# eCognition Developer

## Tutorial 3 - working with maps

www.trimble.com

TRANSFORMING THE WAY THE WORLD WORKS

Trimble.

# Introduction

## About this Tutorial

This tutorial gives you an introduction to the use of so called Maps within **eCognition Developer**. The functionality is explained based on an example of change detection.

The goal of this tutorial is to give you an introduction to this feature, the different algorithms you need to make use of it.

This Module has two lessons:

- Lesson 1 Introduction to maps
- Lesson 2 Using maps: Example change detection

Further information about eCognition products is available on our website:

www.eCognition.com

## Requirements

To perform this Guided Tour, you will need:

- **eCognition Developer** installed on a computer
- A computer **mouse** is highly recommended

All steps of this tutorial can be done using the **eCognition Developer** or the free-trial version.

This tutorial is designed for self-study.

## Data included with the Tutorial

### Image data

We will be working with four QuickBird satellite image (*.tif) files in this tutorial:

- '02MAR02_multi_Maps_ChangeDetection.TIF' contains the **RGB** and **NIR** data for **T1**
- '02MAR02_pan_Maps_ChangeDetection.TIF' contains the **panchromatic** data for **T1**
- '04MAR17_multi_Maps_ChangeDetection.TIF' contains the **RGB** and **NIR** data for **T2**
- '04MAR17_pan_Maps_ChangeDetection.TIF' contains the **panchromatic** data for **T2**

### Rule Sets

A Rule Sets is available representing the final state of Rule Set development. Whenever the tutorial refers to a Rule Set, it can be found in the tutorial folder.

### Project

An eCognition Project is provided for this tutorial and can be found in the tutorial folder.

# Lesson 1 – Introduction to maps

## 1.0 Lesson contents

- About maps
- Application fields for maps

In **eCognition Developer**, you have the possibility to work with so called 'maps'. A map is a “sub-Project” where you can process independently.

- Within one Project you can have several maps.
- Maps are independent "sub-Projects".
- The original scene is always the ‘main’ map, all other, created maps can have individual names.
- You can define Image Layers and the resolution for a new map.

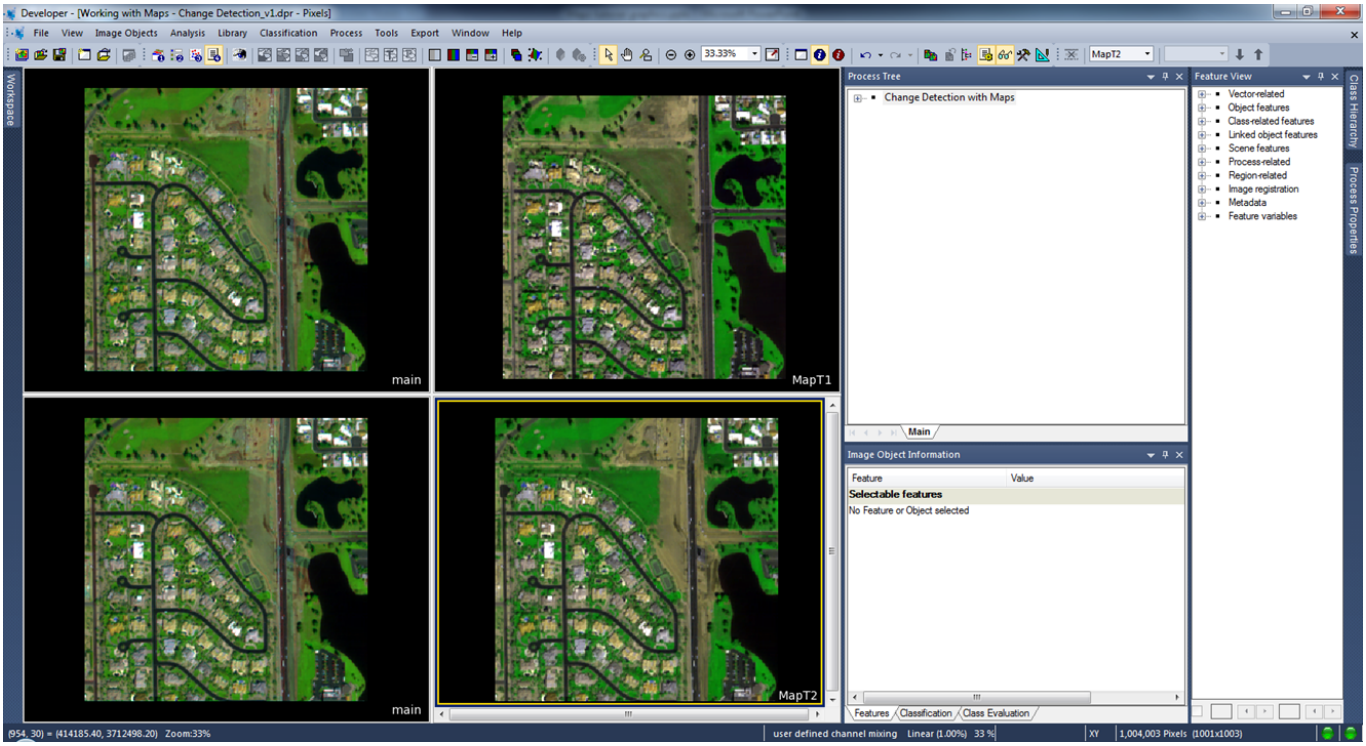



*Figure 1: One project with 4 different, independent maps.*

## 1.1 About maps

Maps can be created in several ways with eCognition:

- Directly when creating the Project (Create Project dialog box or via Customized Import).
- Through the Rule Set using the algorithm 'Copy map'.

In addition, working with maps provides a number of interesting and useful processing options:

- Process on individual maps only - the image object domains of maps act independently from one another, i.e. what you do in what map does not impact another map.
- Content can be exchanged between maps using the algorithm 'synchronize maps' or ‘transfer layer’.
- Maps can be deleted if no longer needed within the Project

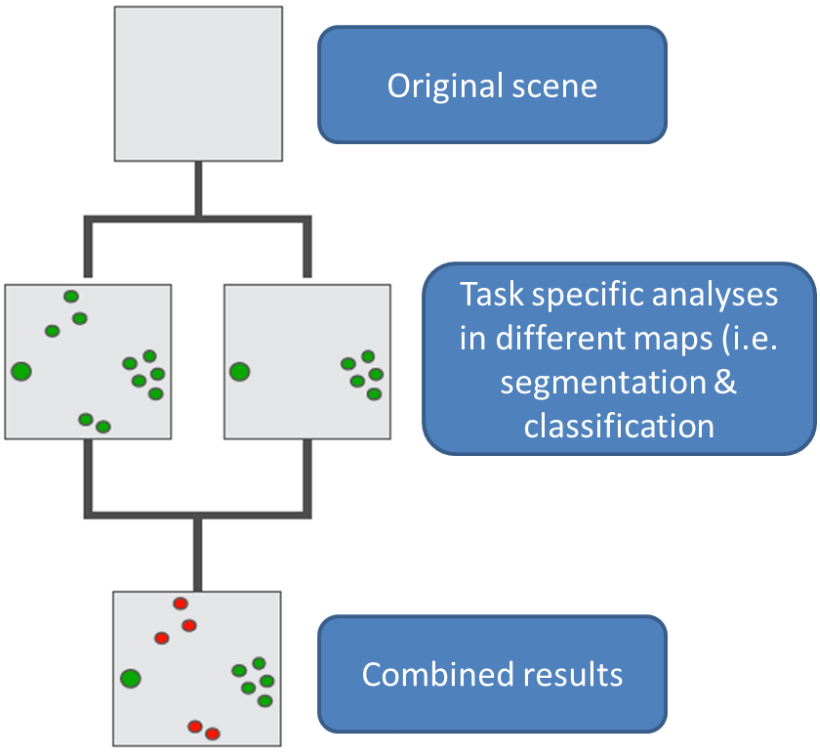



*Figure 2: Schematic figure showing the concept of maps.*

### 1.2 Application fields for maps

**Backup** of original image objects:

During Rule Set development the user can  copy a certain stage of the image analysis into a map and continue on the main map. After trying out a new segmentation or classification, for example, you can then compare the changes to the copied map and also recover the original state by synchronizing the content of the backup map into the main map again.

**Independent segmentation and classification** for different tasks:

If different Image Object Hierarchies are needed, which are independently created, you can do this in separate maps. For example, in one map you can segment using the relevant image layers for Vegetation classification, in the other map those relevant for Water classification.

**Improved performance** using down sampling approach and regions:

Another possibility in combination with maps is that you can lower the resolution of a map to be created. Coarse analysis of a wide area can be done then much faster. Or maps from defined regions can be created; an exact subset will then be the basis for the extend of the new map.

# Lesson 2 - Using maps for change detection

## 2.0 Lesson content

- The image layers of the 'main map'
- Creating two independent maps
- Classifying vegetation in both maps individually
- Synchronizing the contents of maps
- Applying the actual change detection

One application field of maps is for change detection. To set up a change detection using maps, there is a main map containing all image layers from both points of time (T1 and T2), then two independent maps are created, with only the image layers of one point of time. These are segmented and classified separately. In a last step, the actual change detection is applied to the main map, which now contains the results from both maps.

In this lesson you will go through all 4 important steps and learn:

- How to create two independent maps, one representing only T1, the other T2.
- How to classify Vegetation on both maps individually.
- How to synchronize the content of both maps back to the 'main' map.
- How to apply the actual change detection.

## 2.1 The image layers of the 'main map'

To begin this tutorial, please start by opening **eCognition Developer**.

1. In the main menu 'File' choose 'Open Project' or click on the 'Open Project' button in the toolbar.
2. Open the Project '**Maps_ChangeDetection.dpr**' in the tutorial folder where the training data has been saved.
3. Once the Project is open, switch to predefined view setting number 4 '**Develop Rulesets**'

The loaded Project contains **two sets of multispectral and panchromatic image layers** from a subset of a Quickbird scene. T1 is the multispectral and panchromatic layers from March 2002 and T2 is from March 2004.

Evaluate the loaded Image Layers. In the loaded Project, the multispectral layers of T1 are displayed. Also check the layers of T2.

*NOTE:*
*In the lower right corner of the viewer, you see which map is currently displayed. In our example right now, it is the 'main' map.*

4. Click the '**Edit Image Layer Mixing**' button in the 'View' toolbar or go to main menu View > Image Layer Mixing.

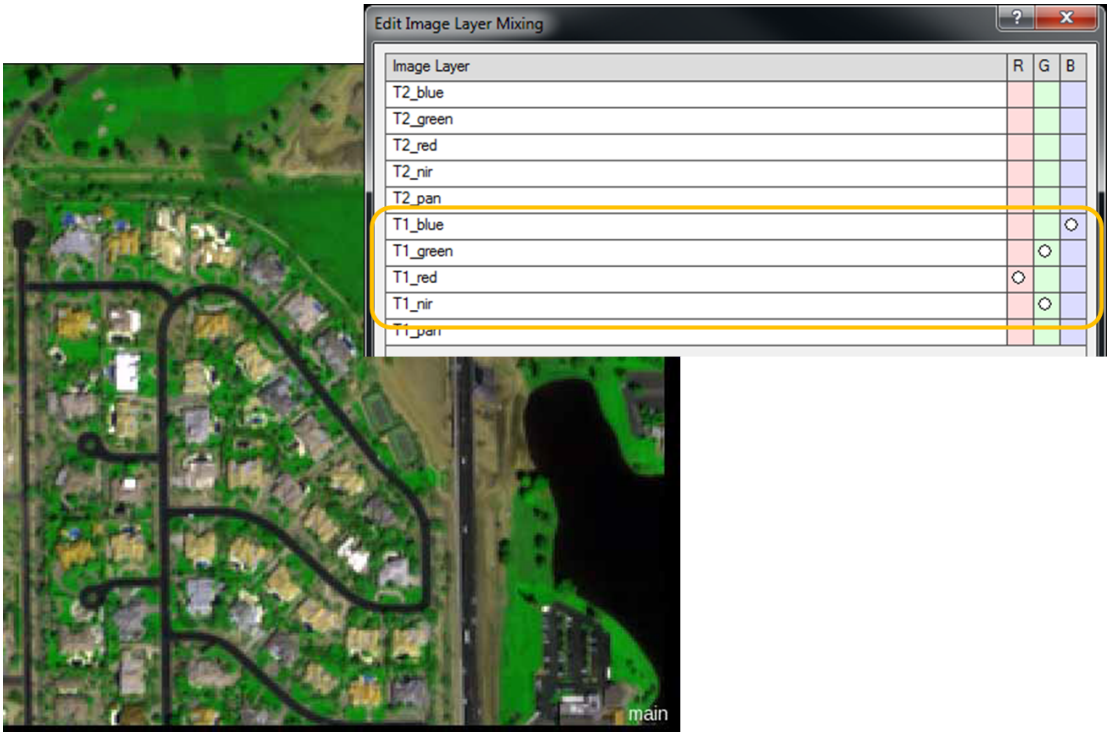



*Figure 3: The Image Layers of T1 are displayed.*

5. Click on the **up arrow** in the lower right of the 'Edit Image Layer' dialog box, until the bullets are moved **completely to the T2** multispectral layers.

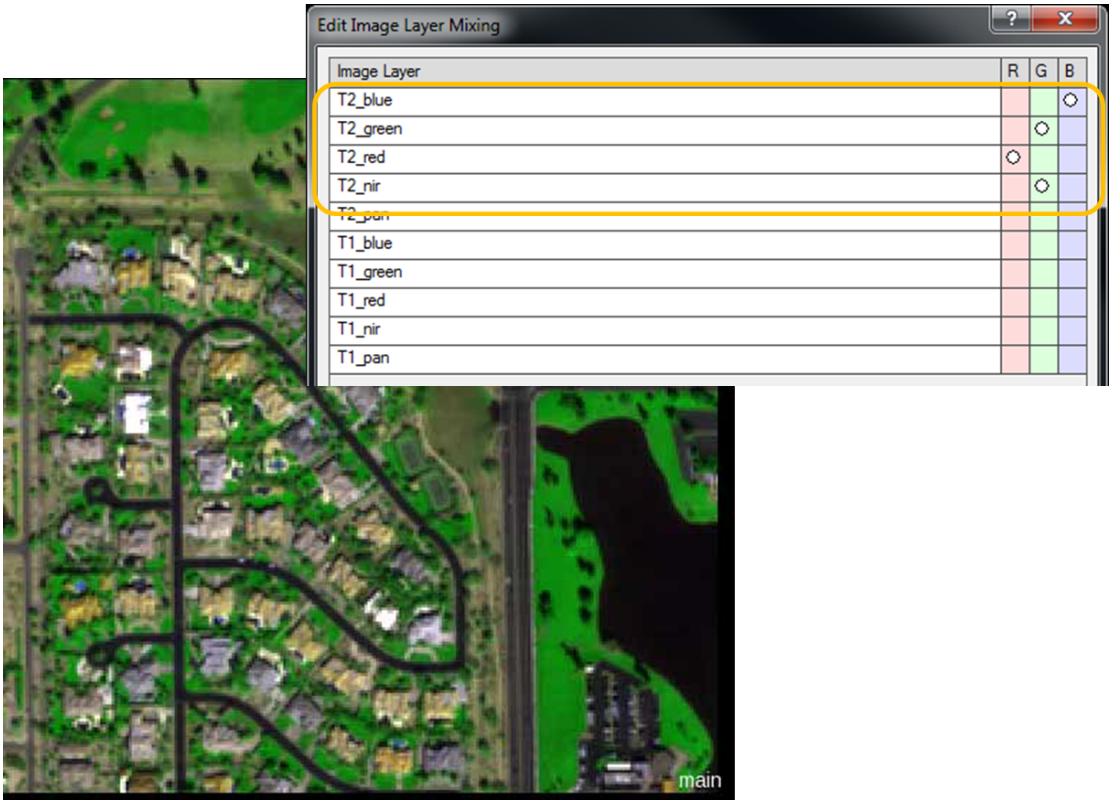



*Figure 4: The Image Layers of T2 are displayed.*

## 2.2 Creating two independent maps

In the process section '01_Create Maps' the two processes to create the necessary maps for the classification of Vegetation in both points of time are stored.

### 2.2.1 Introduction to the algorithm 'copy map'

The most important parameters of this algorithms are:

- The '**Source Region**' defines whether you want to create a map from the full extent of the original scene or if a region is the basis for the new map.
- The '**Target Map Name**' is where the name of the new map is defined.
- If the map to be created should have a different resolution, this is defined in the field '**Scale**'.
- In the field '**Image Layers**' the Image layers needed for the new map are defined. If nothing is set, all Image layers of the source map are copied to the new map.
- In the field '**Thematic Layers**' the thematic layers for the new map are defined.
- If '**Yes**' is set in the field '**Copy Image Object Hierarchy**', the existing Image Object Levels are copied to the new map. If you want to have a backup map, you would use this option.

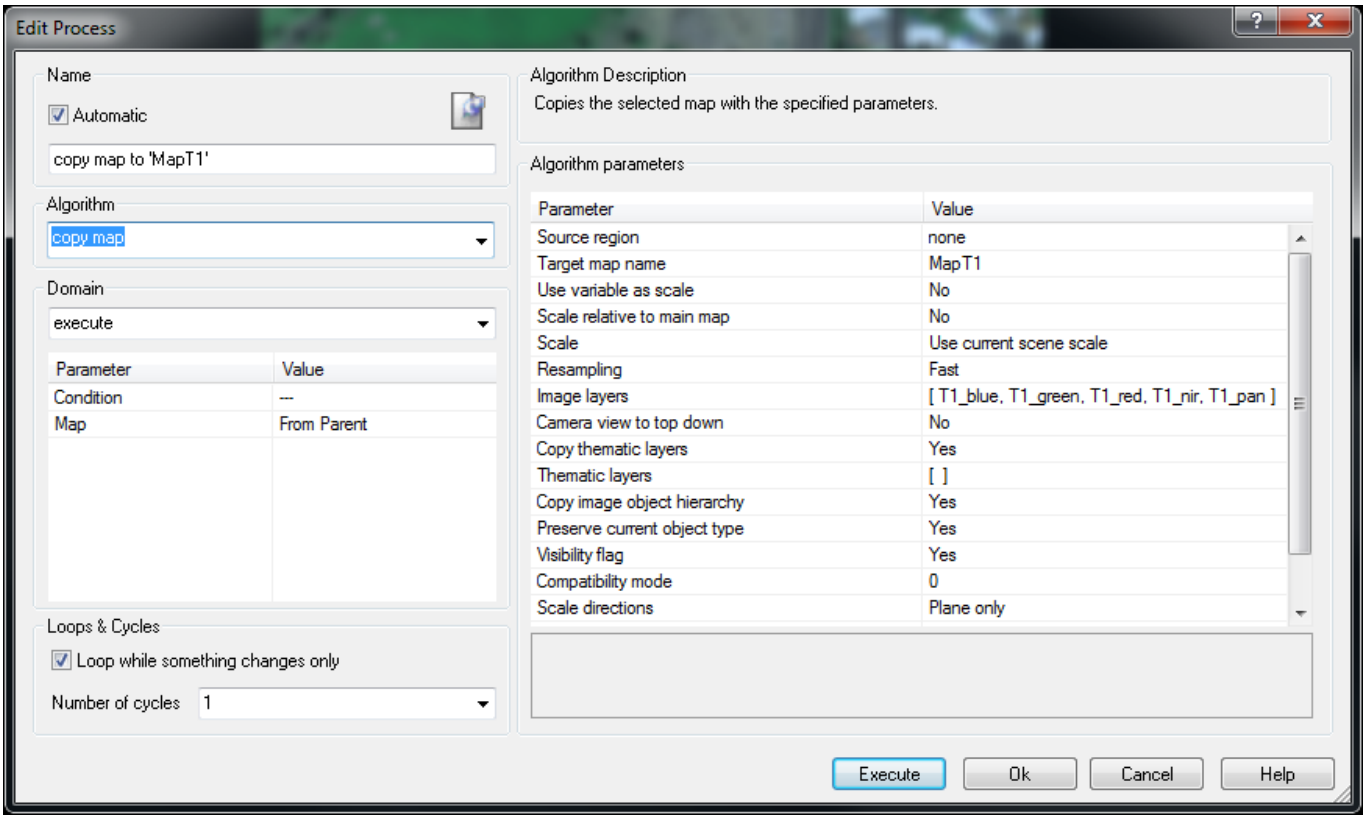



*Figure 5: Process settings of algorithm 'copy map'.*

### 2.2.2 The process settings to create the map for T1

1. Expand the process section '**01_Create Maps**'.

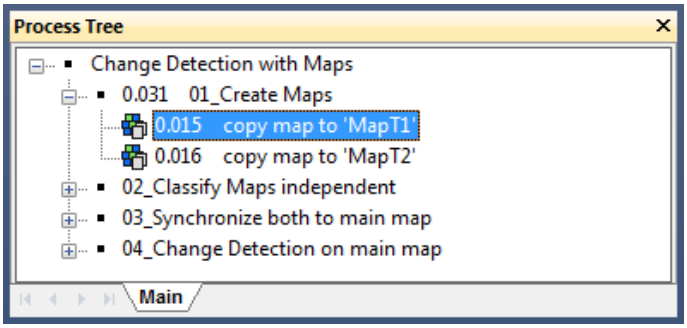



*Figure 6: Process Tree with process to create the MapT1 highlighted.*

2. Double-click on the first child process '**copy map to 'MapT1'**' to open it.
   - In the **Domain** the default settings are kept. No threshold must be set, no specific map, as there is only one existing.
   - **Source Region**: '**none**' is kept, because no region exists, which could be the basis for the map. The full extent of the loaded subset shall be copied to the new map
   - In the field '**Target Map Name**', the name '**MapT1**' is entered. To insert a name for the new map, simply type it in.
   - In the field '**Scale**', the default setting '**Use current scene scale**' is kept, as no change in resolution for the change detection analysis is needed.
   - In the field '**Image Layers**' only the layers from **T1** are chosen. The new map will then contain only these layers.

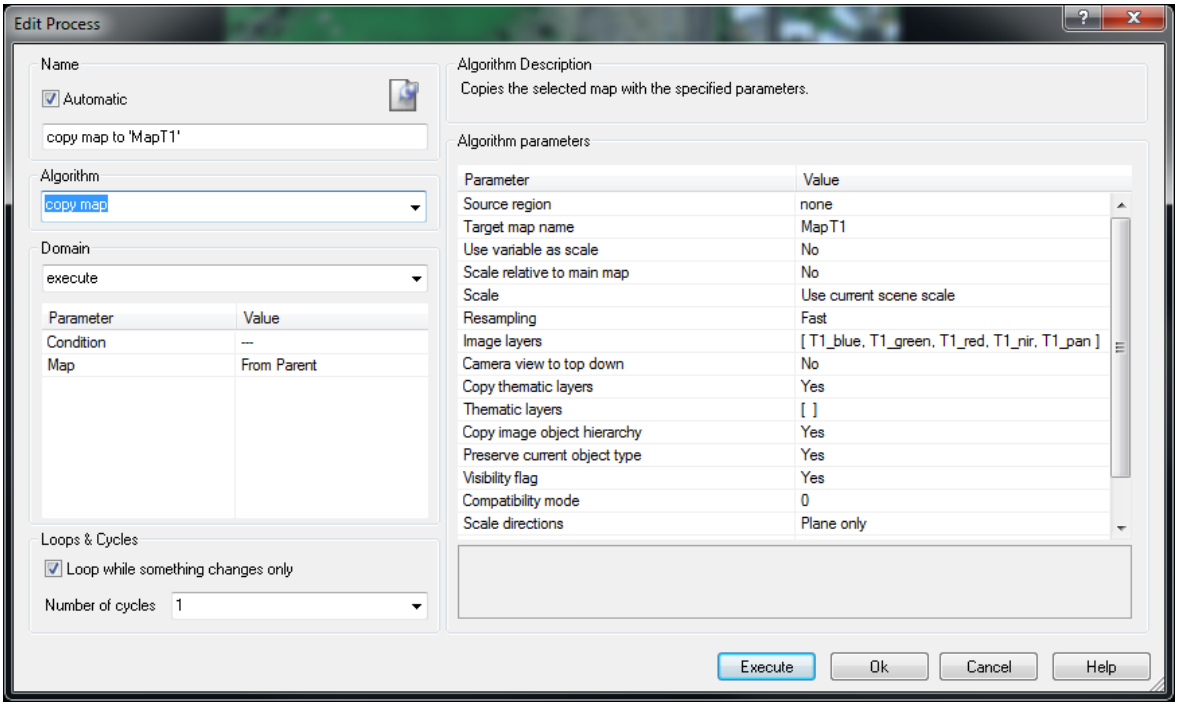



*Figure 7: Process settings to create the map 'MapT1'.*

3. Click on the '**...**' next to the 'Image Layers' field. The '**Select Image Layers**' dialog box opens.

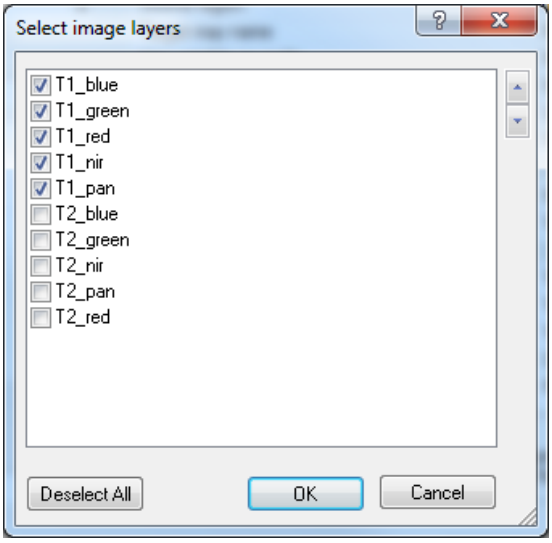



*Figure 8: Only the T1 Layers are selected.*

4. Click on the '**Cancel**' button to close the window.

### 2.2.3 Explore the new map

1. Execute the process to create 'MapT1'.
2. To display a map, use the **drop-down** list in the 'View Navigate' toolbar, right beside the 'Delete Level' button. Select '**MapT1**'.

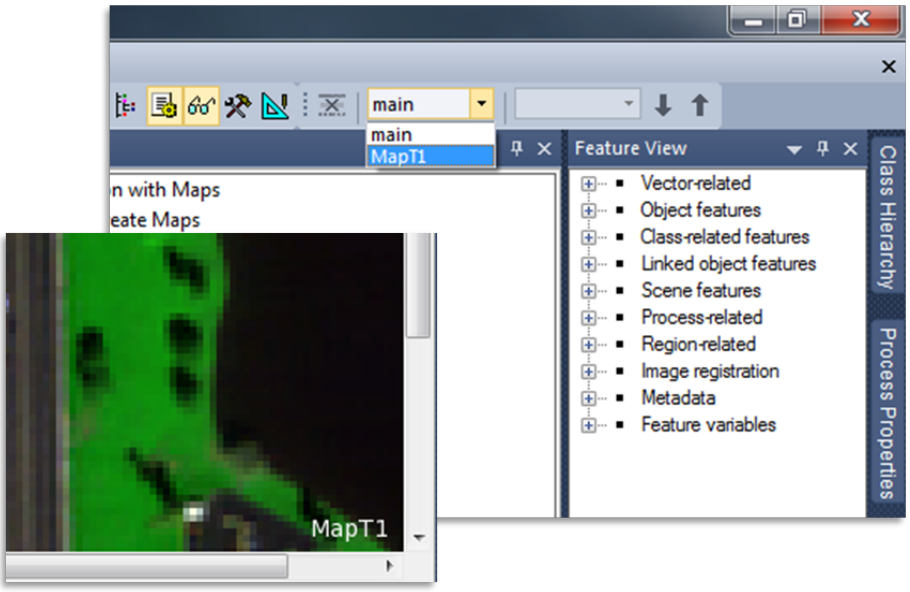



*Figure 9: The name of the displayed map appears in the lower right corner of the viewer window, now 'MapT1' is displayed.*

3. Check the 'Edit Image Layer Mixing' dialog, only image layers corresponding to T1 should be available in the new map.

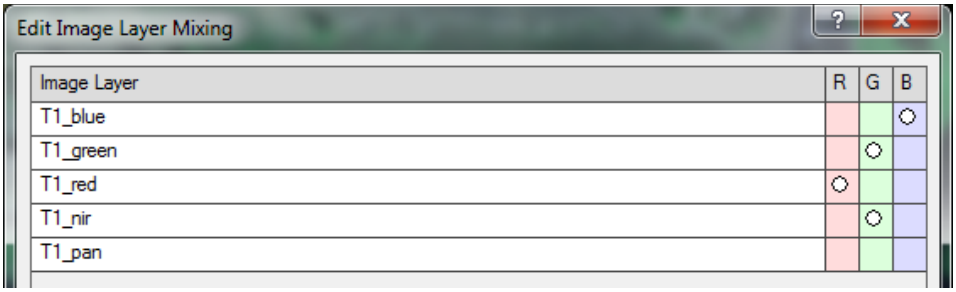



*Figure 10: The new map 'MapT1' contains only the layers of T1, as specified in the process 'copy map'.*

### 2.2.4 The process settings to create a map for T2

1. Double-click on the second child process '**copy map to 'MapT2'**' to view the settings.

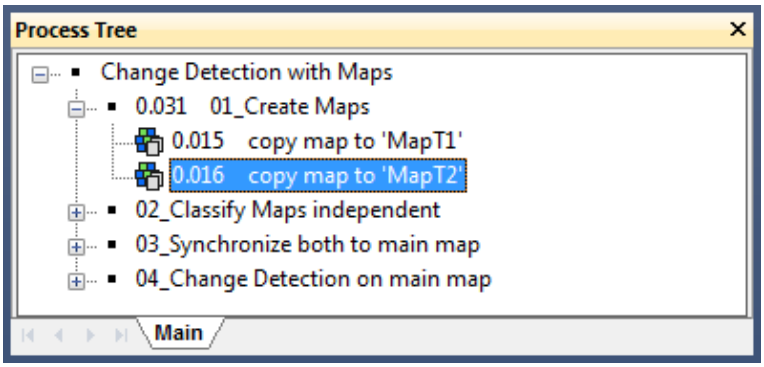



*Figure 11: Process Tree with process to create the MapT2.*

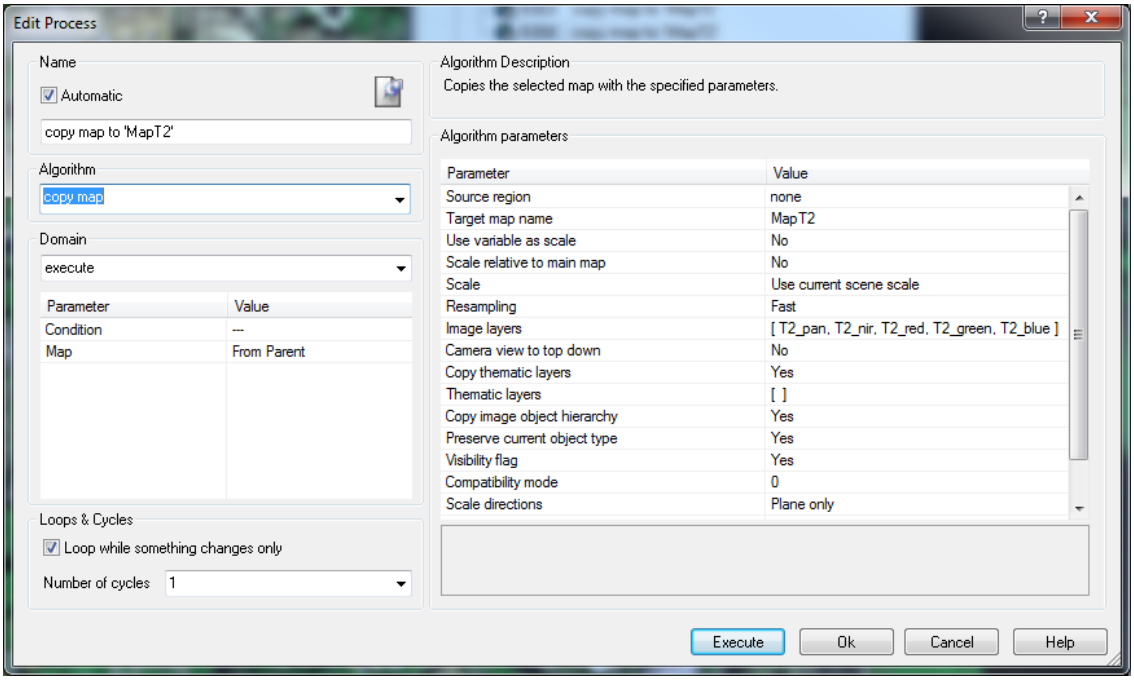



*Figure 12: Process settings to create the MapT2.*

2. Examine the algorithm parameters:
   - As the process before, the default settings are kept for the Domain.
   - Here of course the '**Target Map Name**' is '**MapT2'**.
   - In the field '**Image Layers**' only the **layers of T2** are chosen.

3. **Execute** the algorithm.

### 2.2.5 Evaluate both new maps

First open a second viewer, then display T1 in the one, T2 in the other viewer. Link both viewers with 'Side by Side' mode.

1. To open a **second viewer**, go to main menu **Window** and select either 'Split Horizontally' or '**Split Vertically**'.
2. Click in the **left** viewer window to make it active and select **MapT1** from the dropdown list in the 'View Navigate' Toolbar.

*NOTE:*
*The view that is currently active is marked with an orange outline.*

3. Click in the **right** viewer window to make it active and select **MapT2** from the drop-down list in the 'View Navigate' Toolbar.
4. Go again to the main menu 'Windows' and select '**Side by side View**'.
5. Zoom in the maps and compare the differences in both maps. Especially the Vegetation is quite different in the two maps.


*Figure 13: Two maps are created and the views synchronized: MapT1 (left) and MapT2 (right).*

## 2.3 Classifying vegetation in both maps independently

Now that you have two separate maps, you can segment and classify on both individually. This means you can create two **totally independent Image Object Hierarchies** within one Project, separated in two maps.

Which process is applied to which map is controlled by the algorithm **Domain**. If you need to apply several processes to a map, you can define the map in the **parent process as domain** and use the setting '**From Parent**' in the subsequent child processes.

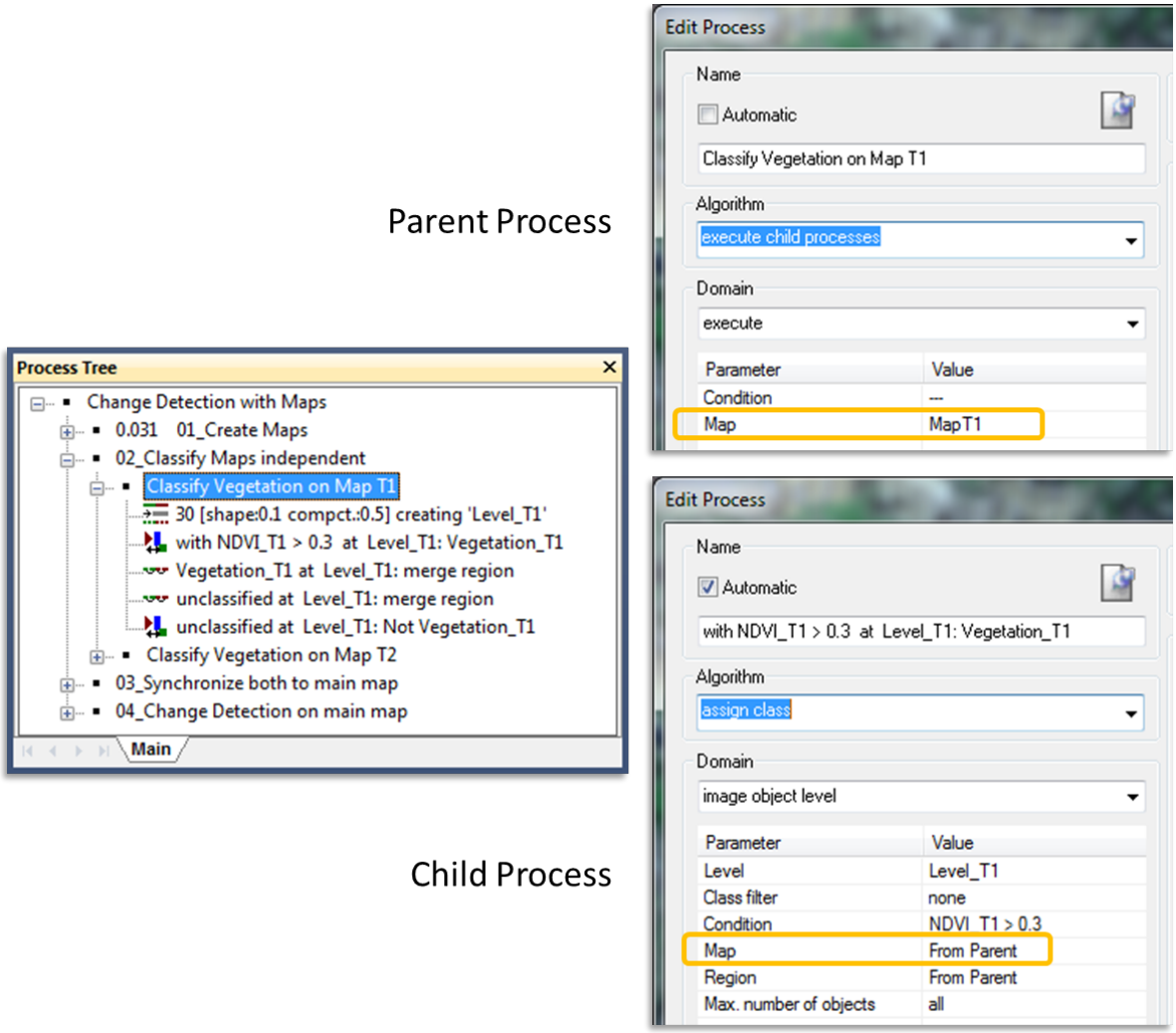



*Figure 14: In the domain of the parent process MapT1 is specified. All child processes refer to this domain.*

### 2.3.1 The process settings to classify vegetation on MapT1

1. Expand the process section '**02_Classify Maps independent**' and also '**Classify Vegetation on Map T1**'.

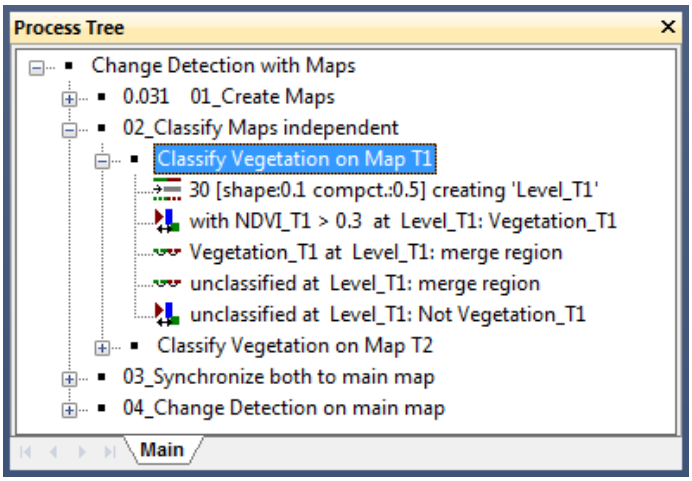



*Figure 15: Process Tree with parent process containing T1 operations.*

2. Double-click on the process '**Classify Vegetation on Map T1**' to open it.
    - As algorithm 'execute child processes' is chosen. This will be automatically set when a child process is added.
    - In the Domain of the parent process, '**MapT1**' is selected from the drop-down list.
3. Click on the 'Cancel' button to close the window.

Explore the image object domain settings of the child processes:

4. Double-click on the first child process '**30 [shape:0.1 compct.:0.5] creating 'Level_T1'**' to open it.
    - As algorithm 'multiresolution segmentation' is chosen.
    - In the Image Object Domain of the child process, 'From Parent' is selected from the drop-down menu.
5. Click on the 'Cancel' button to close the window.
6. Double-click on the second child process '**with NDVI_T1 > 0.3 at Level_T1: Vegetation_T1**' to open it.

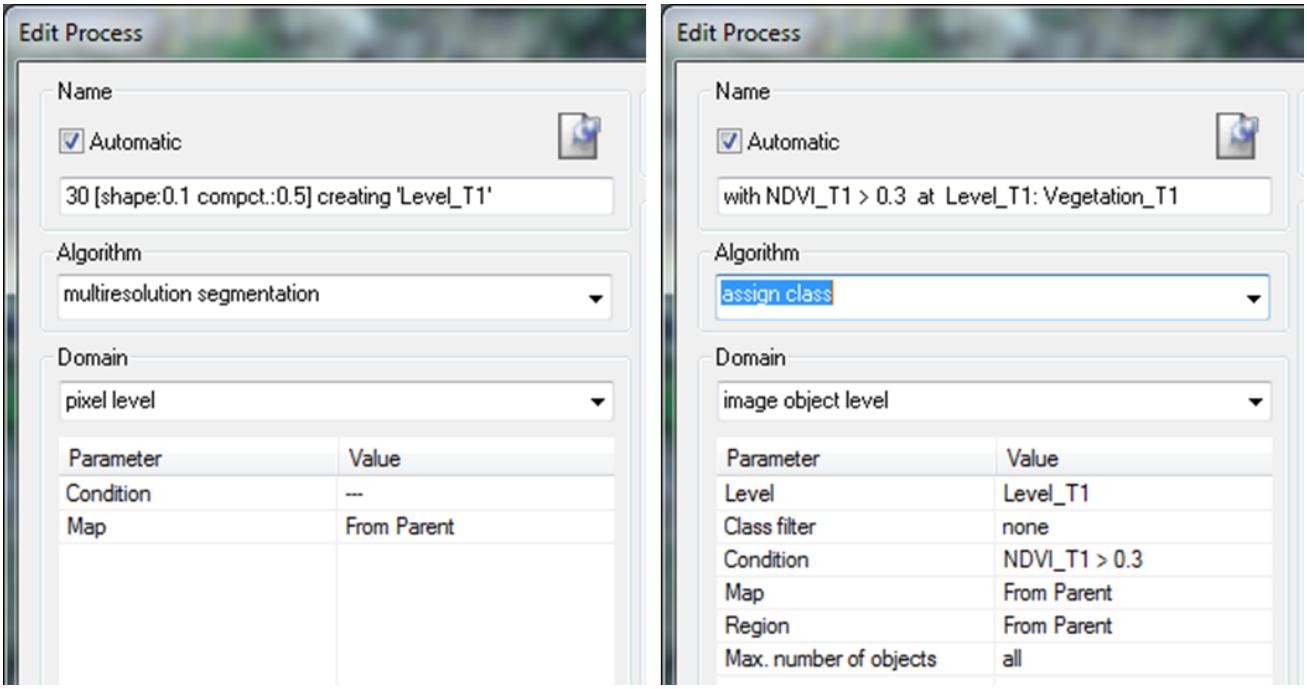



*Figure 16: Both processes point to whatever map is specified in the parent process within their Domain. Here MapT1 is defined in the parent process as domain.*

### 2.3.2 Review the classification of MapT1

1. Right-click on the process '**Classify Vegetation on Map T1**' by either right-clicking on it and select 'Execute' from the context menu or by selecting it and pressing F5 on your keyboard.
2. **Activate** the viewer with MapT1.
3. Switch on the **Classification View** with **transparency** on, additionally you can switch on the object outlines by clicking the '**Transparent/non-transparent outlined object'** button.

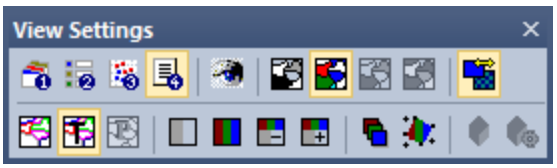



*Figure 17: The 'View Settings' toolbar with the Classification View and 'Transparent/non-transparent outlined objects' viewing options selected.*

Because ‘MapT1’ is set in the Domain of the parent process, all subsequent child processes were applied only to MapT1. Map T2 was not processed, no Image Object Level was created, no classification took place.

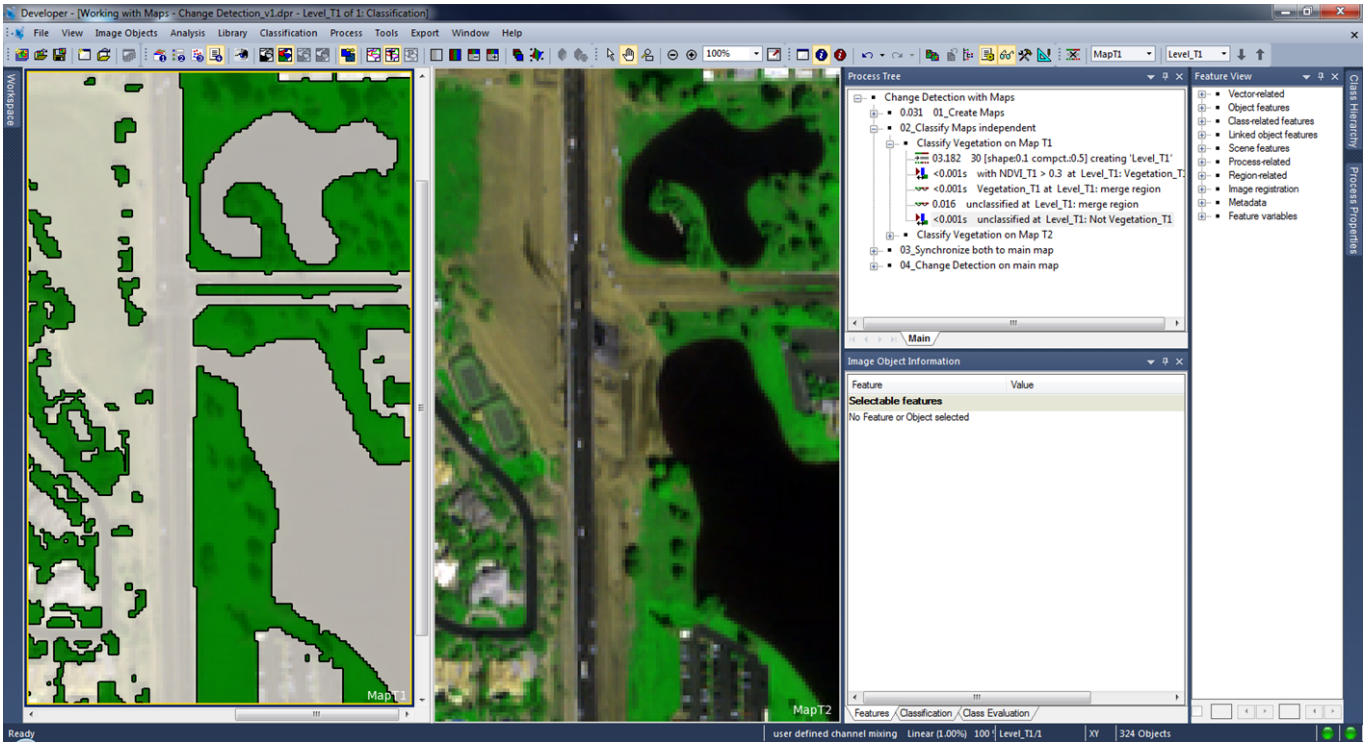



*Figure 18: MapT1 (left) is classified, MapT2 (right) is not.*

### 2.3.3 Classify vegetation in MapT2

To classify Vegetation in MapT2, the same approach is applied, in the parent process ‘Classify Vegetation on Map T2’ the **Domain is set to Map2**, all subsequent child processes are referring to this Domain.

1. Execute the process ‘Classify Vegetation on Map T2’ by either right-clicking on it and select ‘Execute’ from the context menu or by selecting it and pressing F5 on your keyboard.

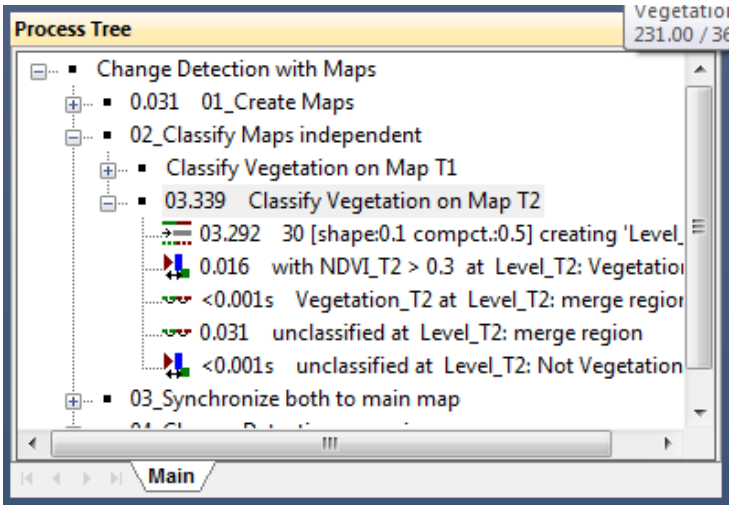



*Figure 19: Process Tree with section for classifying Vegetation on MapT2.*

### 2.3.4 Review the results

1. Activate the viewer with MapT2.
2. Switch on the **Classification View** and transparency on , additionally you can switch on the '**Transparent/non-transparent outlined object**' view.

Both Image Object Levels of both Maps were created independently and are different.

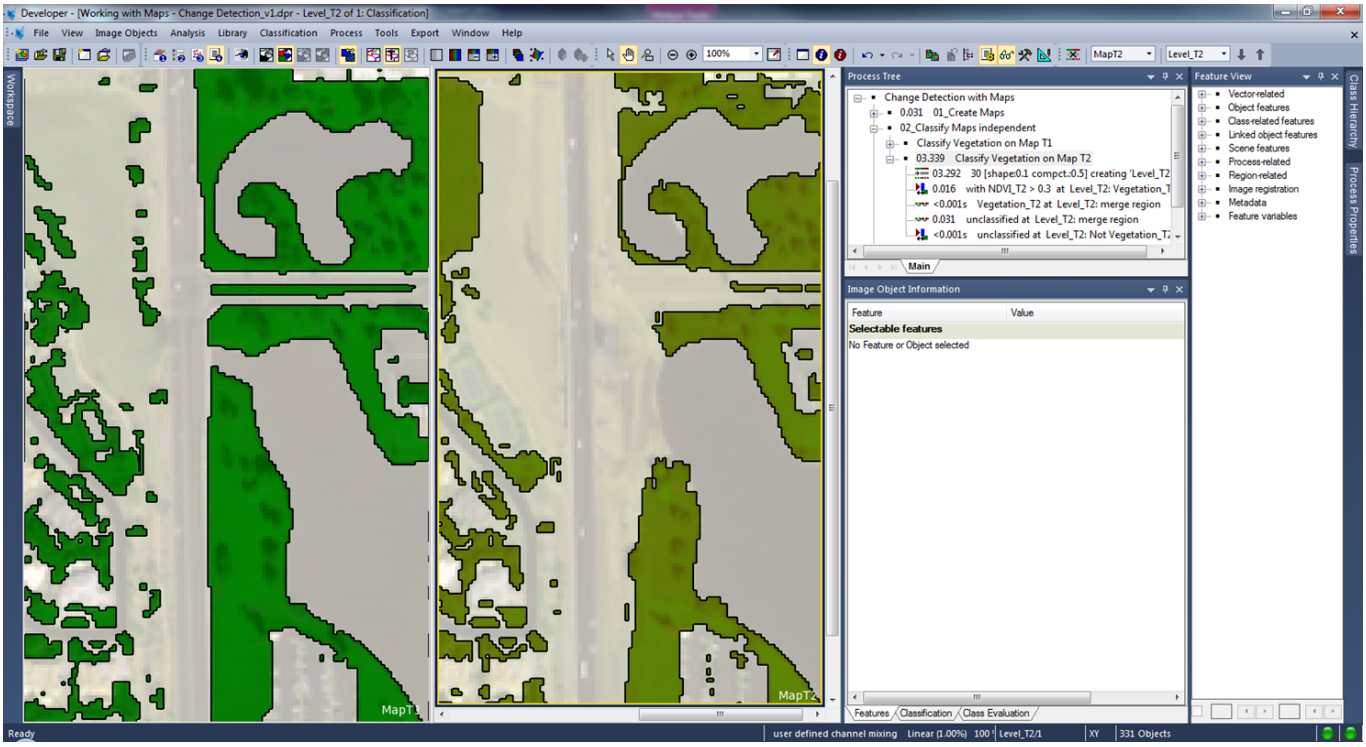



*Figure 20: In the left viewer result for MapT1 is shown, in the right viewer the result for MapT2.*

## 2.4 Synchronizing map content

After both maps have been classified, now the results of both are synchronized back to the main map. The two levels with their different classification of Vegetation then will form the basis for a change detection classification.

**To synchronize multiple levels** (LevelT1 and LevelT2) in one (main) map, the location of the image object hierarchy where the synchronization will take place must be defined.

In the current example,

- the **LevelT1 from Map T1** is copied in an empty 'main' map, no place in the hierarchy must be defined for this step.
- To avoid that the level from MapT2 overwrites the level of MapT1 in the next synchronization step, the **existing LevelT1 is copied and named LevelT2**.
- LevelT2 in the main map is then the level to which the second synchronization process points to.

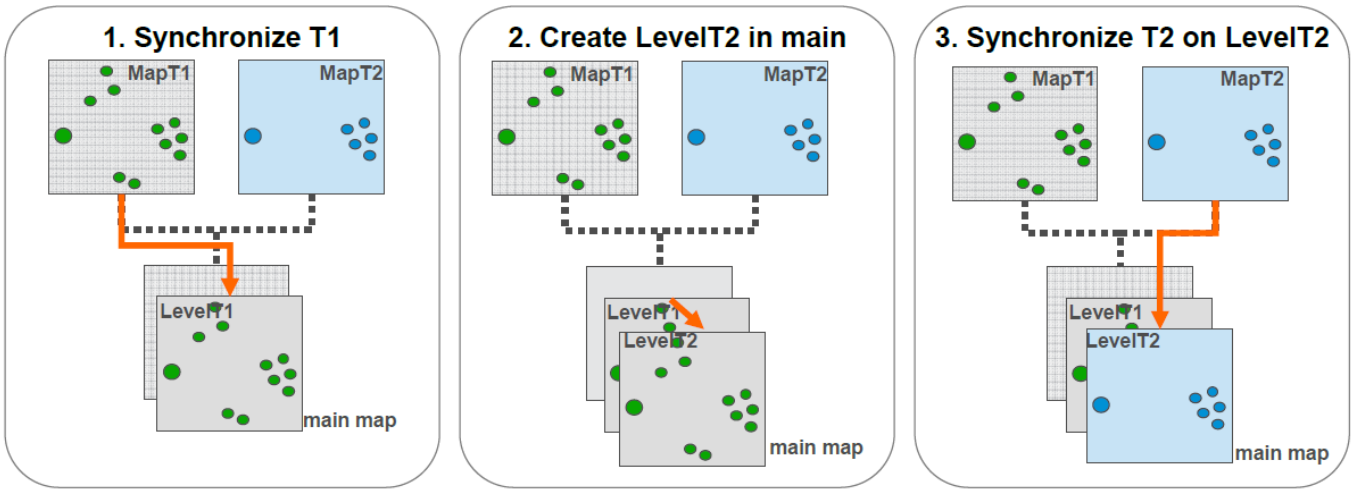



*Figure 21: Three steps to synchronize content of two maps in the main map.*

### 2.4.1 Introduction to the algorithm 'synchronize maps'

To copy the content from one map to another the algorithm 'synchronize maps' is used. This algorithm is part of the 'Maps Operations' section in the algorithm list.

With this algorithm:

- It can be defined in the Image Object Domain, from which map and level the content shall be copied.
- It can be defined in the Algorithm Parameters to which map and level the content should be added.
- It can be defined whether the synchronization is restricted to specific objects of a class.

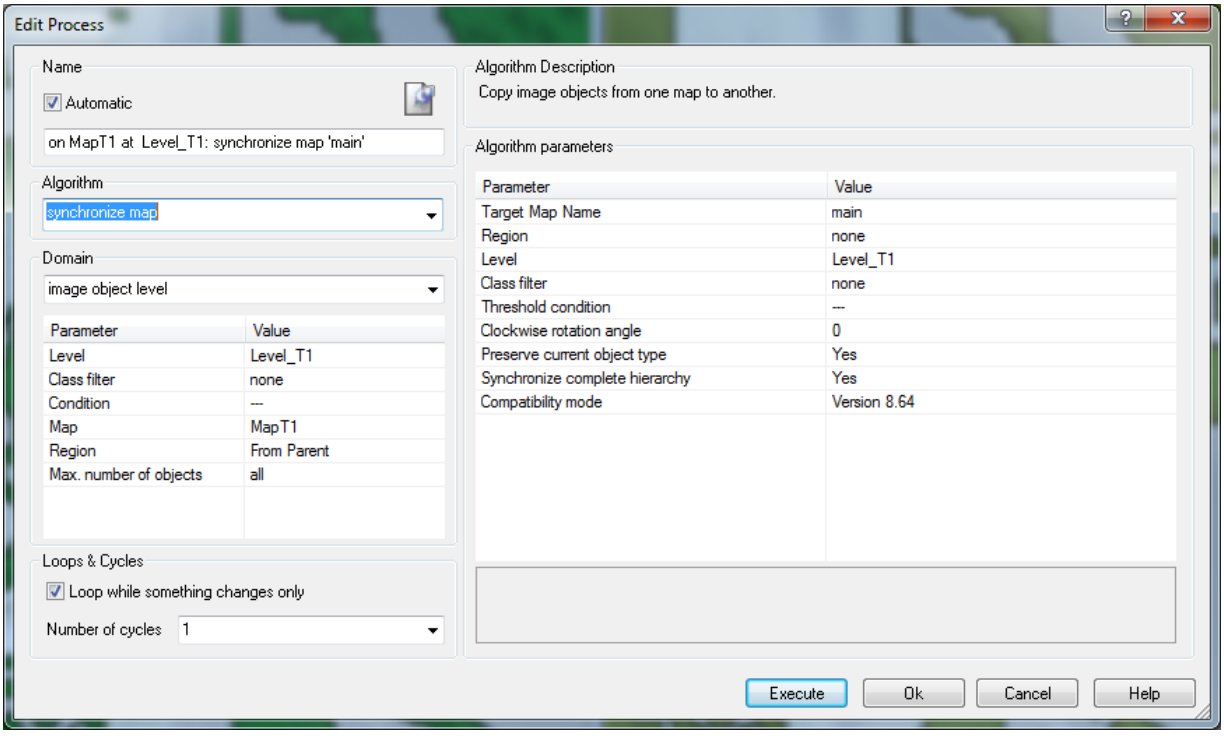



*Figure 22: Process settings of algorithm 'synchronize maps'.*

### 2.4.2 The process settings to synchronize the content from MapT1

1. Expand the process section '**03_Synchronize both to main map**'.

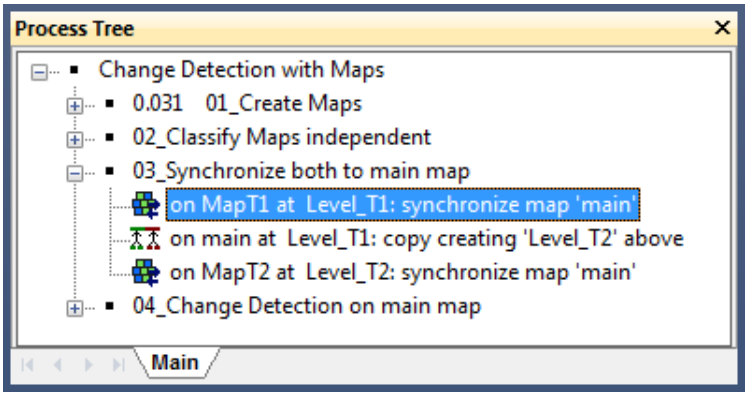



*Figure 23: Process Tree with process to synchronize content of MapT2 to main Map.*

2. Double-click on the first child process '**on MapT1 at Level_T1: synchronize map 'main''** to open it.
   - The Domain defines the source for synchronization, here the image object hierarchy from '**Level_T1**' in '**MapT1**' is chosen.
   - The Algorithm Parameters define the **target of synchronization**.
   - In the drop-down list of the field '**Target Map Name**' the **main** map is chosen.
   - No region is defined in the field 'Region'.
   - In the field 'Level' the name of the new level in main map is defined, here '**Level_T1**'. You can either type in a new name or pick one from the drop-down list.

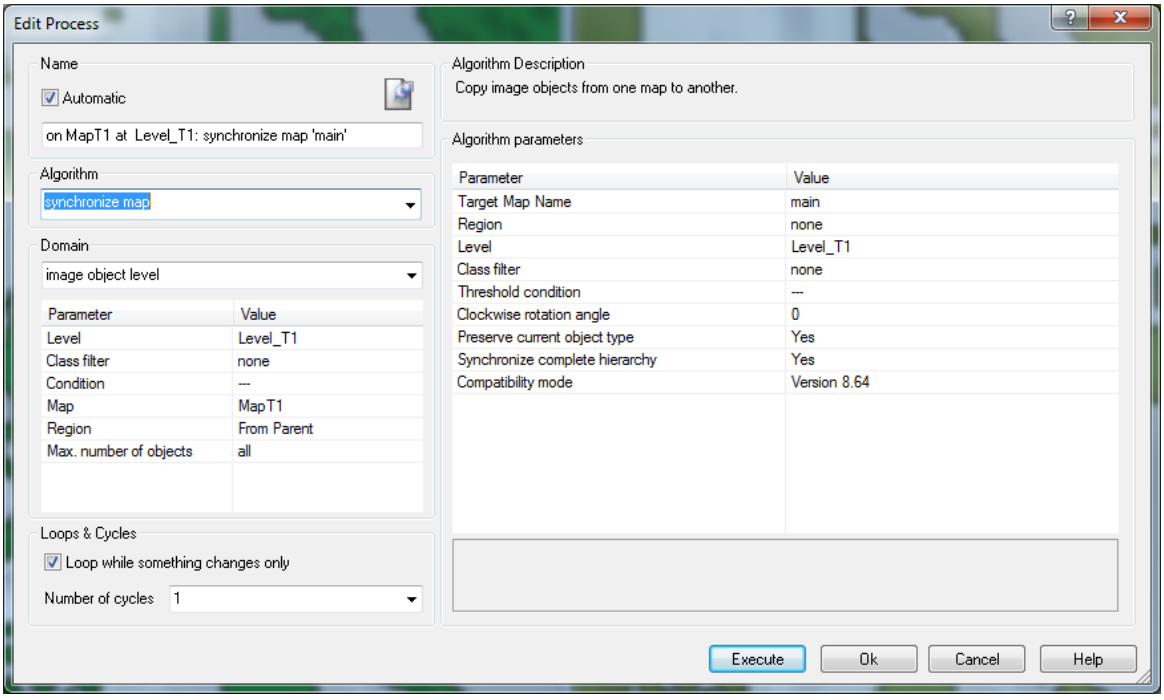



*Figure 24: Process settings to synchronize content of MapT1 to main Map.*

3. **Execute** the process '**on MapT1 at Level_T1: synchronize map 'main''**.
4. Display MapT1 in one viewer and the main map in the other.

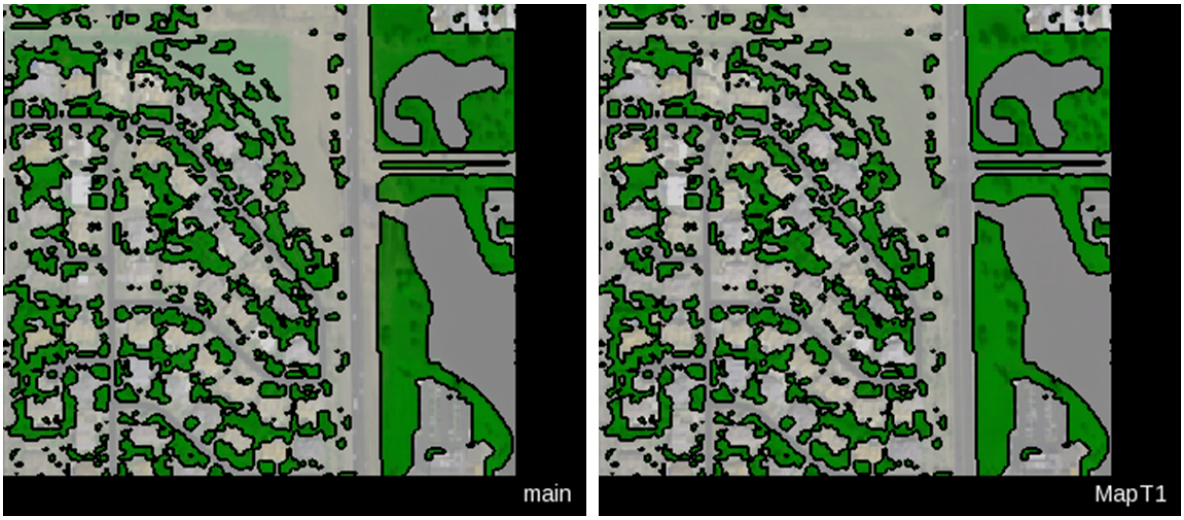



*Figure 25: The content of MapT1 is now also available in main map.*

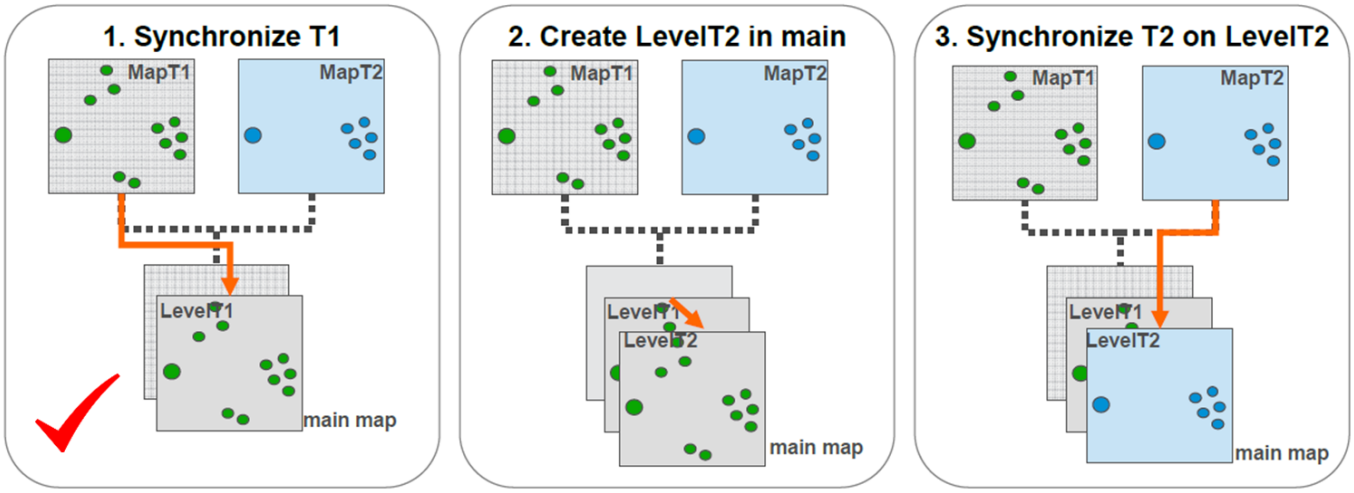



*Figure 26: The first step of the synchronization is complete. The second step of the synchronization process is to create a level in the main map.*

### 2.4.3 The process settings to copy the existing level in the main map

1. Double-click on the second child process '**on main at Level_T1: copy creating 'Level_T2' above**' to open it.
    - In the field 'Level', it is specified that '**Level_T1**' shall be copied.
    - In the field 'Map' it is defined that the process is to be executed in the **main map**, not in any of the others.

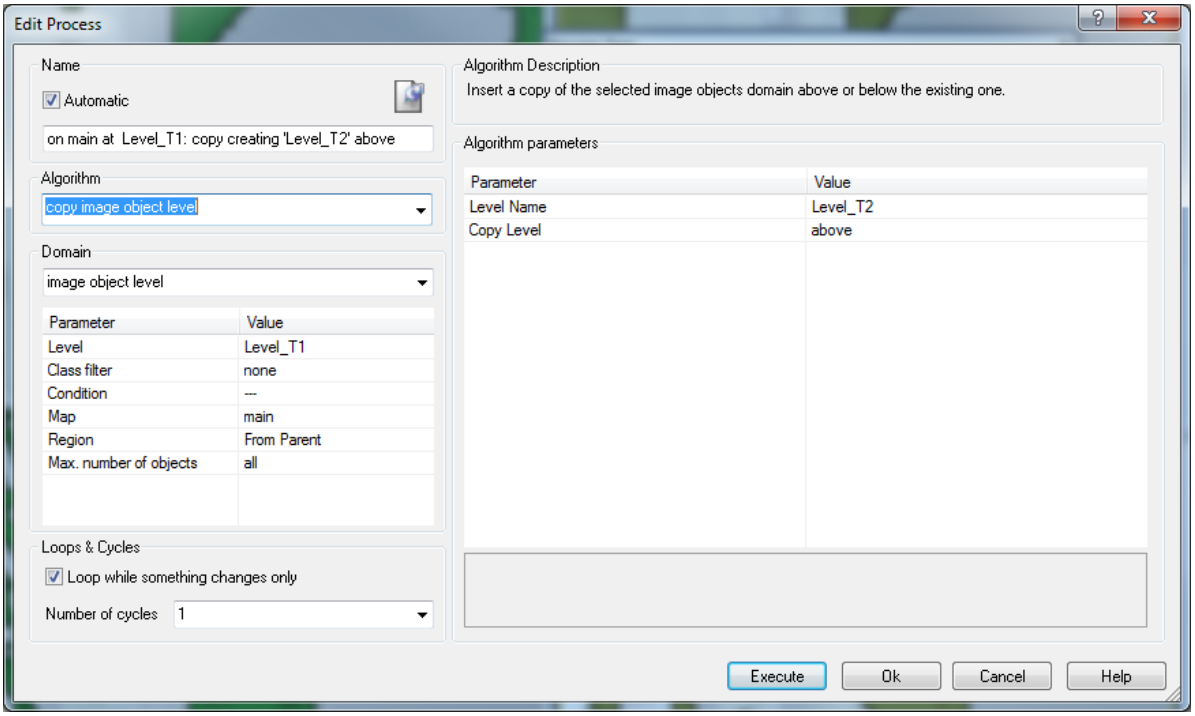



*Figure 27: Process settings to copy 'Level_T1' above. The Domain specifies that this shall only be executed only in the main map.*

2. **Execute** the process.

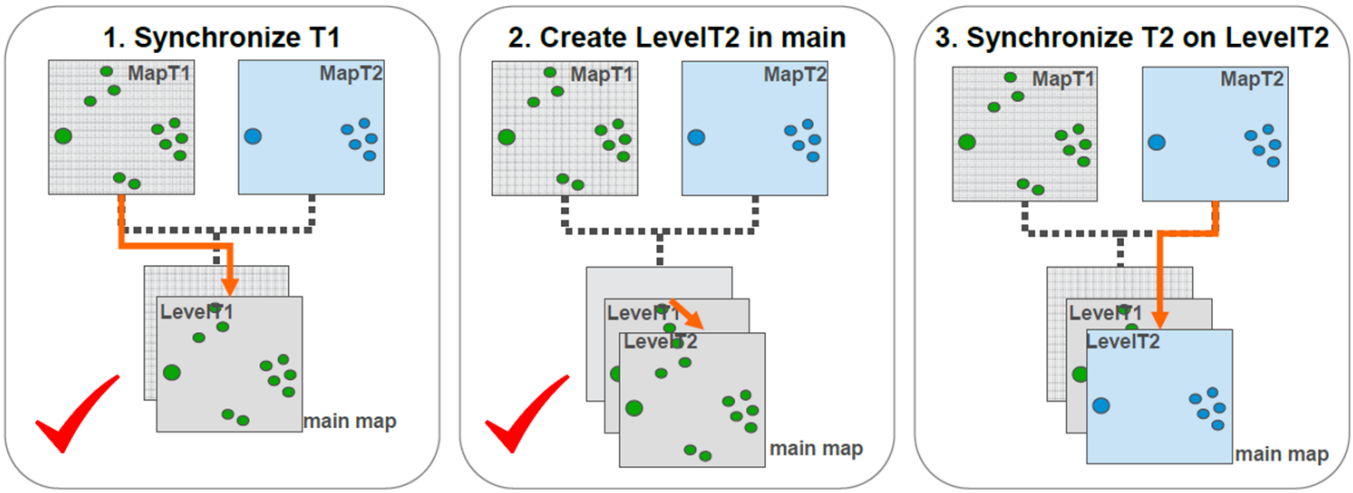



*Figure 28: The second step of the synchronization process is to create a level in the main map.*

A Level_T2, identical with Level_T1, is now created. This new Level_T2 will be the basis for the next synchronization step to bring the content of the MapT2 into the main map.

### 2.4.4 The process settings to synchronize the content of MapT2

1. Double-click on the third child process '**on MapT1 at Level_T1: synchronize map 'main''**' to open it.
    - The Domain defines the source for synchronization, here the image object hierarchy from '**Level_T2**' in '**MapT2**' is chosen.
    - The Algorithm Parameters define the **target** of synchronization.
    - In the drop-down list of the field 'Target Map Name' the main map is chosen.
    - No region is defined in the field 'Region'.

- In the field ‘Level’ the name of the new level in main map is defined, here
- ‘Level_T2’. Here the level has to be picked one from the drop-down list, as it exists already in the main map.

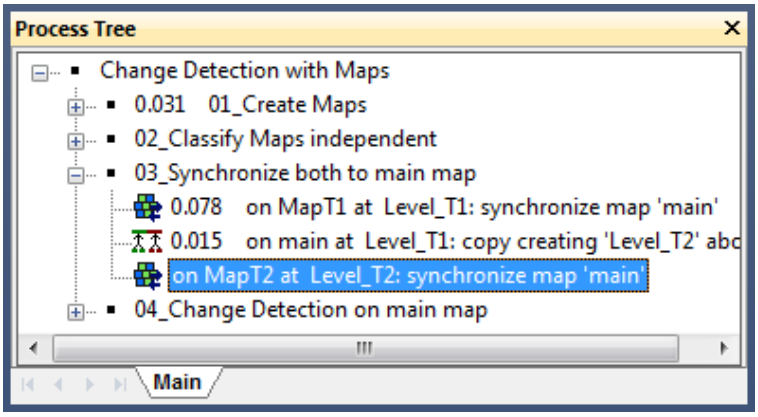



*Figure 29: Process Tree with process to synchronize content of ‘MapT2’ to main map.*

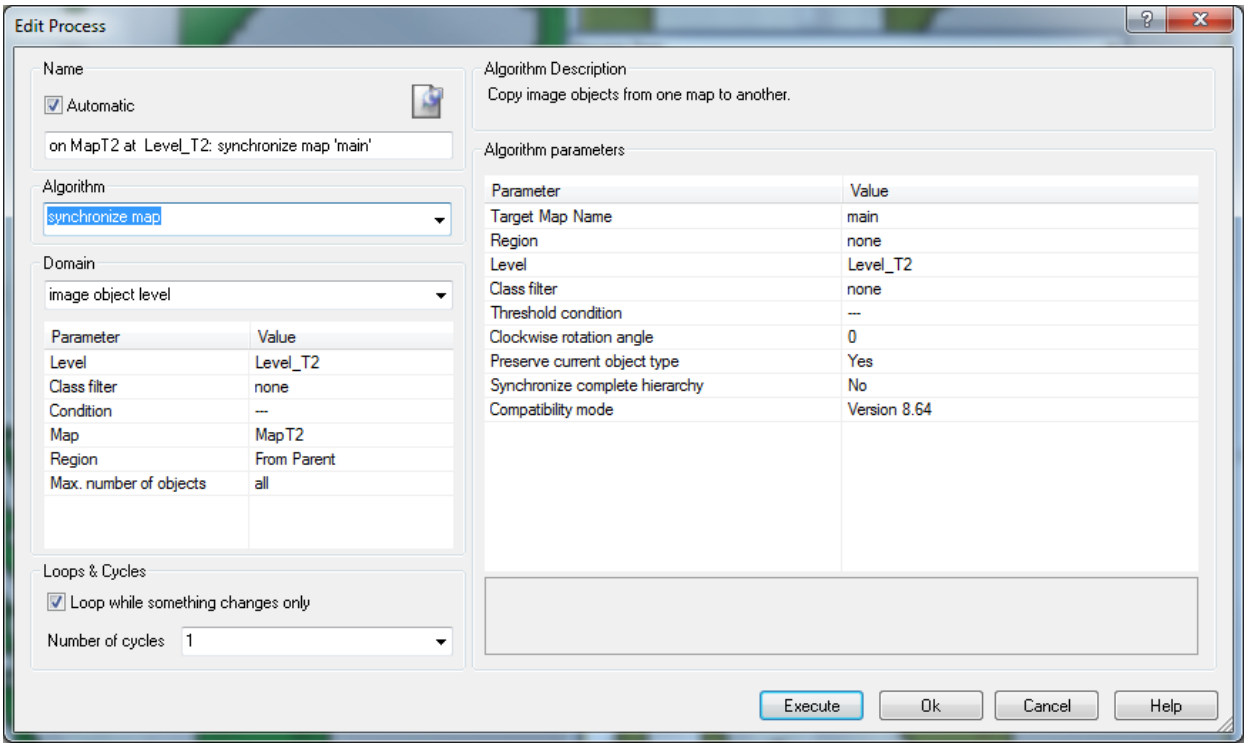



*Figure 30: Process settings to synchronize content of MapT2 to main map.*

2. Close the process by clicking on ‘Cancel’.
3. Execute the process ‘**on MapT2 at Level_T2: synchronize map 'main'**’

### 2.4.5 Review the synchronization results

1. Open 4 viewers by additionally selecting i.e. ‘Split Horizontally’ from the main menu ‘Window’.
2. Display the **main map** and **Level_T1** in the **upper left** viewer , in the **upper right main map** and **Level_T2**. In the **lower left** viewer, display the **MapT1**, in the **lower right MapT2**.

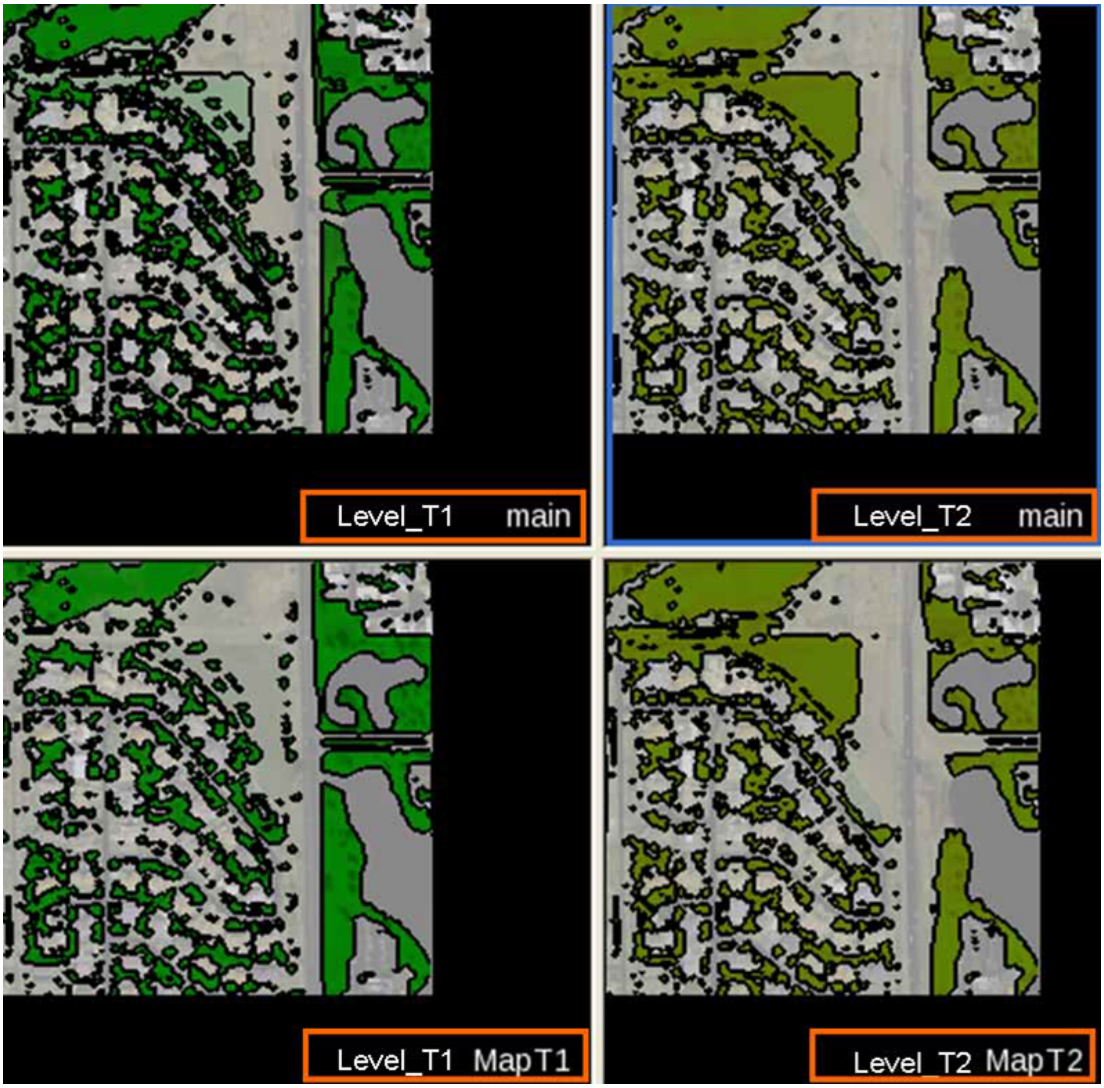



*Figure 31: Level_T1 in the main map (upper left); Level_T2 in the main map (upper right); Level_T1 in the MapT1 (lower left); Level_T1 in the MapT1 (lower right).*

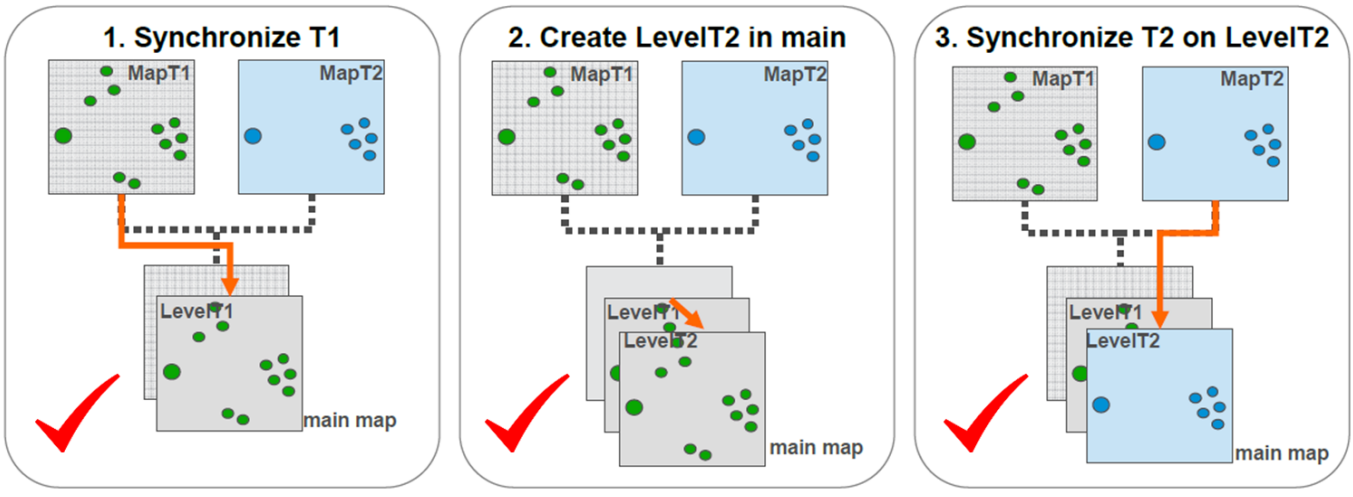



*Figure 32: All three steps to synchronize are executed.*

## 2.5 Applying the change detection

Now that the main map contains both analysis results from T1 and T2, the actual change detection can be applied. The main map is set as the domain in the parent process, similar as in the section to classify the vegetation in the individual maps.

Before the actual change detection classification is applied the Image Object Hierarchy must be prepared.

### 2.5.1 The main map as Domain in the parent process

The processes for detecting the changes are executed in the main map. Therefore, the same approach as for the analysis of the individual maps is chosen. In the parent process the main map is set as domain and all subsequent child processes point to this parent process domain.

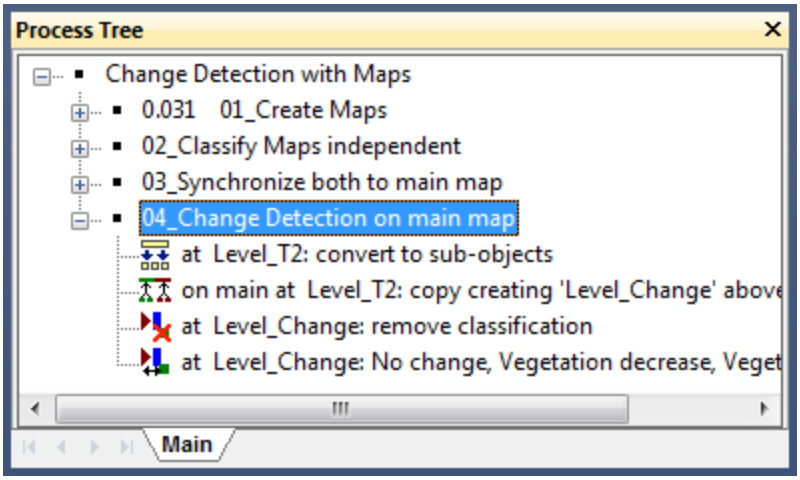



*Figure 33: The child processes get the information to process only on the main map from the parent process.*

### 2.5.2 The process settings to merge image combine outlines from Level_T1 and Level_T2

The objects of a change detection level must represent both, the **outlines of the objects of 'Level_T1' and 'Level_T2'**. Therefore, the algorithm 'convert to sub-objects' is applied to **combine the outlines** of 'Level_T1' in 'Level_T2' without any effect on the classification. Then 'Level_T2' is **copied** above, this level is named '**Level_Change**'.

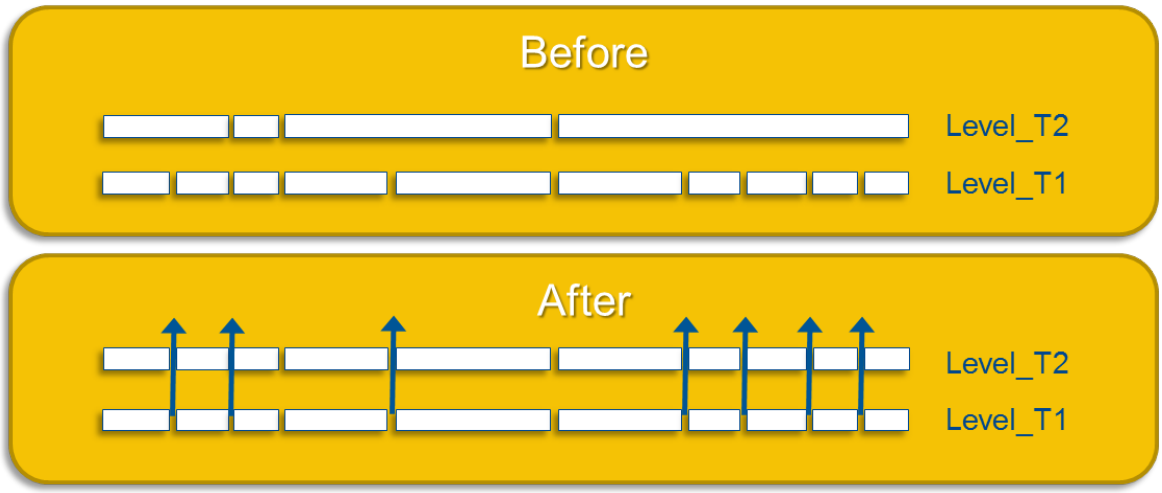



*Figure 34: First the outlines of 'Level_T2' are different from 'Level_T1', after executing the process, the outlines are cut into the upper level.*

1. Expand the process section '**04_Change Detection on main map**'.
2. Double-click on the first child process '**at Level_T2: convert to sub-objects**' to open it.
    - In the Image Object Domain of this algorithm it is defined which level is the one to be cut. Here in this case it is the upper 'Level_T2'.
      Automatically the outlines of the level below(here 'Level_T1') are cut into it.

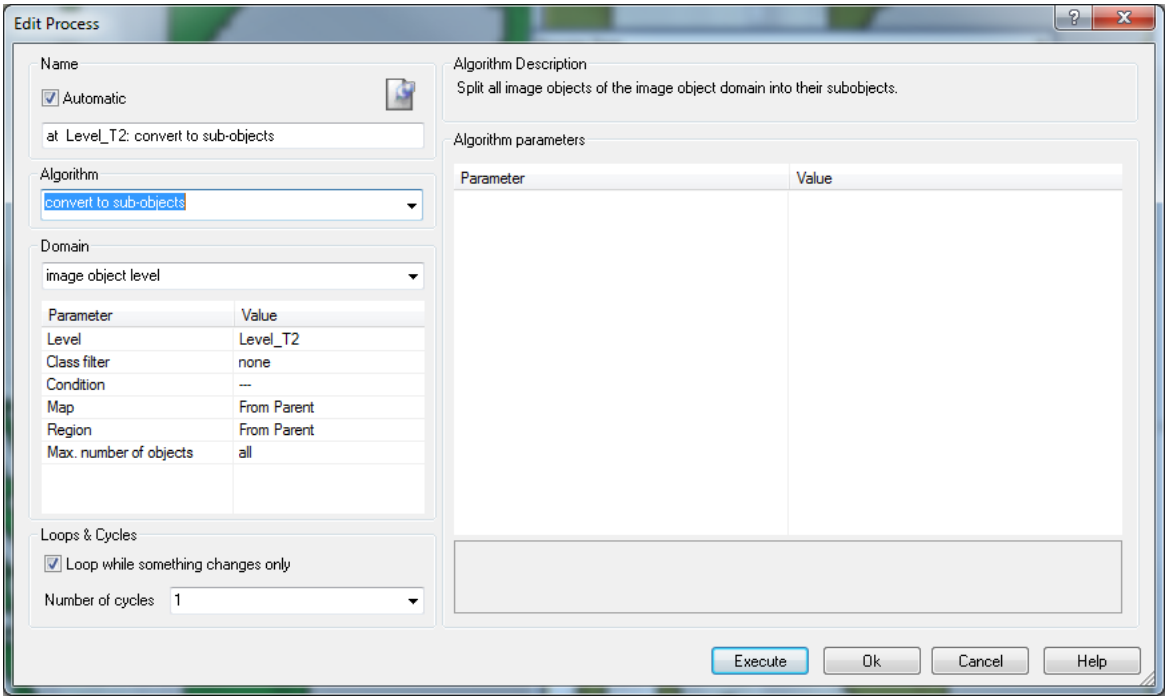



*Figure 35: Process settings to cookie-cut the outlines from 'Level_T1' in 'Level_T2'.*

3. Execute the algorithm and review the resulting objects

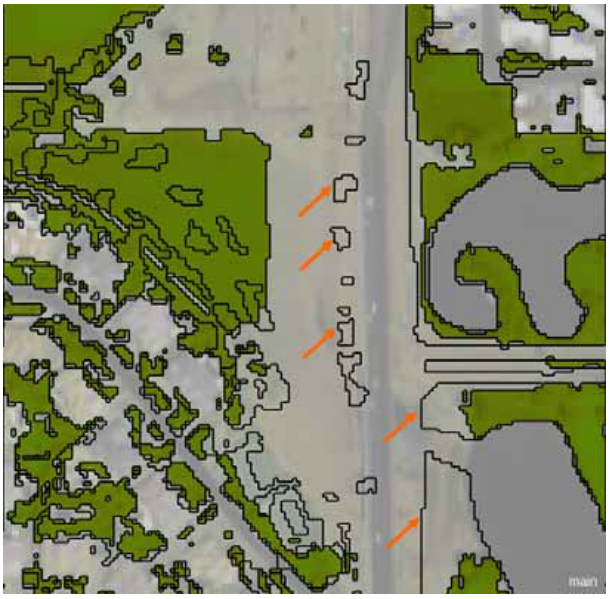


*Figure 36: The outlines of the underlying level are now also cut in the upper level.*

After executing the 'convert to sub-objects' process, the 'Level_T2' also represents the outlines of 'Level_T1'. Now the Object Hierarchy is ready to be copied in an extra level where the changes will be classified.

### 2.5.3 The change detection classification

The rules for change detection are stored in the Class Description of the classes, the Class-Related Feature '**Existence of sub-objects**' is used to describe the three different change classes.

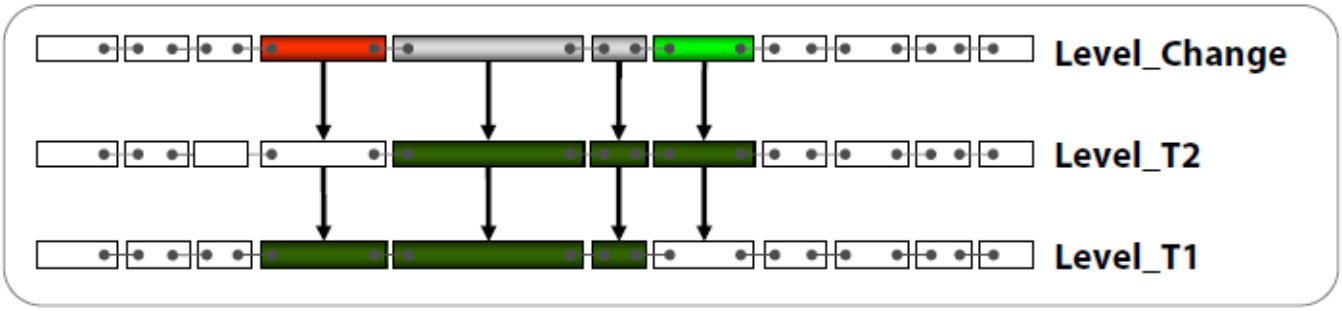
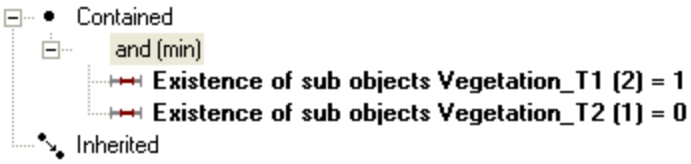
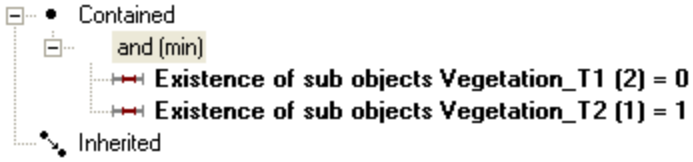



*Figure 37: For the classification of the changes, the feature 'Existence of sub-objects' is used.*

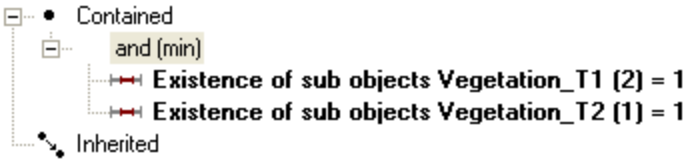
- Vegetation increase:




- Vegetation decreas:




- No change:




1. Execute the process '**on main at Level_T2: copy creating 'Level_Change' above**'.
2. Execute the process **'at 'Level_Change: remove classification**' to have an unclassified 'Level_Change'.
3. Execute the classification process '**at Level_Change: No change, Vegetation decrease, Vegetation increase**'.

### 2.5.4 Review the results

The classes Vegetation decrease, No change, Vegetation increase are classified.

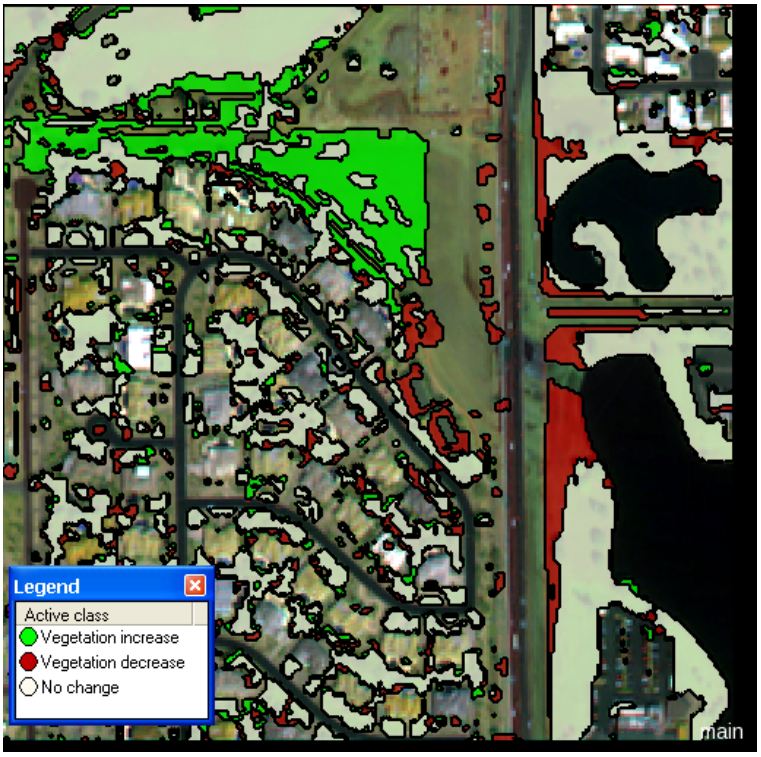



*Figure 38: Classification view of the final change detection level in the main map.*

## 2.6 Conclusion

### Two independent maps were created

Two maps were created using the algorithm 'copy map'. One map contains only Image Layers of T1, the other only image layers of T2.


*Figure 39: Two maps are create: MapT1 and MapT2.*

### Vegetation classified in both maps individually

In the domain of the parent process it is defined that the subsequent child processes are applied only to the specified map.

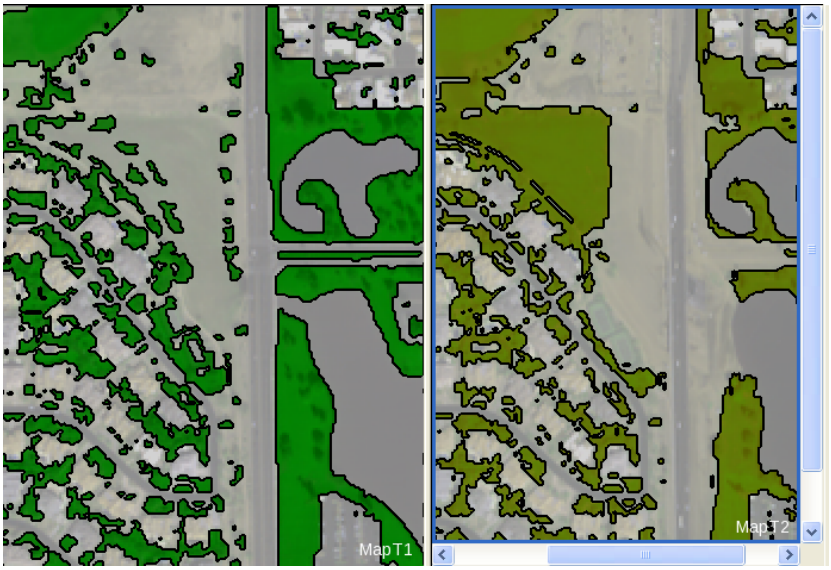



*Figure 40: Classification results for MapT1 (left window); classification results for MapT2 (right window). Both Image Object Levels are independently segmented and classified.*

**The content of both maps was synchronized**

The synchronization was applied in three steps. The content of the first map was copied to the main map using the algorithm 'synchronize maps', then a level was copied in the main map, finally the content of the second map is copied to the newly created level in the main map.

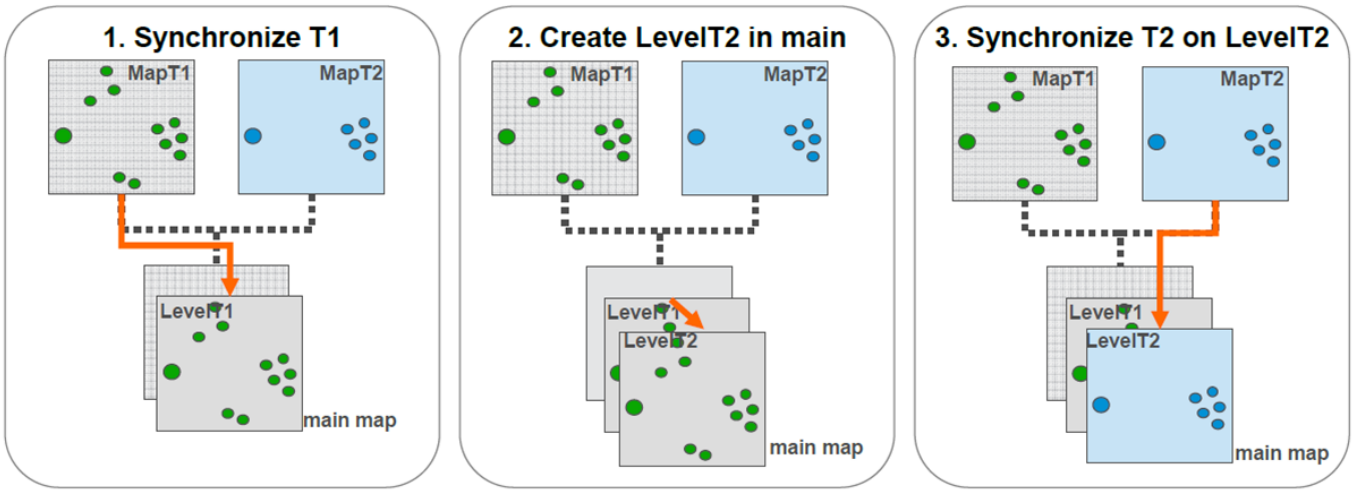



*Figure 41: Schematic workflow to synchronize maps.*

**Change detection classification applied**

The actual change detection was applied in the main map. The Object Hierarchy was equalized for all levels. The Class-Related Feature 'Existence of sub-objects' was used to identify changes.

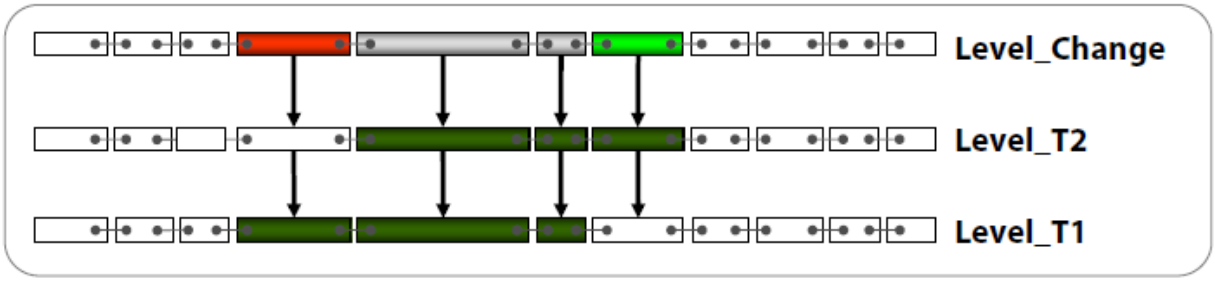



*Figure 42: For the classification of the changes, the feature 'Existence of sub-objects' is used.*

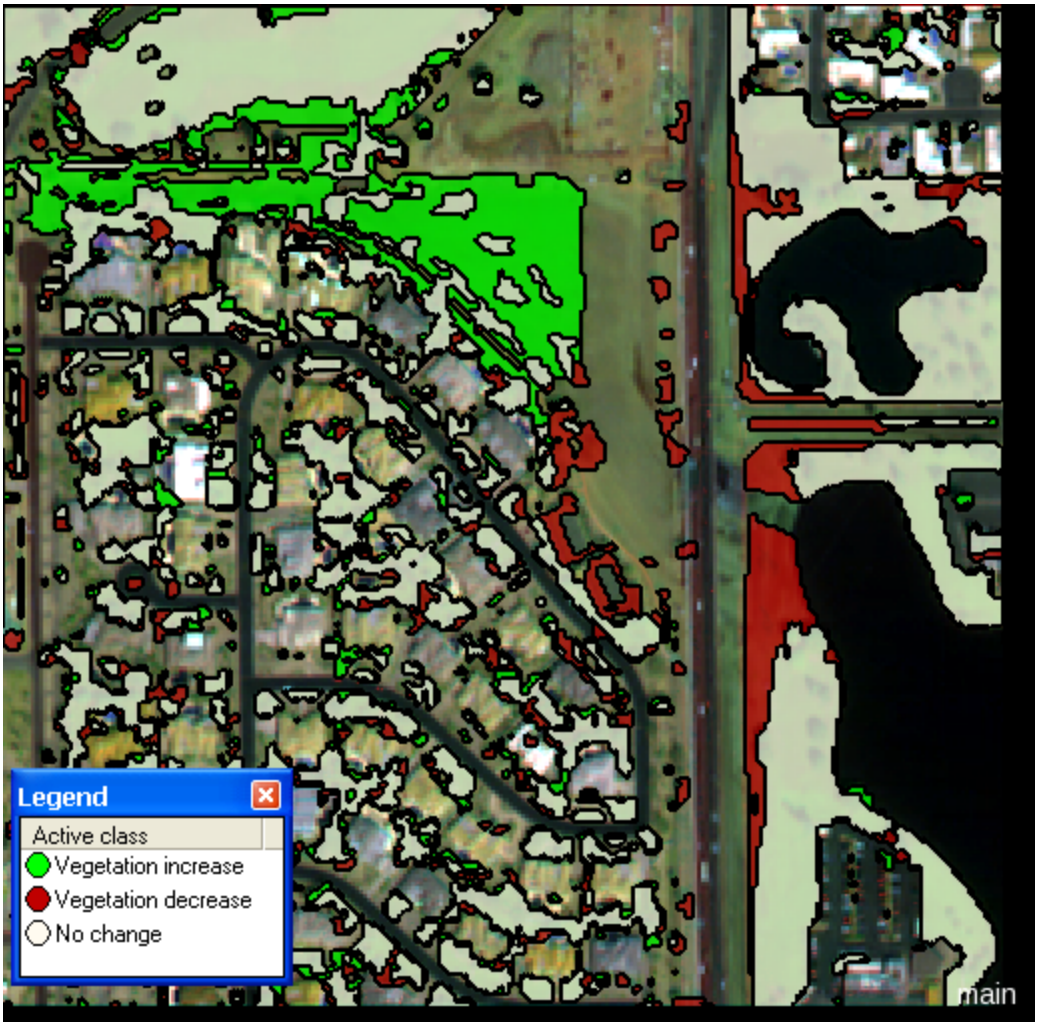



*Figure 43: Final results with three change classes.*

# Where to get additional help & information?

## The eCognition Community

The eCognition Community helps to share knowledge and information within the user, partner, academic and developer community to benefit from each other's experience.


The Community contains content such as:

- **Wiki**: collection of eCognition related articles (e.g. Rule Set tips and tricks, strategies, algorithm documentation...).
- **Discussions**: ask questions and get answers.
- **File exchange**: share any type of eCognition related code such as Rule Sets, Action Libraries, plug-ins...
- **Blogs**: read and write insights about what's happening around our industry…

Share your knowledge and questions with other users interested in using and developing image intelligence applications for Earth Sciences at:

http://community.ecognition.com/.

## The User Guide & Reference Book

Together with the software a User Guide and a Reference book is installed. You can access them in the Developer interface in the main menu 'Help>eCognition Developer User Guide' or Reference Book.

The Reference Book lists detailed information about algorithms and features, and provides general reference information.

## eCognition Training

eCognition Training Services offer a carefully planned curriculum that provides hands-on, real-world exercises. We are dedicated to enhancing customers' image analysis skills and helping these organizations to accomplish their goals.

Our courses are held in our classrooms around the world and on-site in our customer's facilities. We offer regular Open Training courses, where anyone can register and In-Company Training. We also offer Customized Courses to meet a customer's unique image analysis needs, thereby maximizing the training effect.

For more information please see our website or contact us at: eCognition_Training@trimble.com