scene usually contains several image files with multiple image layers and optionally thematic layers.

A scene can include additional information related to the image content, such as metadata, geocoding, or geo information.

Depending on the image reader or camera, a scene combines multiple views of the same piece of reality, each of them in a separate layer. To put it simply, a scene can include several images of the same thing, each of them providing different information.

Seed

With some process-algorithms you need to define a second image object domain. The domain specified within the **Edit Process** dialog box will then be referred to as the **Seed** image object domain. The second domain, which can be the same as the **Seed** image object domain, will be the **Candidate** image object domain. The image object that would result e.g. by merging the seed with a candidate is called the **Target** image object.
In general, the **Seed** domain defines the image objects the algorithm is applied to, whereas the **Candidate** domain defines a set of image objects the algorithm uses during execution.

For example, the algorithm Image object fusion merges objects of the Seed domain with objects of the Candidate domain.The image object that would result by merging the seed with a candidate is called the target image object.

Segmentation

In image analysis, segmentation is the subdividing of an digital image into smaller partitions according to given criteria.

The basic step of any **Definiens** image analysis is a segmentation of a scene (representing an image) into image object primitives. Thus, segmentation is the subdivision of an image into separated regions represented by basic unclassified image objects called image object primitives.

Within the **Definiens** technology, each operation that creates new image objects is called segmentation.

Shapefile

A vector data file format for storage and exchange of location, shape and attributes of objects. It is commonly used in geographic information systems (GIS).

→ Vector File on page 490
→ GIS on page 480

Shared Class

Shared classes are classes that can be shared among several rule sets. They must not be modified in any way.

Shared Object

Shared objects are objects that can be shared among several rule sets. They must not be modified in any way.

Sliver Polygon

A gap or small thin polygons along the edges of topologically overlaid polygons, usually unwanted artifacts. Also called splinter polygons.

Solution

A solution provides an image analysis rule set configured for a specific type of image data. A solution is assembled from predefined building blocks called actions.

A solution can be saved as **.dax** file.

Subdomain

See related entry.

→ Child Domain on page 475

Subobject

An image object that has a superobject within an image object hierarchy.

→ Image Object Hierarchy on page 482

From the perspective of an image object: another image object that covers part or all of the pixels of the image object itself, and lies one level (or n levels) below the image object. It is then referred to as **subobj(1)** or **subobj(n)**.

Subroutine

A subroutine is a separate part of a rule set cut off from the main process tree to apply it on a copy of the main scene.

Superobject

An image object that has subobjects within an image object hierarchy.

From the perspective of an image object: an image object that covers at least the pixels of the image object itself, and lies one level (or n levels) above the image object. It is then referred to as **superobj(1)** or **superobj(n)**.

→ Image Object Hierarchy on page 482

T

Thematic Layer

A vector file or a raster file which has an associated attribute table.

A thematic layer is used to include external data into projects to use the contained information for image analysis. In geographic information systems (GIS), thematic layers contain data like land cover, road networks or slope classes.

You can use thematic layers to store and exchange intermediate image analysis results.

Thematic layers include one type of objects only: polygons, lines, or points.

→ Vector File on page 490
→ GIS on page 480

Thematic Object

A basic element of a thematic layer. A thematic object can be a polygon, a line or a point. It represents positional data of a single object in the form of coordinates and describes the object by attributes.

→ Thematic Layer on page 490

TTA Mask

Training and test area mask.

A TTA mask can be used to create sample objects (training areas in a pixel-based software approach) for supervised classification using the nearest neighbor algorithm. It can also be used to define test areas for an accuracy assessment of the classification results.

V

Vector File

A vector file represents positional data in the form of coordinates of points, lines, or polygons. A vector file can be imported into **Definiens** software or exported as a shapefile.

Shapefile on page 489

W

Workspace

A container for projects. A workspace file contains image data references, projects, exported result values and the used solution. Further it encloses the import and export templates, result states, and metadata.

A workspace is saved as a **.dpj** file together with a workspace information file (file extension **.dpm**).

16 Index

I

K

L

M

N

O

P