

What's new eCognition Developer 8?

Working with Maps

New OBIA dimensions

DEEPER INSIGHTS
FASTER RESULTS
BETTER DECISIONS

www.earth.definiens.com



Imprint and Version

Document Version

Copyright © 2009 Definiens AG. All rights reserved.

Published by

Definiens AG
Trappentreustr. 1
D-80339 München
Germany

Phone +49-89-231180-0

Fax +49-89-231180-90

Web <http://earth.definiens.com>

Legal Notes

Definiens[®], **Definiens Challenger**[®] and **Definiens Cognition Network Technology**[®] are registered trademarks of Definiens AG in Germany and other countries. **Cognition Network Technology**[™], **Definiens eCognition**[®], **Enterprise Image Intelligence**[™], and **Understanding Images**[™], are trademarks of Definiens AG in Germany and other countries.

All other product names, company names, and brand names mentioned in this document may be trademark properties of their respective holders.

Protected by patents US 7146380, US 7117131, US 6832002, US 6738513, US 6229920, US 6091852, EP 0863485, WO 00/54176, WO 00/60497, WO 00/63788 WO 01/45033, WO 01/71577, WO 01/75574, and WO 02/05198. Further patents pending.

Table of Contents

Working with Maps	1
Imprint and Version	2
Legal Notes	2
Table of Contents	3
Introduction to this Module	4
This Module has two Lessons: 0Symbols at the side of the document	4
Symbols at the side of the document	4
Lesson 1 Introduction to Maps	5
1.1 About Maps	6
1.2 Application fields of Maps	6
Lesson 2 Using Maps: Example Change Detection	7
2.1 The Image Layers of the 'main' Map	8
2.2 Creating two independent Maps	10
2.2.1 Introduction to the algorithm 'copy map'	10
2.2.2 The Process settings to create the Map for T1	11
2.2.3 Execute the first Process, display and explore the new created a Map	12
2.2.4 The Process settings and execution to create the Map for T2	13
2.2.5 Evaluate both new a Maps	14
2.3 Classifying Vegetation on both Maps individually	16
2.3.1 The Process settings to classify Vegetation on MapT1	17
2.3.2 Execute and review the classification of MapT1	18
2.3.3 Execute Vegetation Classification for MapT2	19
2.3.4 Review the result	19
2.4 Synchronizing content of Maps	20
2.4.1 Introduction to the algorithm 'synchronize maps'	21
2.4.2 The Process settings to synchronize the content of MapT1	21
2.4.3 Execute the synchronization of MapT1 with main Map	23
2.4.4 Review the result	23
2.4.5 The Process settings and execution to copy the existing Level in the main Map	23
2.4.6 The Process settings to synchronize the content of MapT2	24
2.4.7 Execute the synchronization of MapT2 with main Map	26
2.4.8 Review the result of both synchronization steps	26
2.5 Applying the actual Change Detection	29
2.5.1 Main Map as Domain in the Parent Process	29
2.5.2 The Process settings to cookie-cut the outlines of LevelT1 in Level_T2	30
2.5.3 Execute and review the result	31
2.5.4 The Change Detection classification	32
2.5.5 Review the Result	33
2.6 Summary	34

Introduction to this Module

This Module gives you an introduction to a new function in eCognition Developer 8, the so called Maps. The functionality is explained on an example of Change Detection.

Goal of this Module is to give you an introduction to this new feature, the different algorithms you need to make use of it.

This Module has two Lessons:

Lesson 1 Introduction to Maps and Lesson 2 Using Maps: Example Change Detection.

Symbols at the side of the document

The symbols at the side of the document shall guide you through the exercises and help you to identify whether to read something or an action is needed or whether the screenshot is meant to be compared with settings in the software.

Introduction

If the side is hachured and 'Introduction' is added, this indicates that a text is giving a general introduction or methodology about the following chapter, method or exercise.

Information

If the side is hachured and 'Information' is added, this indicates that a text is giving information about the following exercise.



If this symbol is shown, you have to follow the numbered items in the text. If you just want to work through the exercises without reading the theory part, follow only this sign.

Action!



**Settings
Check**

If this symbol is shown, compare the settings shown in the screenshot with the settings in the according dialog box in the software.



**Rule Set
Check**

If this symbol is shown check the screenshot of the Process Tree with the content of the Process Tree in the software.



**Result
Check**

If this symbol is shown check the screenshot aside with the result in the software. It should look similar.

Lesson 1 Introduction to Maps

This Lesson has the following chapters

- *About Maps*
- *Application fields of Maps*

In eCognition Developer 8, you have the possibility to work with so called 'Maps'. A Map is a "Sub-Project" where you can process independently.

introduction

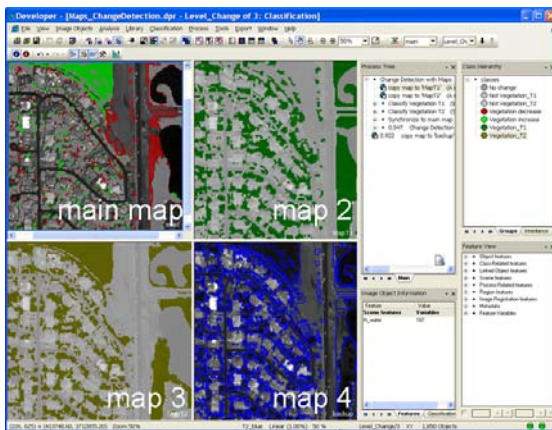
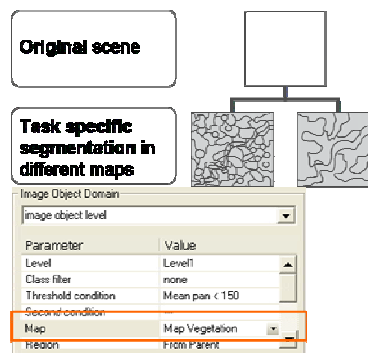


Figure 1: One Project with 4 different, independent Maps.

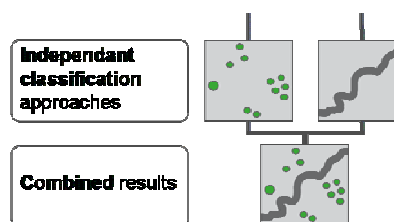
- Within one Project you can have **several** Maps.
- Maps are **independent** "Sub-Projects".
- The original scene is always the 'main' Map, all other, created Maps can have individual names.
- You can **define** Image Layers and resolution for a new Map.

1.1 About Maps

- You can create Maps directly when creating the Project (Create Project dialog box or Customized Import).
- You can create Maps using algor. '**copy map**'.
- **Process on individual** Maps only (new item in the Image Object Domain).



- You can exchange content between Maps using algorithm '**synchronize maps**'.



- You can delete Maps, if not needed anymore.

1.2 Application fields of Maps

Backup of original image objects:

During Rule Set development there is now the possibility to copy a certain stage of your processing in a Map and continuing on the main Map. After trying out a new segmentation or classification you can then **compare** the changes to the copied Map and also **recover** the original state by synchronizing the backup Maps content 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** only can be created, an exact subset will be then bases for the extend of the new Map.

Lesson 2 Using Maps: Example Change Detection

This Lesson has the following chapters

- The Image Layers of the 'main' Map
- Creating two independent Maps
- Classifying Vegetation on both Maps individually
- Synchronizing content of Maps
- Applying the actual Change Detection
- Summary

One application field of Maps is the Change Detection. To set up a Change Detection using Maps, there is a main Map containing **all image layers** from both points of time, 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.

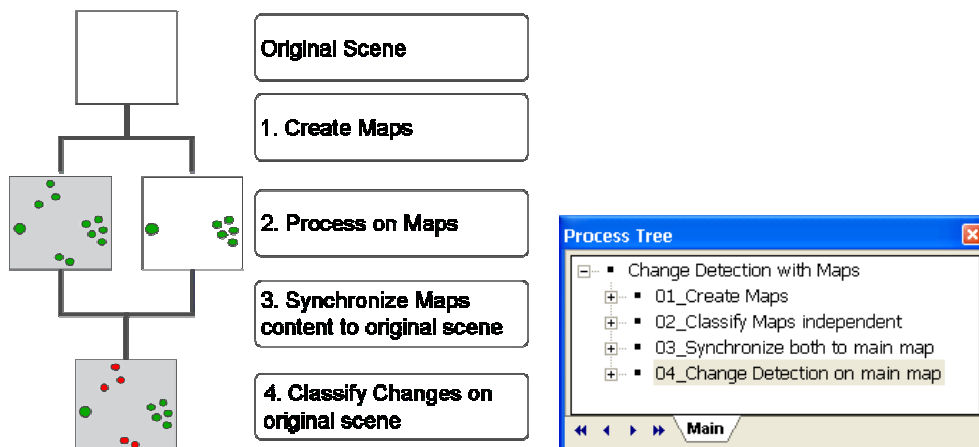


Figure 2: Left: schematic workflow of change detection using Maps; Right: overview over Rule Set sections.

introduction

2.1 The Image Layers of the 'main' Map



Preparation

Action!

1. Start Definiens eCognition Developer in 'Rule Set' mode.
2. Switch to predefined view setting number 4 'Develop rulesets'



3. In the main menu 'File' choose 'Open Project' or click on the 'Open Project' button in the toolbar.
4. Open the project '**Maps_ChangeDetection.dpr**' in the folder '**...\01_eCognitionDeveloper8_WhatsNew\03_Working_with_Maps**' at the location where the training data is stored.



Result Check

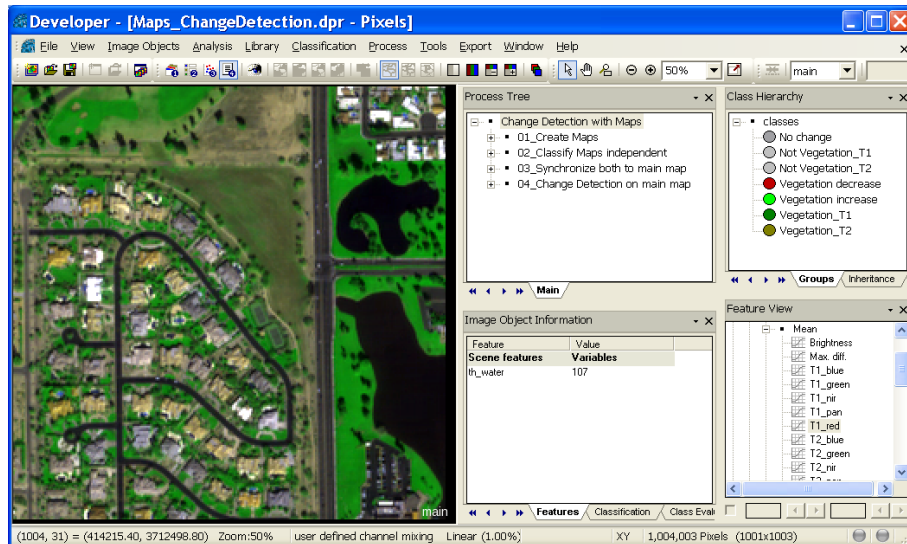


Figure 3: Loaded Project containing Rule Set, nothing has been processed yet.

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.

Information

Evaluate the loaded Image Layers

In the loaded Project, the multispectral layers of T1 are displayed. Check also the layers of T1.

Tip:

In the lower right corner of the Viewer, you see which Map is displayed currently. In our example right now, it is the 'main' Map.



Action!

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



Figure 4: The Image Layers of T1 are displayed.

2. 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.

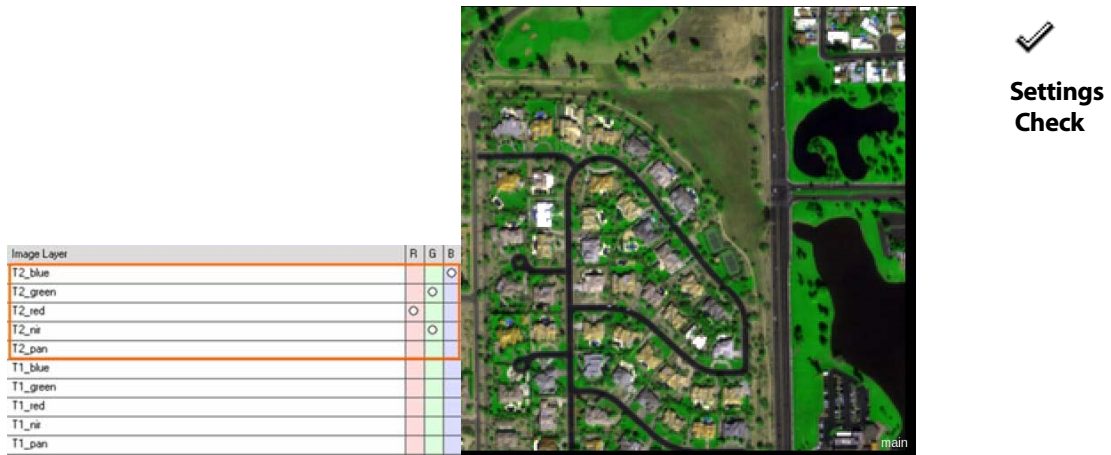


Figure 5: The Image Layers of T2 are displayed.

2.2 Creating two independent Maps

This Chapter has the following sub-chapters

- Introduction to the algorithm 'copy map'
- The Process settings to create the Map for T1
- Execute the first Process, display and explore the new created a Map
- The Process settings and execution to create the Map for T2
- Evaluate both new a Maps

In the Process section '01_Create Maps' the two Processes to create the necessary Maps for Vegetation classification of both points of time are stored.

2.2.1 Introduction to the algorithm 'copy map'

The most important functionalities of this algorithms are:

- In the field '**Source Region**' **1** it is defined whether you want to create a Map from the **full extend** of the original scene or if a **Region** is the basis for the new Map.
- In the field '**Target Map Name**' **2** the name of the new Map to be created is defined.
- If the Map to be created shall have a different resolution, this in the field '**Scale**' **3**.
- In the field '**Image Layers**' **4** 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**' **5** the thematic layers for the new Map are defined.
- If '**Yes**' is set in the field '**Copy Image Object Hierarchy**' **5** the existing Image Object Levels are copied in the new Map. E.g. if you want to have a back up map, you would use this option.

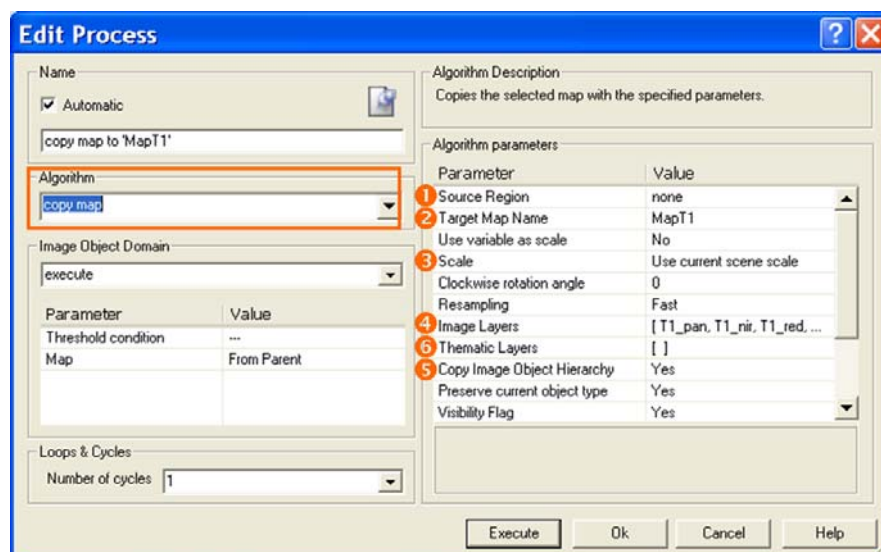


Figure 6: Process settings of algorithm 'copy map'.

2.2.2 The Process settings to create the Map for T1

1. Expand the Process section '01_Create Maps'.
2. Double-click on the first Child Process 'copy map to 'MapT1'' to open it.



Action!



Rule Set Check

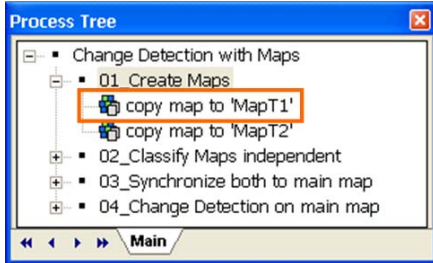
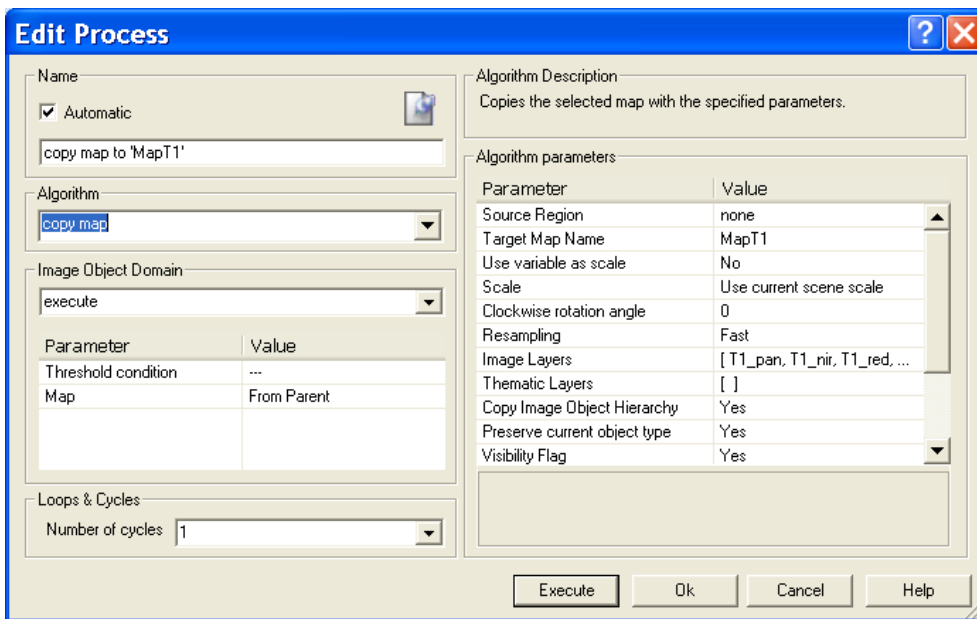


Figure 7: Process settings to create the MapT2.



Settings Check

Figure 8: Process Tree with Process to create the MapT2.

- The algorithm 'copy map' from the algorithm section 'Map Operations' is chosen.

Image Object Domain

- In the Image Object Domain the default settings are kept. No threshold must be set, no specific Map, as there is only one existing.

Algorithm Parameters

- **Source Region:** 'none' is kept, because no Region is existing, which could be the basis for the Map. The full extent of the loaded subset shall be copied in the new Map
 - In the field '**Target Map Name**' '**MapT1**' is defined. 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 Change Detection is needed.
 - In the field '**Image Layers**' only the layers from **T1** are chosen. The new Map will then contain only these layers.
3. Click on the '...' next to the 'Image Layers' field. The '**Select Image Layers**' dialog box opens.



Settings Check

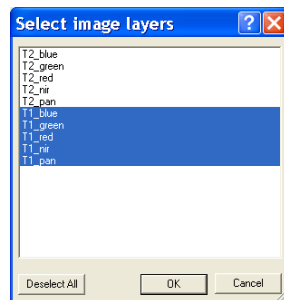


Figure 9: Only the T1 Layers are selected.

Tip:

To select multiple Layers, keep the **Ctrl** key pressed and click on the layers you want to choose.



Action!

4. Click on the 'Cancel' button to close the window.
- All other fields are also kept with their default settings.

2.2.3 Execute the first Process, display and explore the new created a Map

Execute the Process

1. **Close** the 'Edit Process' window.

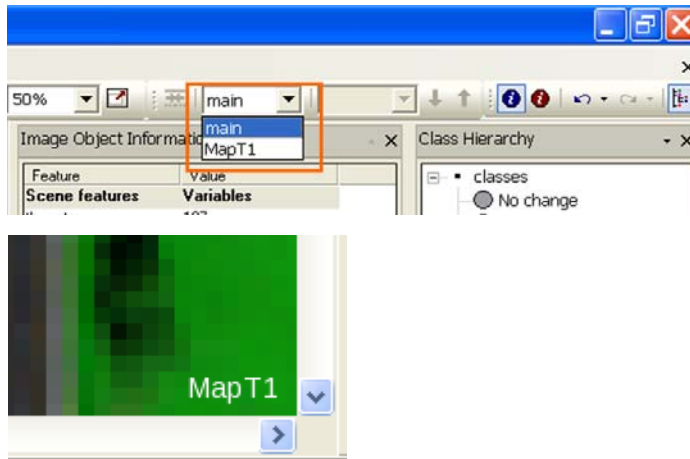


Action!

2. **Execute** the Process, by either right-clicking on it and select 'Execute' from the context menu or by selecting it and pressing **F5** on your keyboard.

Display the new created Map

3. To display a Map, use the **drop-down list** in the 'View Navigate' toolbar, right beside the 'Delete Level' button. Select **'MapT1'**.



**Result
Check**

Figure 10: In the Viewer, in the lower right corner, now 'MapT1' is displayed.

Check the image layers

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

Image Layer	R	G	B
T1_blue			○
T1_green		○	
T1_red	○		
T1_nir		○	
T1_pan			



**Settings
Check**

Figure 11: In the new Map 'MapT1' contains only the layers of T1, as specified in the Process 'copy map'.

2.2.4 The Process settings and execution to create the Map for T2

Explore the settings of the second Process

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



Action!



Rule Set Check

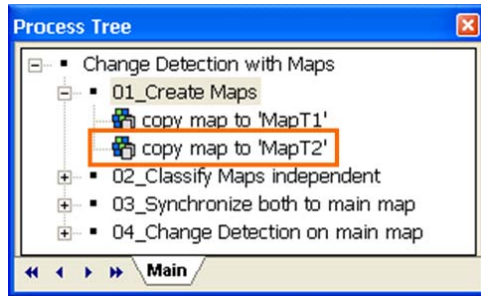


Figure 12: Process Tree with Process to create the MapT2.



Settings Check

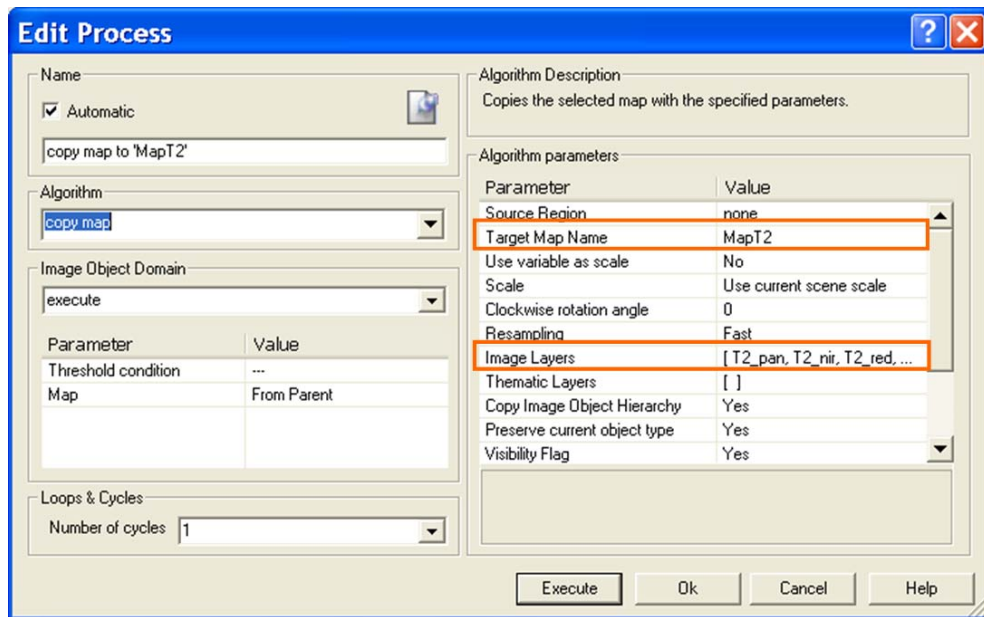


Figure 13: Process settings to create the MapT2.

- As for the Process before, the default settings are kept for the Image Object Domain.
- In difference to the Process before, here of course the **'Target Map Name'** is **'MapT2'**.
- In the field 'Image Layers' only the **layers of T2** are chosen.

Execute the Process



Action!

2. **Execute** the Process, by either right-clicking on it and select 'Execute' from the context menu or by selecting it and pressing **F5** on your keyboard.

2.2.5 Evaluate both new a Maps

Open and link a second Viewer to display both Maps

Information

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



Action!

1. To **open a second Viewer**, go to main menu **Window** and select either 'Split Horizontally' or **'Split Vertically'**.

- Click in the **left** Viewer window to make it active and select **MapT1** from the drop-down list in the 'View Navigate' Toolbar.

Tip:

The currently active Viewer is marked with a blue outline.



- Click in the **right** Viewer window to make it active and select **MapT2** from the drop-down list in the 'View Navigate' Toolbar.
- Go again to the main menu 'Windows' and select '**Side by side View**'.

Explore both Maps

- Zoom in the Maps and compare the differences in both Maps. Especially the Vegetation is quite different in the two Maps.



**Result
Check**

Figure 14: Two Maps are create: MapT1 and MapT2.

2.3 Classifying Vegetation on both Maps individually

This Chapter has the following sub-chapters

- ➔ The Process settings to classify Vegetation on MapT1
- ➔ Execute and review the classification of MapT1
- ➔ Execute Vegetation Classification for MapT2
- ➔ Review the result

Information

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 **Image Object Domain**, where Maps are a new item in eCognition 8.

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.

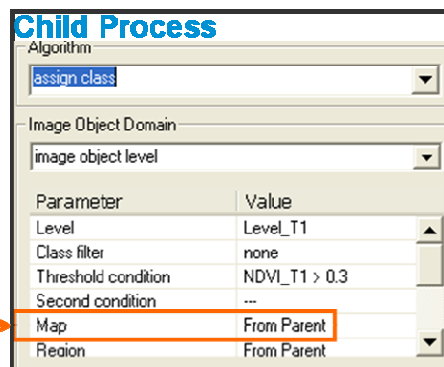
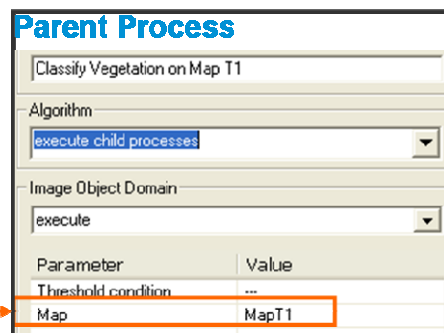
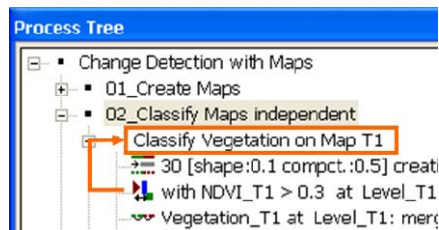


Figure 15: In the domain of the Parent Process MapT1 is specified. All Child Processes refer to this domain.

2.3.1 The Process settings to classify Vegetation on MapT1

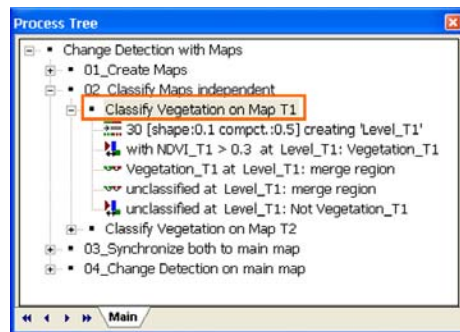
1. Expand the Process section '**02_Classify Maps independent**' and also '**Classify Vegetation on Map T1**'.



Action!

Explore the Parent Process

2. Double-click on the Process '**Classify Vegetation on Map T1**' to open it.



Settings Check



Rule Set Check

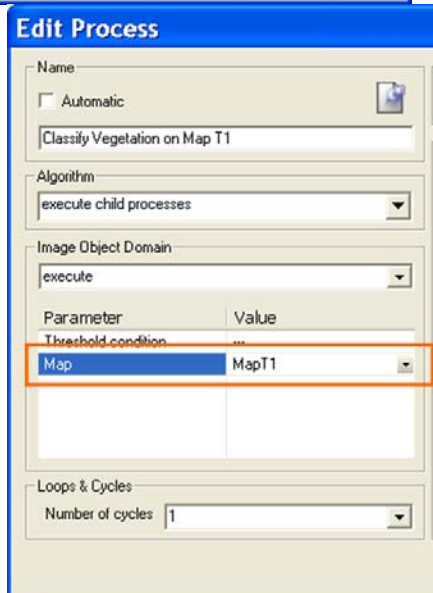


Figure 16: In the domain of the Parent Process MapT1 is specified. All Child Processes refer to this domain.

- As algorithm 'execute child processes' is chosen.
 - In the Image Object Domain of the Parent Process, 'MapT1' is selected from the drop-down list.
3. Click on the 'Cancel' button to close the window.



Action!

Explore Image Object Domain of Child Processes

1. 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.
2. Click on the 'Cancel' button to close the window.

3. Double-click on the second Child Process '**with NDVI_T1 > 0.3 at Level_T1: Vegetation_T1**' to open it.



Settings Check

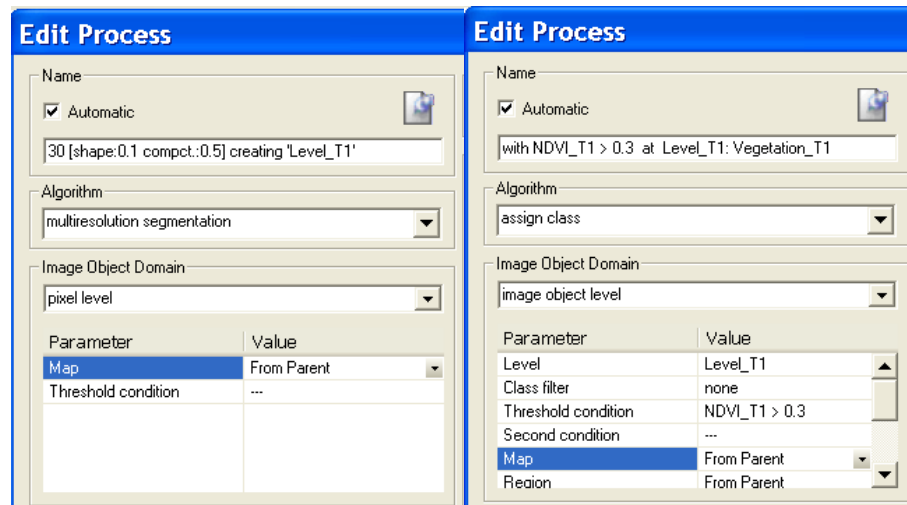


Figure 17: Both Processes point in their Image Object Domain to whatever Map is specified in the Parent Process. Here MapT2 is defined the Parent Process as domain.

2.3.2 Execute and review the classification of MapT1



Action!

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 new functionality '**Transparent/non-transparent outlined object**'.



Figure 18: New functionality: 'Transparent/non-transparent outlined object'.

Review the classified Map



Result Check

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.

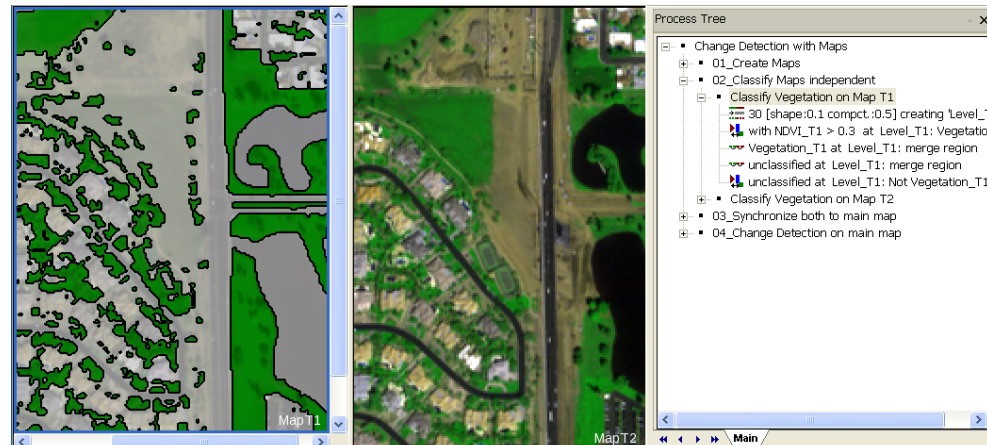


Figure 19: MapT1 is classified, MapT2 not.

2.3.3 Execute Vegetation Classification for 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. Right-click on 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.



Action!



Rule Set Check

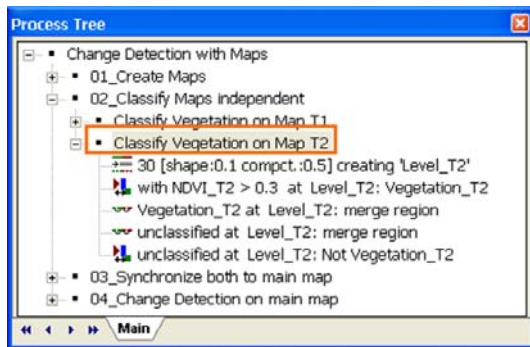


Figure 20: Process Tree with section for classifying Vegetation on MapT2.

2.3.4 Review the result

2. Activate the Viewer with MapT2.
3. Switch on the **Classification View** and **transparency** on , additionally you can switch on the new functionality '**Transparent/non-transparent outlined object**'.

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



Result Check

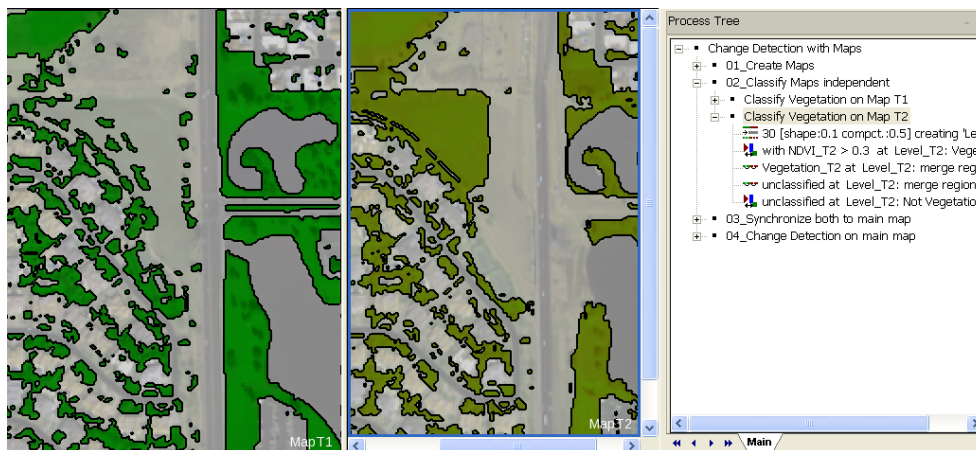


Figure 21: In the left Viewer result for MapT1 is shown, in the right Viewer the result for MapT2.

2.4 Synchronizing content of Maps

This Chapter has the following sub-chapters

- Introduction to the algorithm 'synchronize maps'
- The Process settings to synchronize the content of MapT1
- Execute the synchronization of MapT1 with main Map
- Review the result
- The Process settings and execution to copy the existing Level in the main Map
- The Process settings to synchronize the content of MapT2
- Execute the synchronization of MapT2 with main Map
- Review the result of both synchronization steps

Introduction

After both Maps have been classified, now the results of both are synchronized back to the main Map. The two Levels with their classification of Vegetation are then in a later step basis for Change Detection classification.

To **synchronize multiple Levels** (LevelT1 and LevelT2) in one (main) Map, it must be defined to which place of the hierarchy the Levels it should be synchronized.

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 is overwriting 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 where the second synchronization process points to.

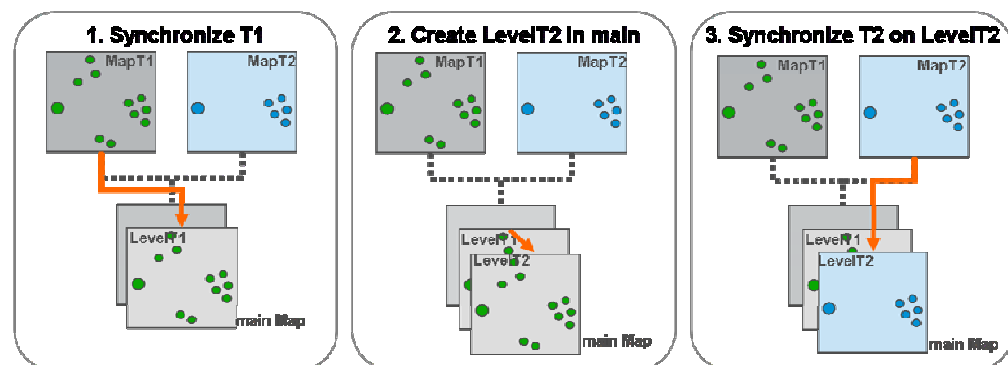


Figure 22: Three steps to synchronize content of two Maps in the main Map.

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 **1**, from which **Map** and **Level** the content shall be copied.
- It can be defined in the Algorithm Parameters **2** 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 **3**.

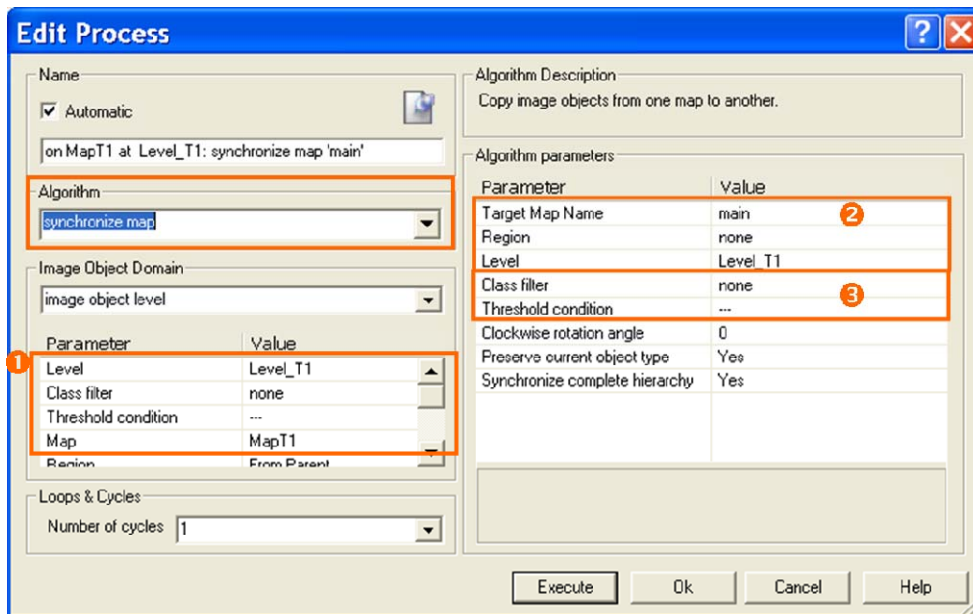


Figure 23: Process settings of algorithm 'synchronize maps'.

2.4.2 The Process settings to synchronize the content of MapT1

1. Expand the Process section '**03_Synchronize both to main map**'.
2. Double-click on the first Child Process '**on MapT1 at Level_T1: synchronize map 'main'**' to open it.



Action!



Rule Set Check

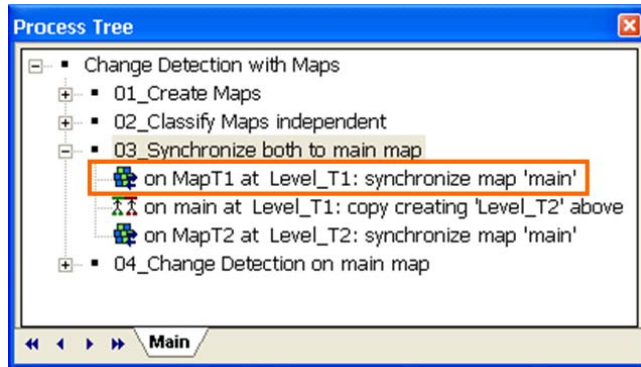


Figure 24: Process Tree with Process to synchronize content of MapT2 to main Map.



Settings Check

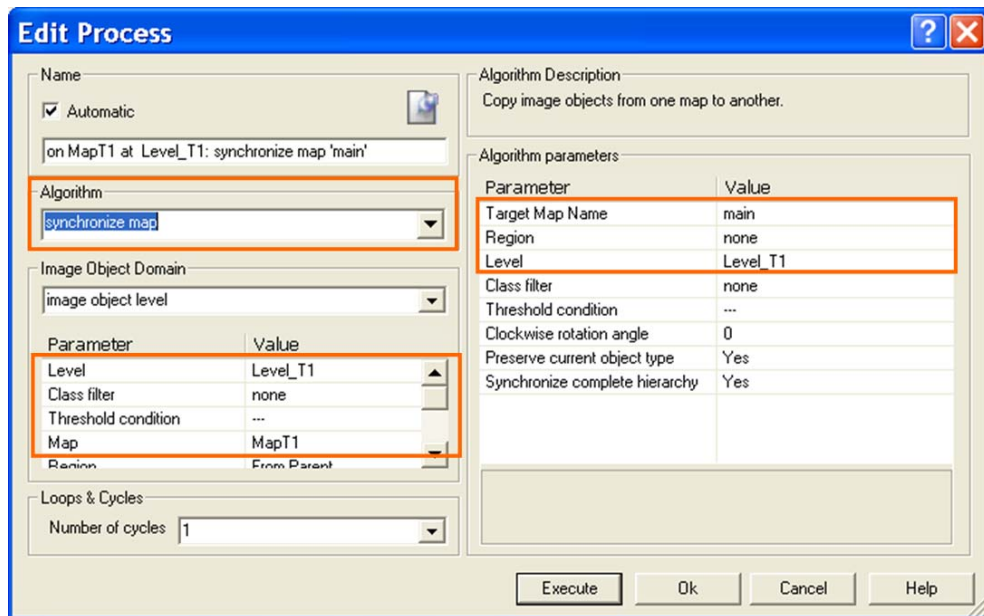


Figure 25: Process settings to synchronize content of MapT1 to main Map.

Image Object Domain

The Image Object Domain here defines the **source of synchronization**.

- As Level '**Level_T1**' is chosen.
- As Map '**MapT1**' is chosen.

Algorithm Parameters

The Algorithm Parameters here 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.

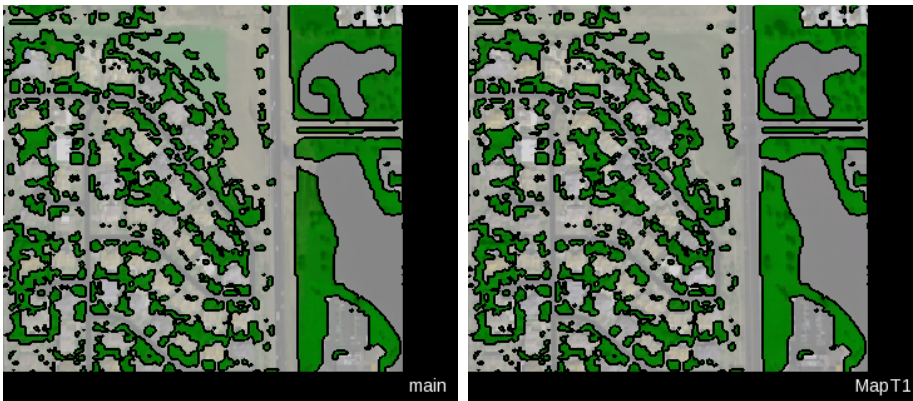
2.4.3 Execute the synchronization of MapT1 with main Map

1. Execute the Process **'on MapT1 at Level_T1: synchronize map 'main'** by either right-clicking on it and select 'Execute' from the context menu or by selecting it and pressing **[F5]** on your keyboard.
2. Display in one Viewer MapT1 and in one Viewer main Map.



Action!

2.4.4 Review the result



Result Check

Figure 26: The content of MapT1 is now also available in main Map.

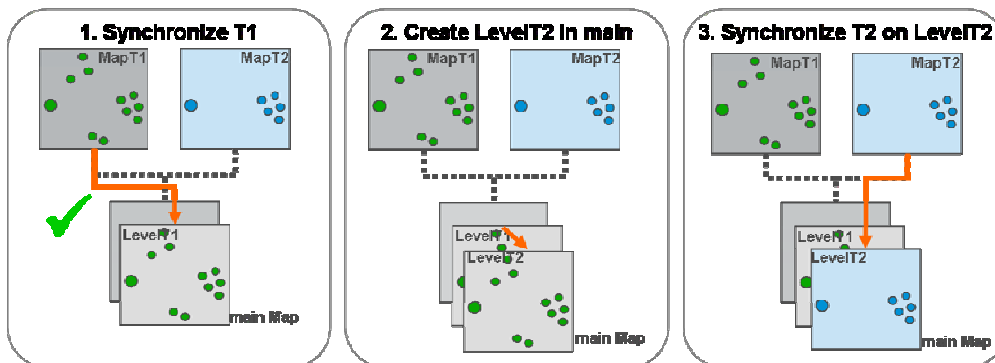


Figure 27: The second step of the synchronization Process is to create a Level in the main Map.

2.4.5 The Process settings and execution 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



Action!



Settings Check

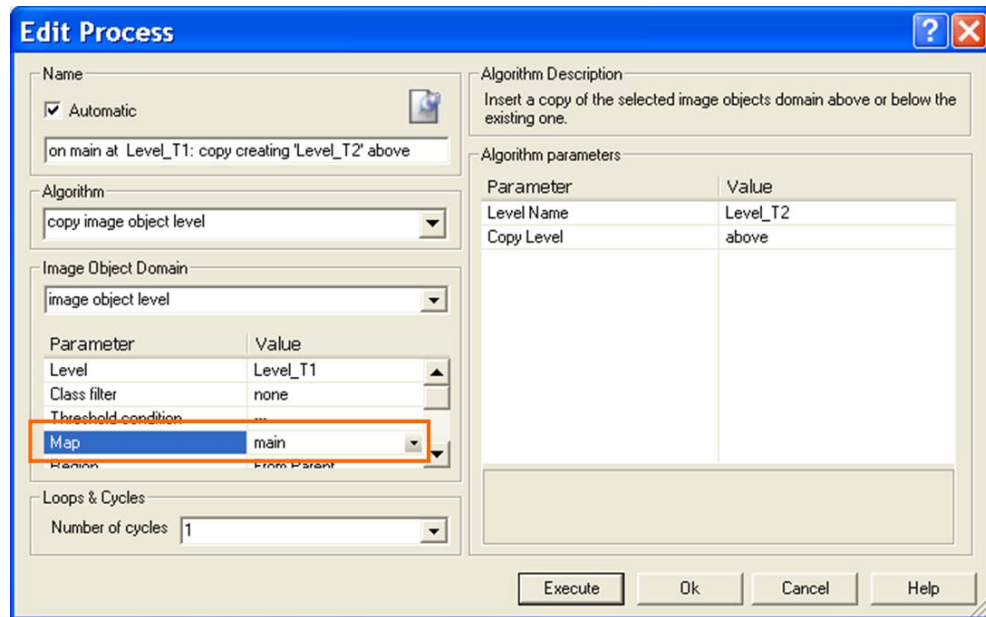


Figure 28: Process settings to copy 'Level_T1' above. In the Image Object Domain, it is specified that this shall executed only in the main Map.

Image Object Domain

- In the field 'Level', it is specified that 'Level_T1' shall be copied.
 - In the field 'Map' it is defined that the Process is be executed in the **main Map, not in any of the others.**
2. Execute the Process.



Action!

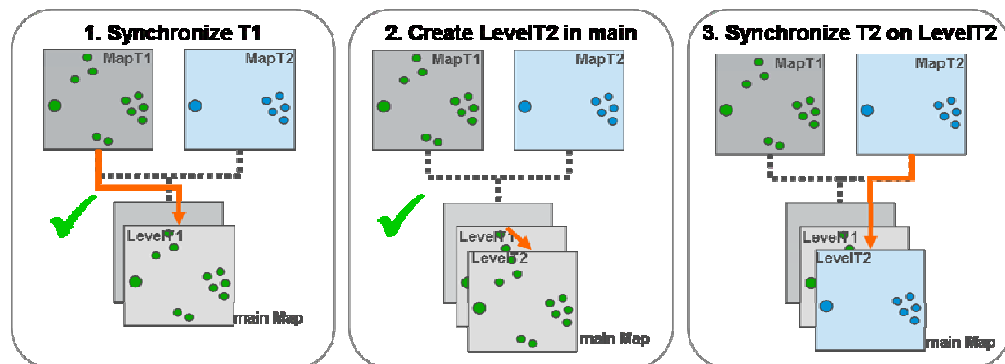


Figure 29: The second step of the synchronization Process is to create a Level in the main Map.

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

2.4.6 The Process settings to synchronize the content of MapT2



Action!

1. Double-click on the third Child Process '**on MapT1 at Level_T1: synchronize map 'main'**' to open it.

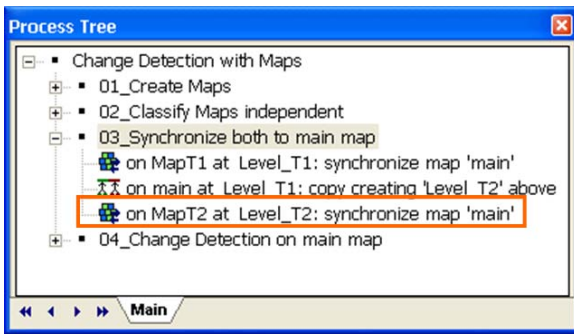


Figure 30: Process Tree with Process to synchronize content of 'MapT2' to main Map.

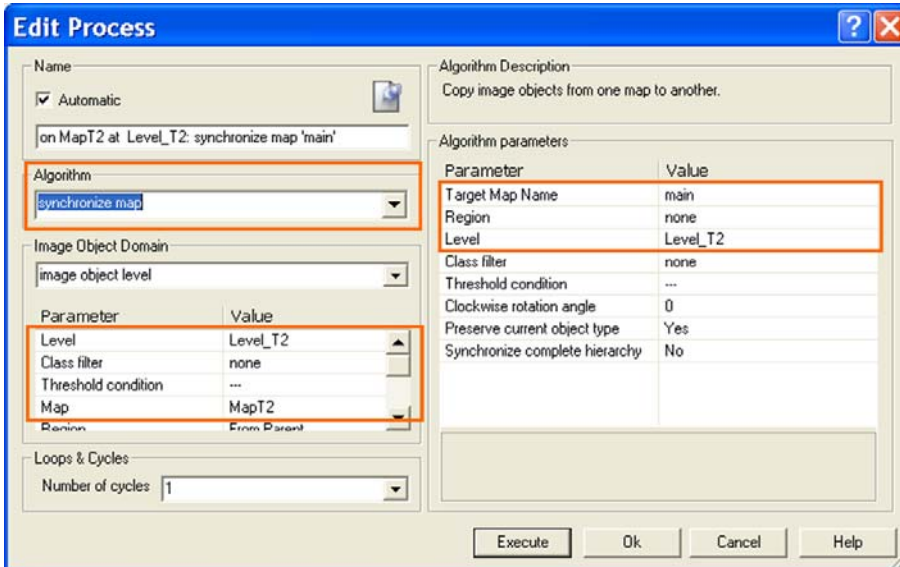


Figure 31: Process settings to synchronize content of MapT2 to main Map.

Image Object Domain

The Image Object Domain here defines the **source of synchronization**.

- As Level '**Level_T2**' is chosen.
- As Map '**MapT2**' is chosen.

Algorithm Parameters

The Algorithm Parameters here 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.
2. Close the Process by clicking on 'Cancel'.



2.4.7 Execute the synchronization of MapT2 with main Map

1. Execute the Process 'on MapT2 at Level_T2: synchronize map 'main"' by either right-clicking on it and select 'Execute' from the context menu or by selecting it and pressing **F5** on your keyboard.

2.4.8 Review the result of both synchronization steps

2. Open 4 Viewers by selecting additionally i.e. 'Split Horizontally' from the main menu 'Window'.
3. Display in the **upper left** Viewer the **main map** and **Level_T1**, in the **upper right main map** and **Level_T2**. Display in the **lower left** Viewer the **MapT1**, in the **lower right** MapT2.



Result Check

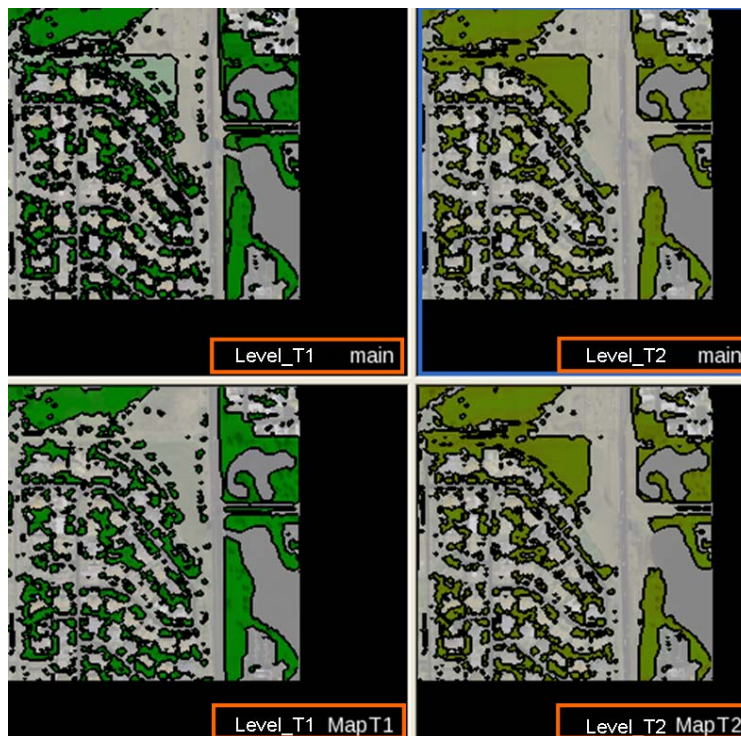


Figure 32: Upper left: Level_T1 in the main Map; Upper right: Level_T2 in the main Map; Lower left: Level_T1 in the MapT1; Lower right: Level_T1 in the MapT1.

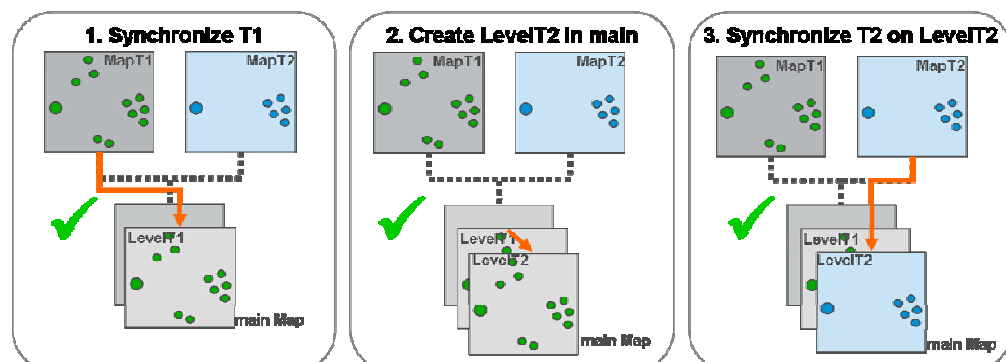


Figure 33: All three steps to synchronize are executed.

2.5 Applying the actual Change Detection

This Chapter has the following sub-chapters

- *Main Map as Domain in the Parent Process*
- *The Process settings to cookie-cut the outlines of Level_T1 in Level_T2*
- *Execute and review the result*
- *The Change Detection classification*
- *Review the Result*

Now that in the main Map both analysis results from T1 and T2 are available, the actual Change detection can be applied.

The **main Map as domain** is set 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.

Information

2.5.1 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** are pointing to this **Parent Process Domain**.

Information

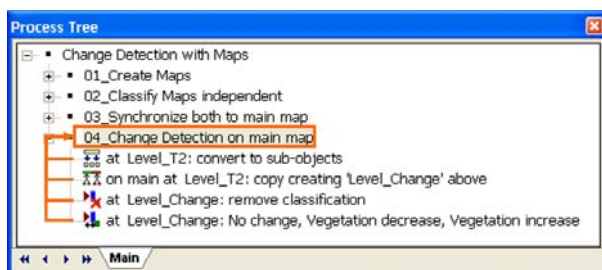


Figure 34: The Child Processes get the information to process only on the main Map from the Parent Process.

Information

2.5.2 The Process settings to cookie-cut the outlines of Level_T1 in 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 **cookie-cut 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'**.

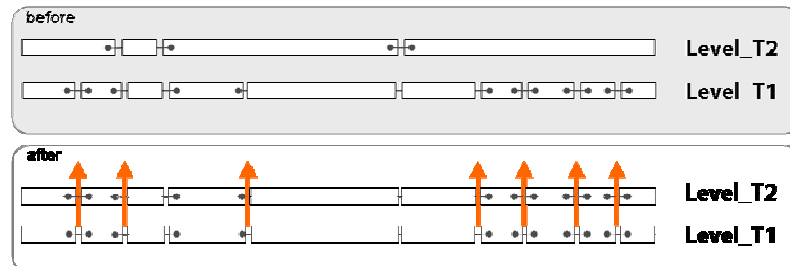


Figure 35: First the outlines of 'Level_T2' are different from 'Level_T1', after executing the Process, the outlines are cut into the upper Level.



Action!



Rule Set Check

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.

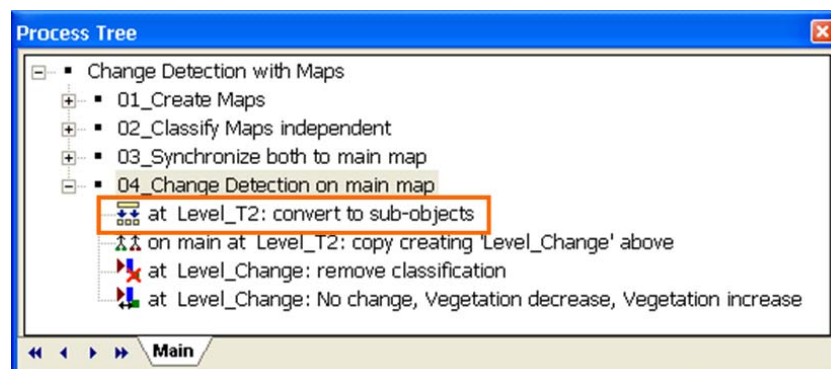


Figure 36: Process Tree with Process cookie-cut the outlines from 'Level_T1' in 'Level_T2'.



Settings Check

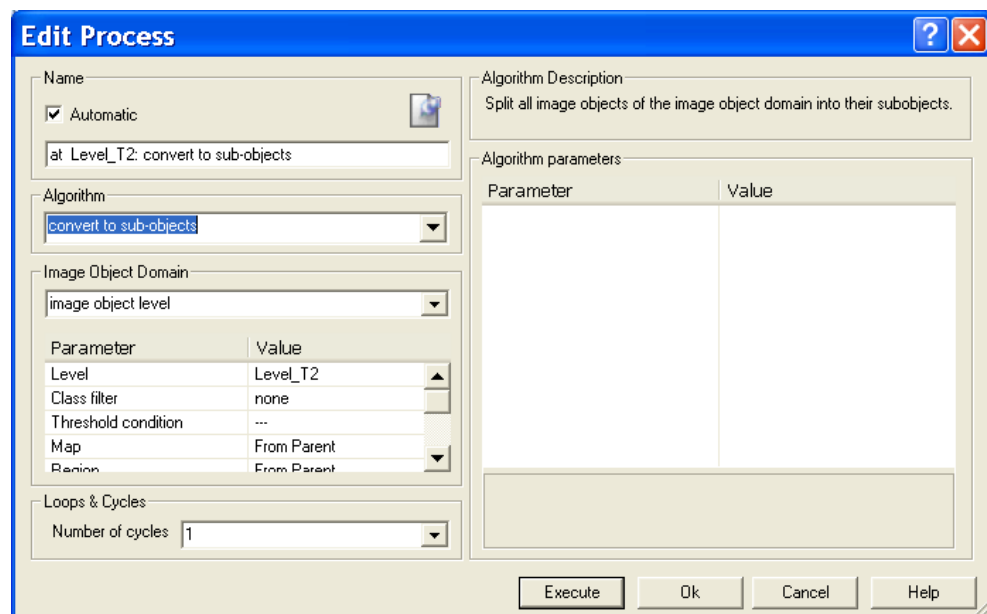


Figure 37: Process settings to cookie-cut the outlines from 'Level_T1' in 'Level_T2'.

Image Object Information

- In the Image Object Information 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.

2.5.3 Execute and review the result

3. Execute the Process.



Figure 38: The outlines of the underlying Level are now also cut in the upper Level.



**Result
Check**

After executing the 'convert to sub-objects' Process, the '**Level_T2**' represents **also the outlines of 'Level_T1'**. Now the Object Hierarchy is ready to be copied in an extra Level to classify the changes.

2.5.4 The Change Detection classification

Information

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.

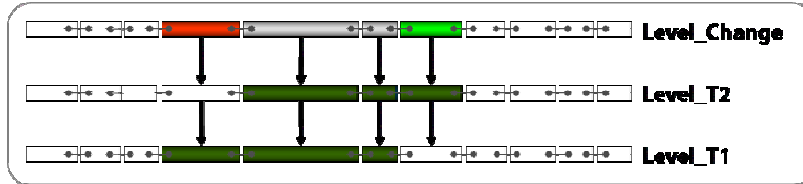
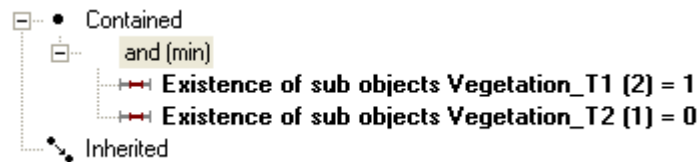
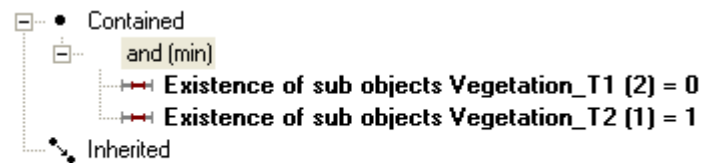


Figure 39: For the classification of the Changes, the feature 'Existence of sub-objects' is used.

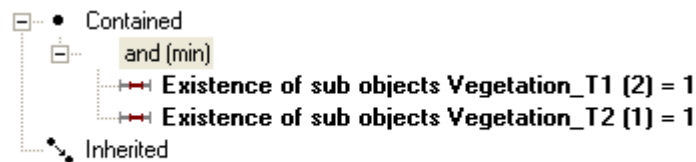
- Vegetation increase:



- Vegetation decrease:



- No change:



Preparation



Action!

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'.

Execute the Change Detection Process

3. Execute the classification Process **'at Level_Change: No change, Vegetation decrease, Vegetation increase'**.

2.5.5 Review the Result

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



**Result
Check**

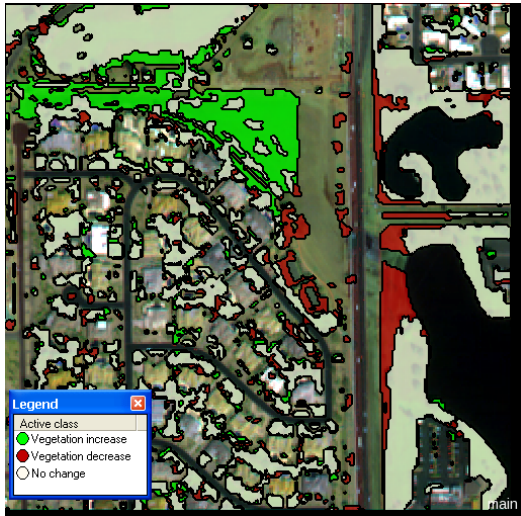


Figure 40: Classification view of the final Change Detection Level in the main Map.

2.6 Summary

Create two independent Maps

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 41: Two Maps are create: MApT1 and MapT2.

Classify Vegetation 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.

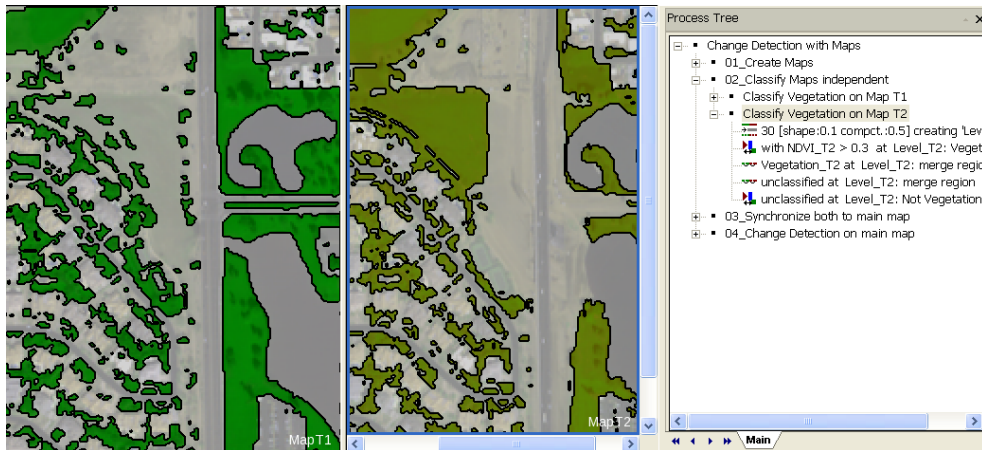


Figure 42: In the left Viewer result for MapT1 is shown, in the right Viewer the result for MapT2. Both Image Object Levels are independently segmented and classified.

Synchronize the content of both Maps

In three steps the synchronization is applied. The content of the first Map is copied to the main map using the algorithm 'synchronize maps', then a Level is copied in the main Map, as a third step the content of the second Map is copied to the new created Level in the main Map.

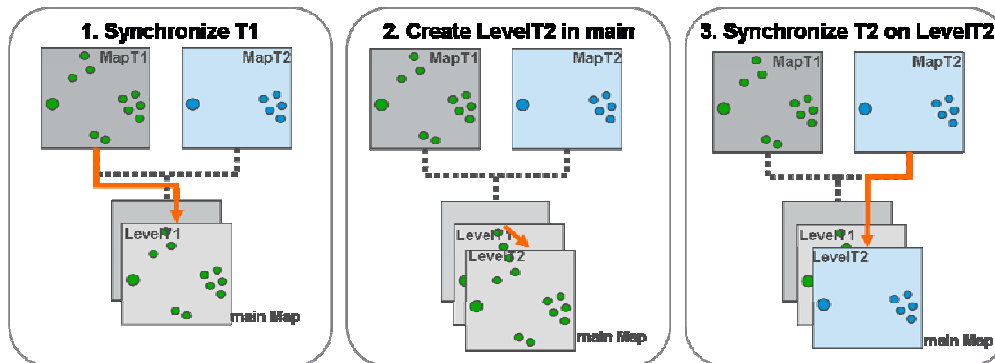


Figure 43: Schematic workflow to synchronize Maps.

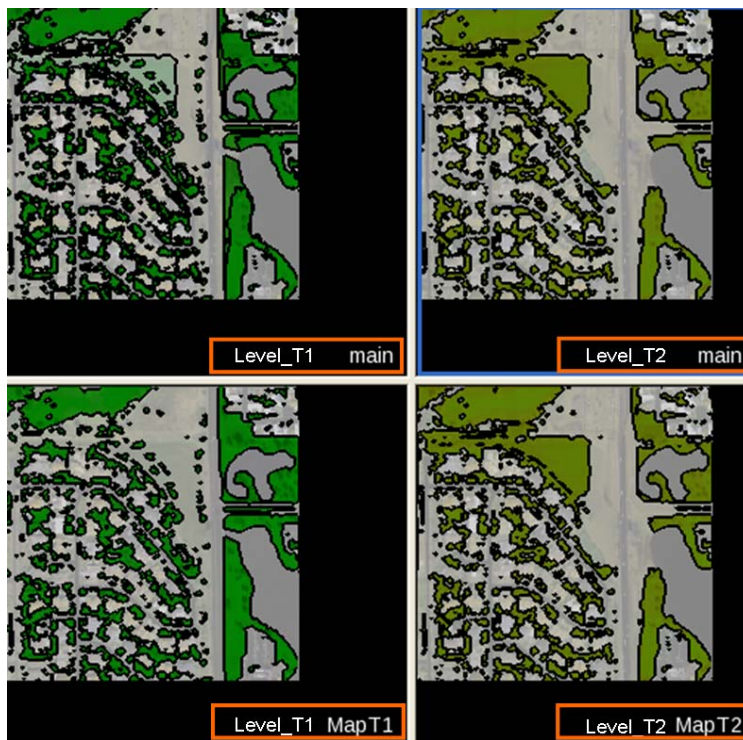


Figure 44: Upper left: Level_T1 in the main Map; Upper right: Level_T2 in the main Map; Lower left: Level_T1 in the MapT1; Lower right: Level_T2 in the MapT1

Apply the actual Change detection

The actual Change Detection is applied in the main Map. The Object Hierarchy has to be equalized for all Levels. The Class-Related Feature 'Existence of sub-objects' is used to identify changes.

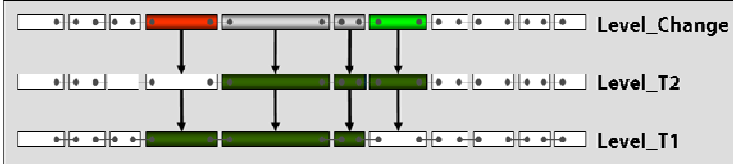


Figure 45: For the classification of the Changes, the feature 'Existence of sub-objects' is used.

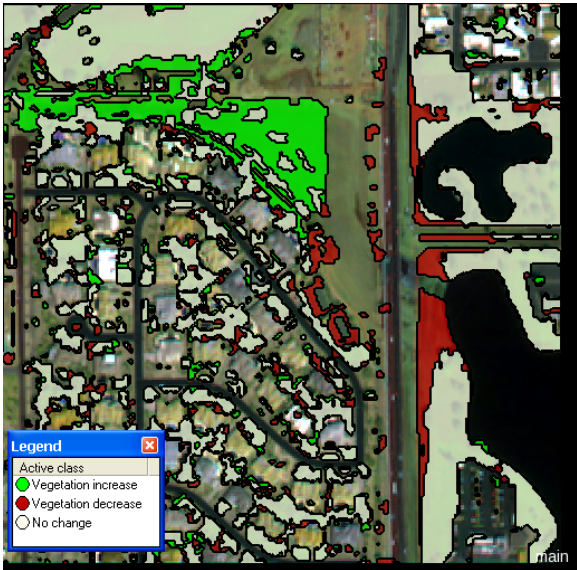


Figure 46: Classification view of the final Change Detection Level in the main Map.