SIEMENS

SIMIT 7

Migration

User Manual

Issue Edition

January 2013

Siemens offers simulation software to plan, simulate and optimize plants and machines. The simulation- and optimization-results are only non-binding suggestions for the user. The quality of the simulation and optimizing results depend on the correctness and the completeness of the input data. Therefore, the input data and the results have to be validated by the user.

Trademarks

SIMIT ® is a registered Trademark of Siemens AG in Germany and in other countries.

The remaining trademarks in this publication may be trademarks use of which by third parties for their own purposes could violate the rights of the owner.

Copyright © Siemens AG 2013 All rights reserved

The reproduction, transmission or use of this document or its contents is not permitted without express written authority. Offenders will be liable for damages. All rights, including rights created by patent grant or registration of a utility model or design, are reserved.

Siemens AG
Industry Sector
Industry Automation Division
Process Automation

SIMIT HB-V7MIG-2013-01-en

Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Because deviations cannot be precluded entirely, we cannot guarantee full agreement. However, the data in this manual are reviewed regularly and any necessary corrections included in subsequent editions. Suggestions for improvement are welcomed.

© Siemens AG 2013

Subject to change withour prior notice.

Content

Copyright © Siemens AG, 2013

List of Figures

Copyright © Siemens AG, 2013

List of Tables

1 PREFACE

1.1 Target Group

This manual addresses you as a user of the SIMIT simulation system. Here you will get to know how to use projects in SIMIT 7 which were originally created with SIMIT 5.

As a prerequisite you need to be familiar with the general use of a personal computer and windows. Furthermore, knowledge of both SIMIT 5 and SIMIT 7 is required.

1.2 Content

This manual describes all aspects relevant for migrating projects, macro components and templates from SIMIT 5 to SIMIT 7.

In the introductory chapter 2 general aspects concerning migration of SIMIT projects are described and a step-by-step procedure for migrating projects is shown.

Chapter 3 shows how to migrate self-written component types.

Chapter 4 shows how to migrate diagrams and operating windows. Since migration of macro components and templates is based on migration of diagrams and operating windows, this chapter is essential for all aspects described in the chapters to follow.

Chapter 5 shows how to migrate gateways.

Chapter 6 shows both a step-by-step migration of macro components as well as migration of diagrams that contain macro components.

Chapter 7 shows how to migrate templates.

Chapter 8 concludes with a description of how the handling of ACI-scripts has changed.

1.3 Symbols

Particularly important information is highlighted in the text as follows:

➤ **NOTE**

Notes contain important supplementary information about the documentation contents. They also highlight those properties of the system or operator input to which we want to draw particular attention.

⚠ **CAUTION**

This means that the system will not respond as described if the specified precautionary measures are not applied.

Copyright © Siemens AG, 2013

STOP **WARNING**

This means that the system may suffer irreparable damage or that data may be lost if the relevant precautionary measures are not applied.

Copyright © Siemens AG, 2013

2 MIGRATING SIMIT PROJECTS

SIMIT 7 provides a full redesign not only with respect to the graphical user interface. There are also many enhancements with respect to functional features compared to SIMIT 5. Although the basic ideas of SIMIT have remained almost untouched, some changes with respect to SIMIT 5 have taken place. These changes as well as reworked file formats of SIMIT projects make SIMIT 5 projects incompatible to SIMIT 7 projects. However, projects created with SIMIT 5 can be used in SIMIT 7 after migration.

2.1 Project items to be migrated

SIMIT 5 projects contain of different items such as diagrams, operating windows, components, gateways etc. When migrating a SIMIT 5 project the following items need to be handled:

- **Component types**

  User components that have been created with SIMIT 5 can be migrated with the component type editor (CTE) in SIMIT 7. Information about migrating component types and specifics of migration can be found in the component type editors manual.

  Component types that are part of the SIMIT 5 standard library require no migration since SIMIT 7 provides equivalent component types and controls in its standard library.

- **Diagrams and operating windows**

  Diagrams in SIMIT 7 comprise both diagrams and operating windows as known in SIMIT 5. Hence both diagrams and operating windows will be turned into diagrams in SIMIT 7, specifics are described in chapter 3.

- **Macro components**

  In SIMIT 5 a macro component is a diagram that provides connection points resulting from macro connectors. Macro components are created using the diagram editor. In SIMIT 7 macro components are diagrams that are created using the macro editor. Definition of a macros connection points does not require macro connectors any more.

  Macro components are migrated similar to diagrams, a description can be found in chapter 6.

- **Templates**

  In SIMIT 5 a template is a diagram or an operating window that contains a placeholder instead of a value as signal, parameter etc. Templates are created using the diagram editor or the operating window editor. In SIMIT 7 templates are diagrams that use placeholders, they are edited using the template editor.

  Like for macro components, migration of templates is based on the migration of diagrams and is described in chapter 7.

- **Gateways**

  Gateways need to be newly created in SIMIT 7. In case of a Profibus DP gateway or a Profinet IO gateway the hardware configuration needs to be imported from the SIMATIC project – just like in SIMIT 5.

Copyright © Siemens AG, 2013

The gateways signals have a wider range of data types in SIMIT 7 compared to SIMIT 5. Furthermore signals in SIMIT 7 are assigned to a source. Chapter 5 explains specifics resulting hereof and shows how to migrate a gateway.

- **Signal Groups**

  SIMIT 7 does not provide signal groups any more, hence signal groups cannot be migrated.

  Signals groups were replaced with a powerful signal archive in SIMIT 7. This archive is part of the additional module “Trend and Messaging Editor” (TME).

- **Snapshots**

  Snapshots cannot be migrated but need to be newly created in SIMIT 7.

2.2 Migrating projects step-by-step

To migrate a SIMIT 5 project, please follow these steps:

Step 1:

Migrate all user components that are used in this project. Detailed information can be found in the component type editor’s manual for SIMIT 7.

Step 2:

Migrate all macro components that are used in this project. Proceed as described in chapter6.

Step 3:

Export the project from SIMIT 5.

Caution

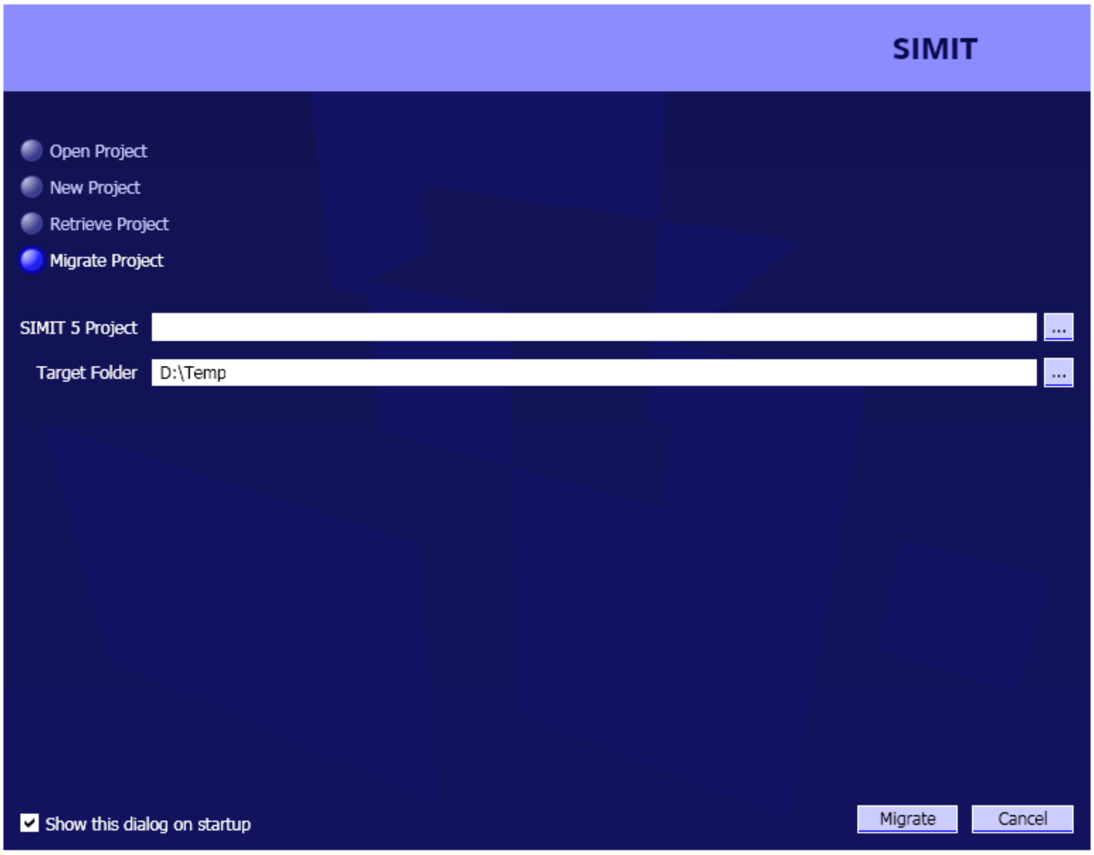
Please note that only SIMIT projects can be migrated that were exported with SIMIT 5.4 **SP1**!

Step 4:

Import the project you just exported in SIMIT 5 using the menu *Project* | *Migrate* in SIMIT 7 in order to migrate it.

You will see a dialog as shown in Figure 2-1. Just select the SIMIT 5 project to be migrated and provide a target folder for the resulting SIMIT 7 project.

Copyright © Siemens AG, 2013



SIMIT 5 Project

Target Folder D:\Temp

☑ Show this dialog on startup

Figure 2-1: Dialog to migrate a project

Step 5:

Configure all required gateways in the SIMIT 7 project (see chapter 5).

Step 6:

Rework the project in SIMIT 7. Inconsistencies within the migrated project will be shown in the consistency check. A description of possible inconsistencies and strategies about how to solve them can be found in chapter 3.

Templates are independent of a specific project, so they can be migrated independent from a project. Please see chapter 3 for detail.

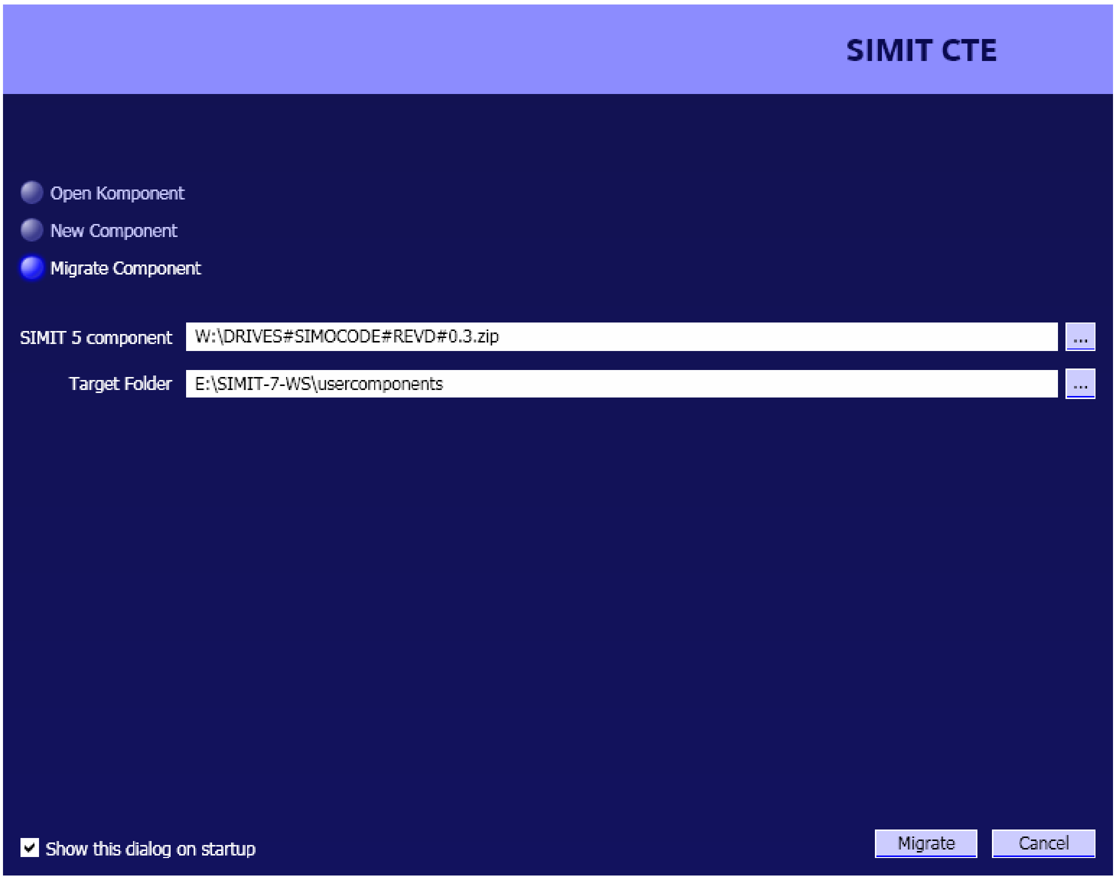
2.3 Log file

Upon migrating a project a log file *protocol.txt* will be created in the subfolder *migration* within the workspace folder of SIMIT 7. This file contains information about where you might need to rework the migrated project. After migrating please open this file in a text editor and search for the keyword “warning”.

3 MIGRATING COMPONENT TYPES

Component types that were exported using SIMIT 5.4 SP1 can be converted into SIMIT 7 format by use of the SIMIT 7 Component Type Editor CTE.

Select *Migrate Component* in the dialog as shown in Figure 4-1 and select the component to be migrated in the file chooser.



SIMIT 5 component: W:\DRIVES#SIMOCODE#REVD#0.3.zip ...

Target Folder: E:\SIMIT-7-WS\usercomponents ...

☑ Show this dialog on startup

Figure 3-1: Dialog for reading component types in the CTE

Necessary modifications to the component are done automatically as far as possible. Since some concepts have changed fundamentally in SIMIT 7 further manual editing of the component type may be necessary. The navigation tree within the component type editor will make use of an overlay (⚠) to indicate sections which have caused problems.

In the chapters to follow aspects are described that usually require further manual editing. You are provided with advice how to modify the migrated component type.

3.1 Data types

The basic data types *logical* and *double* are called *binary* and *analog* in SIMIT 7, respectively. As far as possible, renaming will be performed automatically. All additionally used data types need to be adapted in the behaviour description according to Table 3-1:

.

Copyright © Siemens AG, 2013

SIMIT 5	SIMIT 7
dimension	int
BYTE	sbyte
INT2	short
INT4	int
INT8	long
WORD	short
DOUBLE	double
UBYTE	byte
UINT2	ushort
UINT4	uint
UINT8	ulong

Table 3-1: Migrating data types

3.2 Variable definitions

Both keywords *static* and *const* are not used any more in SIMIT 7. Please remove these keywords in your behaviour description.

3.2.1 Data type *bool*

In SIMIT 7 the data type *binary* (or *bool*) can only have values *False* and *True.* Binary variables cannot be assigned numbers. Also, comparison of binary values is possible to other binary values only, not to numbers.

For example, instead of

```
if (intVar1 & intVar2)
```

you now need to write

```
if ((intVar1 & intVar2) != 0)
```

since `intVar1 & intVar2` yields a number, not a boolean value.

During migration of a component type simple expressions and assignments will be recognized and converted automatically as shown in Table 3-2 and Table 3-3.

Modifications necessary within more complex expressions will be displayed as a syntax error in the CTE which is to be corrected manually.

Copyright © Siemens AG, 2013

SIMIT 5	SIMIT 7

`boolVar = 1;`
`boolVar = 0;`
	`boolVar = True;`
	`boolVar = False;`

`logical  boolVar = 1;`
`logical  boolVar = 0;`
	`binary boolVar = True;`
	`binary boolVar = False;`

`boolVar == 1`
`boolVar == 0`
	`boolVar == True`
	`boolVar == False`

Table 3-2: Replacements within the command-oriented behaviour description

SIMIT 5	SIMIT 7

`boolVar = 1;`
`boolVar = 0;`
 	
`boolVar = { ... : 0; ELSE 1};`
	`boolVar = True;`
	`boolVar = False;`
	
	`boolVar = { ... : False; ELSE True };`

`logical boolVar = 1;`
`logical boolVar = 0;`
	`binary boolVar = True;`
	`binary boolVar = False;`

`boolVar = 1`
`boolVar = 0`
	`boolVar = True`
	`boolVar = False`

Table 3-3: Replacements within the equation-oriented behaviour description

3.2.2 Functions for text processing

SIMIT 5 does not provide a data type for text. Here you need to use arrays of type *char* and use functions for text processing. SIMIT 7 provides a data type *text* (or equally *string*).

You may use the “+“-operator to concatenate strings, there are no special functions required any more.

The data type *char* still exists in SIMIT 7. However, it can be used for character variables only, not for numerical values any more. Violations will be reported and need to be fixed manually.

3.2.3 String constants

In SIMIT 7 octal escape sequences (\123) are not allowed within string constants. You need to modify such string constants manually.

3.2.4 The *switch* statement

In SIMIT 7 all *case*-blocks need to be closed with a *break*-statement unless the case is empty. Violations will be reported and need to be fixed manually.

Copyright © Siemens AG, 2013

3.3 Operating window and *appearance*-variable

In SIMIT 5 you may use the system variable *appearance* to make a certain variant of a components operating view visible in the simulation. In SIMIT 7 there exists one single link view only. During migration all variants of a components operating view will be converted into images and will be used as animation of a rectangle within the migrated component type.

This animation will be controlled by a variable that is also called *appearance* but which is not a system variable any more. It needs to be created as a standard output of type *integer*. It is advisable to declare this output “visible only in CTE”.

Please note that in SIMIT 5 *appearance* was a variable of type double so you may need to change assignment expressions.

⚠ **CAUTION**

When text elements were used in an operating view in SIMIT 5 automatic migration will not work, since static images will be created within which text cannot be replaced later.

3.3.1 Name identifiers _*KKS1* and _*KKS2*

In SIMIT 5 a component is identified by its name and suffix. Within the component type you may access name and suffix using the variables _*KKS1* and _*KKS2* respectively.

In SIMIT 7 there is just a name which is accessed using the variable _*NAME*. There is no automatic migration, you need to manually adapt access to _*KKS1* and _*KKS2.*

3.4 Complex connection types

Complex connection types that exceed a certain number of signals cannot be automatically migrated from SIMIT 5. In the migrated component type you will notice the corresponding connections to be missing. In this case in the migrated component type you need to newly create both the connection type and the connections that make use of this connection type.

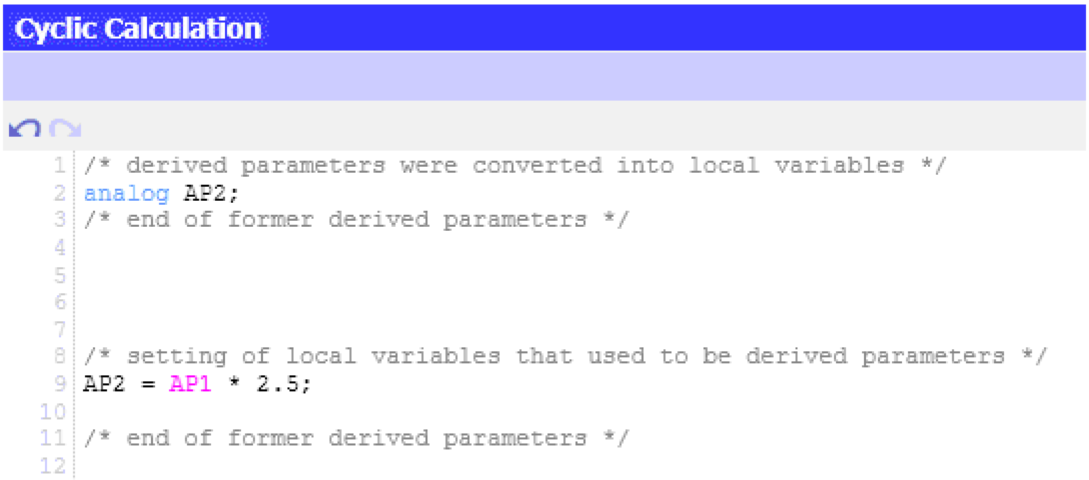
3.5 Enumerations

Enumeration types will not be automatically migrated from SIMIT 5. In the migrated component you need to newly create both the enumeration type and the parameters that make use of this enumeration type.

In SIMIT 7 an enumerations element is accessed by its name only. Also, the enumerations name must preceed the elements name, using a dot as separator, the whole expression needs to be written in single quotes: *‘enumeration.element’*.

3.6 Derived parameters

SIMIT 7 does not provide “derived parameters“ anymore. During migration there will be a local variable of same name introduced automatically within the cyclic behaviour description as shown in Figure 3-2. The parameters computation will be migrated.



```
/* derived parameters were converted into local variables */
analog AP2;
/* end of former derived parameters */




/* setting of local variables that used to be derived parameters */
AP2 = AP1 * 2.5;

/* end of former derived parameters */

```

Figure 3-2: Example for migration of a “derived parameter“

In case you make use of such parameters in the initialization also, you need to copy the generated code parts into the initialization section.

3.7 Widget “Image display“

SIMIT 5 allows the widget “Image Display“ to be used in a components operating window. In SIMIT 7 use of image sequence and image toggle animations is planned instead of a control. Currently, animations cannot be used within a components operating window, however.

3.8 Dynamic widgets

Until SIMIT 5.1 there were “dynamic“ widgets which worked with rising and trailing edges. In versions to follow up until 5.4 these widgets could not be used in new components any more, old components remained functional, however.

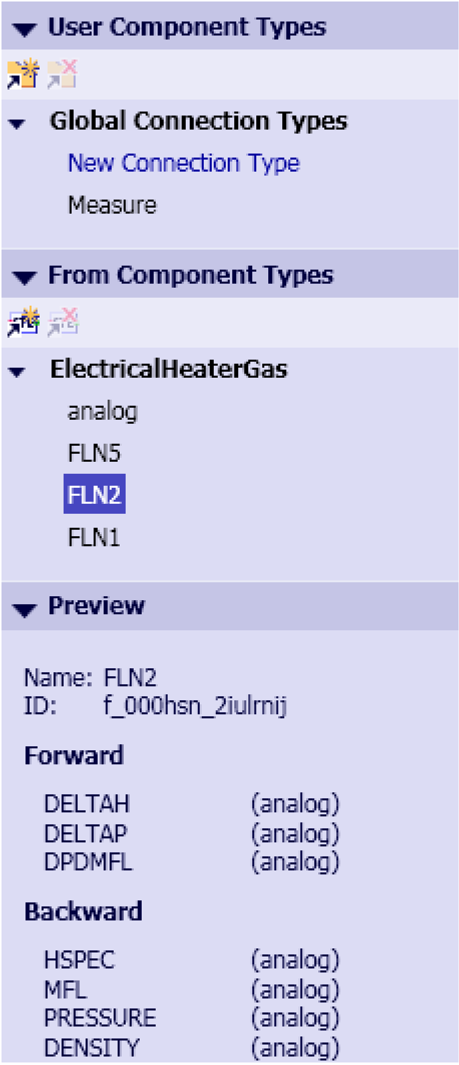
In SIMIT 7 there are no controls that work based on edges anymore. Hence dynamic controls will be mapped to existing controls during migration without assignment of signals. You need to manually rebuild the working of an edge-controlled widget within the migrated component type.

3.9 Using migrated types in the global section

In case you want to make use of connection types or enumeration types from a migrated component for later use in the CTE, just copy the types of interest into the global section.

Copyright © Siemens AG, 2013

To do so, just open a component that makes use of the desired type in the taskcard *connection types* or *enumeration types* (s. Figure 3-3) using the command and copy the types into the section *Global connection types* or *Global enumeration types*.



User Component Types

Global Connection Types

New Connection Type

Measure

From Component Types

ElectricalHeaterGas

analog

FLN5

FLN2

FLN1

Preview

Name: FLN2
ID: f_000hsn_2iulrnij

Forward

DELTAH	(analog)
DELTAP	(analog)
DPDMFL	(analog)

Backward

HSPEC	(analog)
MFL	(analog)
PRESSURE	(analog)
DENSITY	(analog)

Figure 3-3: Task-Card *Connection Types*

3.10 Assigning migrated components for project import

When migrating projects from SIMIT 5, user-defined components can be used provided they were migrated in advance.

Since there is no unique identification number (ID) in SIMIT 5, the assignment must be solely done using the components name and folder hierarchy. Please put your migrated components into the section “User Components” including the same folder hierarchy as in SIMIT 5 and do not modify the components name. The root folder must be named *user*, since in SIMIT 5 this was the only location to store user-defined components.

Copyright © Siemens AG, 2013

CAUTION

As the components version is of no importance here, please make sure that SIMIT cannot find several components of the same name in one folder because in this case you cannot tell which of these components will be used.

4 MIGRATING DIAGRAMS AND OPERATING WINDOWS

In SIMIT 7 there is no distinction between diagrams and operating windows any more. Hence both diagrams and operating windows from SIMIT 5 are converted into diagrams in SIMIT 7. Migration is done using project migration as described in chapter 2.2.

In the following some specifics concerning migration of diagrams and operating windows are explained.

CAUTION

⚠ After migrating a project always open the consistency check in SIMIT 7. All resulting warnings and errors require reworking of the migrated project!

4.1 Migrating diagrams with sheets

In SIMIT 5 a diagram may consist of several sheets. SIMIT 7 does not split a diagram in sheets any more, hence a diagram will be converted into one new diagram for each of its sheets. The name will be the diagrams name and a consecutive number indicating the sheet. A diagram called *Diagram* with 3 sheets will be converted into three diagrams *Diagram*, *Diagram2* and *Diagram3*.

4.2 Migrating component names

Component names will not be changed during migration. In case this leads to non-unique component names in the migrated project, the consistency check will show a corresponding error message.

In SIMIT 5 a component name contains the name itself and an extension. In SIMIT 7 this extension has been omitted, the scheme for converting component names is shown in Table 4-1.

SIMIT 5	SIMIT 7
Name, no Extension	Name
Name and Extension	Name/Extension
No Name, Extension	/Extension

Table 4-1: Migrating component names

In SIMIT 7 unique component names are mandatory. The consistency check will reveal any ambivalence.

Copyright © Siemens AG, 2013

4.3 Migrating signal names

When defining a signal to connect to a component and/or widget in SIMIT 5, component name and connection point name are concatenated using a slash '/', i.e. *component name/connection point name.* In SIMIT 7 there is always a signals source and a signals name to be provided. The source of a signal is the name of a component, a control or a gateway. The name of a signal is a connection points name within a component or a signals name within a gateway.

During migration the name of a signal in SIMIT 5 is split in such a way that the part from the beginning to the last slash is regarded as the signals source, the part from the last slash to the end is regarded as the signals name.

4.4 Migrating data types

The data types *logical* and *double* in SIMIT 5 correspond to *binary* and *analog* in SIMIT 7. In addition to this, SIMIT 7 provides an *integer* data type. Some components in the SIMIT standard library do not have *analog* connection points anymore, but *integer*. This holds true for all converter components in the *Conv* section and for *Multiplexer* and *Simulation Time* in the standard library.

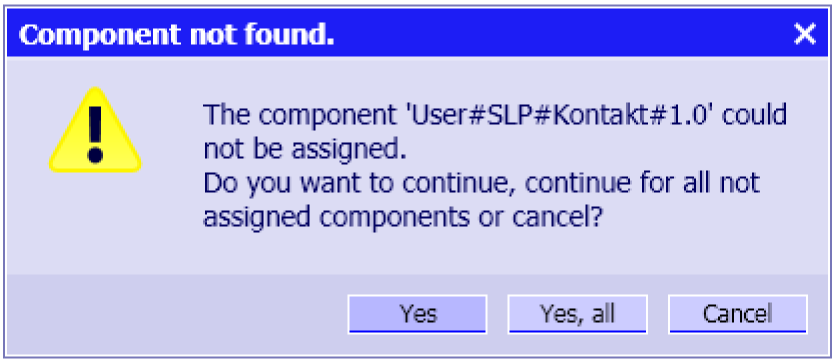
All components that are connected to state- or controlwords now have connection points of *integer* type. This holds true for drive components *PROFIDrive* and *SIMOCODEpro*. When these connection points have been connected to input- and output connectors directly, migration will be consistent since the corresponding signals in the gateway are now of *integer* type, too. When connection points of a drive component are not connected to input- and output connectors directly, inconsistencies may occur.

If migration would lead to connections of incompatible data types between components, these connections will be removed. In case of gateway signals that have data type *integer* in SIMIT 7, please proceed as shown in chapter 5.2.

4.5 Specifics when migrating components

During migration components are mapped from types known in SIMIT 5 to types known in SIMIT 7. This mapping is provided for the SIMIT standard library, including some specifics as shown in the sections to follow.

In case your SIMIT 5 project contains user components that have no mapping in SIMIT 7, an error dialog will show (see Figure 4-1).



Component not found.

The component 'User#SLP#Kontakt#1.0' could not be assigned.
Do you want to continue, continue for all not assigned components or cancel?

Figure 4-1: No component mapping during migration

You may abort migration or continue and ignore this or all similar cases.

4.5.1 Global connectors

SIMIT 7 does not distinguish between global input and output connectors, there is just one global *Connector*. During migration both input and output connectors are mapped to this connector with appropriate connection points.

4.5.2 Diagram connectors

In SIMIT 7 both *Diagram Input* and *Diagram Output* connectors known in SIMIT 5 have been removed and will be replaced by the global connector. The former parametrization of the diagram connectors using *Connector Name* and *Diagram Name* will be converted as shown in Table 4-2, with *Location* depicting the diagram the connector is located in.

SIMIT 5	**SIMIT 7 (Name of global connector)**
Diagram Output Connector Name: “Connector” Diagram Name: “Diagram”	“Connector/Diagram/Location”
Diagram Input Connector Name: “Connector“ Diagram Name: “Diagram”	“Connector/Location/Diagram”

Table 4-2: Migrating diagram connectors

4.5.3 Characteristic

The characteristic component will be migrated without its control points.

To transfer control points, please use SIMIT 5 to export the control points of a certain characteristic component in form of a .csv file, convert the .csv file into a tab-separated .txt file and import the .txt file into the corresponding characteristic component in SIMIT 7 after migration.

4.5.4 Components in section *Misc*

The standard library in SIMIT 5 provides some component types in the *Misc* section that allow for widgets in SIMIT 5 to be functionally used in a diagram.

Since SIMIT 7 allows controls to be connected directly, these components are not necessary any more, hence they are not converted. This holds true for component types *Button*, *Display*, *Indicator*, *Input*, *Pushbutton, Slider* und *Stepper*. In the migrated project just place an appropriate control in your diagram for any of these components, if necessary.

A similar matter holds true for component types *ASwitch* and *BSwitch*. Figure 4-2a shows an example of how an ASwitch is used. Since SIMIT 7 allows a signal connection to be broken at any place using the *Signal Disconnector* control, these component types are not necessary any more, hence are not contained in the standard library.

Copyright © Siemens AG, 2013

In order not to limit functionality, *ASwitch* and *BSwitch* will be replaced by components that just copy the input signal to their output (Figure 4-2b). If a manual setting of the value is required in the migrated project, just place additional controls as shown in Figure 4-2c.

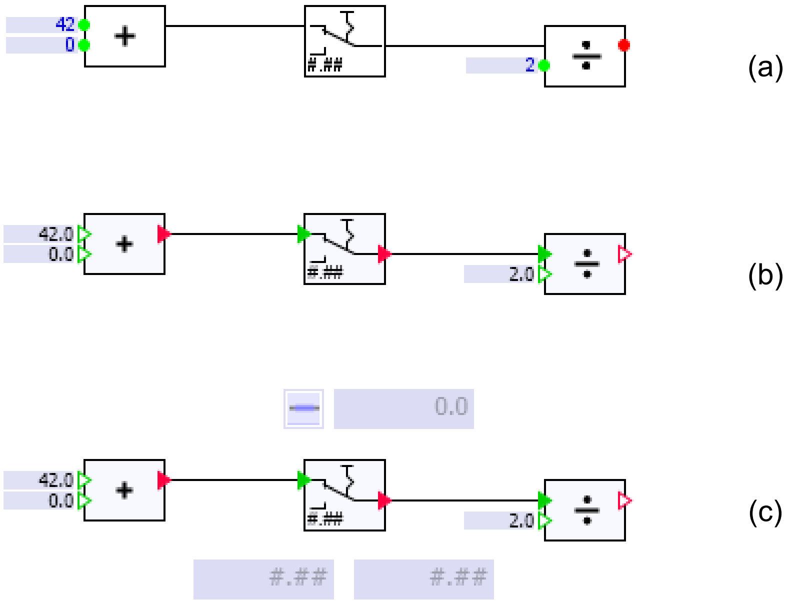


Figure 4-2: Migrating *ASwitch*

4.6 Migrating graphics and animation

When migrating a project, graphics will be converted as far as possible, including its parameters. Animations need to be newly created for any migrated graphics object.

4.6.1 Migrating rectangles and ellipses

Rectangles and ellipses are converted along with their properties. Differences will show only if a rectangles edges in SIMIT 5 have a radius, which is not supported in SIMIT 7. So there will be no rounded edges in a SIMIT 7 rectangle.

NOTE

Please note that in SIMIT 5 a rectangle or ellipse is automatically converted into a (Bezier-)curve when rotated. Hence a rotated rectangle will be a curve after migration.

4.6.2 Migrating lines and curves

Lines and curves are converted along with all their properties.

Copyright © Siemens AG, 2013

4.6.3 Migrating text

Text will be migrated including its font type and size. Distortion, i.e. horizontal and vertical scaling, will not be migrated. Figure 4-3 shows the result: On the left hand side you see the original text, the right hand side shows the migration result.

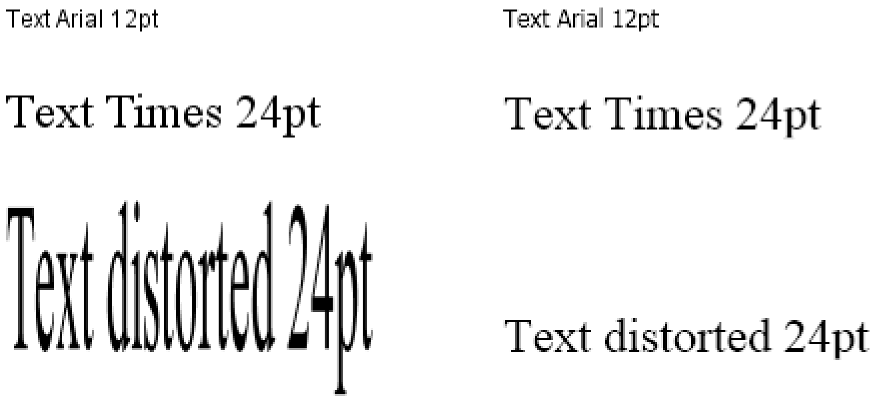


Figure 4-3: Migrating Text

NOTE

Text that was rotated by an angle that is not a multiple of 90° in SIMIT 5 will not be shown correctly in the migrated project.

4.6.4 Operating symbol of a component

Since SIMIT 7 does not distinguish between diagrams and operating windows any more, there is only one single view of a component as shown on the diagram. Hence in SIMIT 7 there is no correspondence to a components operating symbol in SIMIT 5 any more, so it will not be migrated. Migrated operating windows will not show these symbols.

4.6.5 Migrating animations

SIMIT 7 allows you to animate a graphic object in a much more intuitive way by just drawing the animation action. Within grouped objects any individual element can now have its own animation, which considerably enhances animation abilities.

Animation defined in a SIMIT 5 project will not be migrated to SIMIT 7, so if required it has to be newly created.

4.7 Migrating widgets

Widgets as known in SIMIT 5 will be replaced by controls as known in SIMIT 7 according to Table 4-3. Despite their different look and feel a widgets size will be transferred to a controls size to a close match. A controls abilities to be parametrized depends on its predefined color scheme and differs from a widgets parameters, parameters will be migrated as far as possible.

Widget in SIMIT 5	**Control in SIMIT 7**
Bar display	Bar Indicator
Digital display	Digital Display
Binary display	Binary Display
Speedometer	Analog Display
Image display	Animation (see below)
Digital input	Digital Input
Digital input with override	Digital Input and Signal Disconnector
Slider	Slider
Pushbutton	Pushbutton
Switch	Switch
Button with override	Switch and Signal Disconnector
Image button	Pushbutton with Image or Stepping Switch with Image
Image pushbutton	Pushbutton with Image

Table 4-3: Migrating Widgets

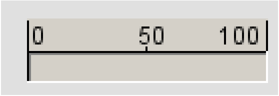
CAUTION

Your SIMIT 5 project may contain widgets that were connected to components that were not migrated, hence are not contained in the migrated project. Consequently the signal the migrated control is associated with does not exist any more. The consistency check will report this as a *Control with unknown signal*.

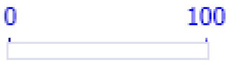
In the following you find widgets and their corresponding controls compared and also specifics when migrating those widgets.

4.7.1 Bar Display

Widget in SIMIT 5



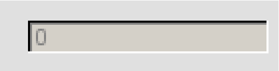
Control in SIMIT 7



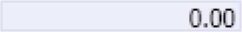
Copyright © Siemens AG, 2013

4.7.2 Digital Display

Widget in SIMIT 5



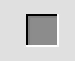
Control in SIMIT 7



You cannot provide two signals anymore from which one single value was computed. Only the *Display Signal (low byte)* will be used. If required, you may use standard components to compute one value from two signals.

4.7.3 Binary Display

Widget in SIMIT 5

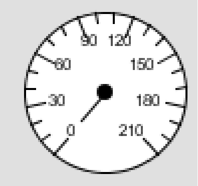


Control in SIMIT 7

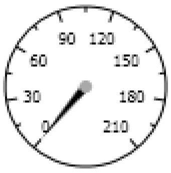
If required, the tridimensional appearance from SIMIT 5 can be recreated using animation. Just replace the control in the migrated project with an *Image Alternation* animation.

4.7.4 Speedometer

Widget in SIMIT 5



Control in SIMIT 7



4.7.5 Image Display

There is no control in SIMIT 7 to match the *Image Display* widget in SIMIT 5. During migration the *Image Display* will be replaced by an animation without loss of functionality. An *Image Display* that is connected to a binary signal will be replaced by an *Image Alternation* animation. An Image Display that is connected to an analog signal will be replaced by an *Image Sequence* animation.

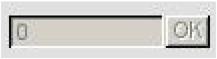
NOTE

Please note that you require the additional module DGE (Dynamic Graphics Editor) to edit graphics and animation.

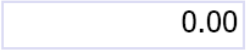
Copyright © Siemens AG, 2013

4.7.6 Digital Input

Widget in SIMIT 5

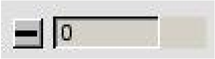


Control in SIMIT 7

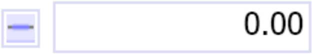


4.7.7 Digital Input with Override

Widget in SIMIT 5

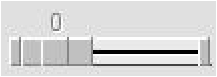


Controls in SIMIT 7

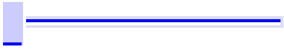


4.7.8 Slider

Widget in SIMIT 5



Control in SIMIT 7



The numerical value will not be displayed on top of the slider any more. If required, use the *Show Value* property in the *View* section to display the value.

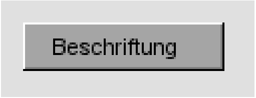
CAUTION

When moving the slider in SIMIT 5, a new value will not be effective unless the slider is released. In SIMIT 7 intermediate values will be effective while the slider is moved. Please verify that this behavior is tolerable.

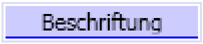
To imitate the behavior of a slider in SIMIT 5 just proceed like this: Activate the *Show Value* property within the sliders *View* properties. An additional field will be displayed that shows the sliders current value. When the simulation is running you may enter a value directly in this field which will be effective immediately.

4.7.9 Pushbutton

Widget in SIMIT 5



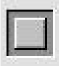
Control in SIMIT 7



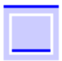
Copyright © Siemens AG, 2013

4.7.10 Switch

Widget in SIMIT 5



Control in SIMIT 7



The *Switch* in SIMT 7 shows locking behavior, hence there is no need to show its current state using two different colors. So the color settings from SIMIT 5 have no match in the control as used in SIMIT 7.

If you need to copy the color change, just use a setup as shown in Figure 4-4. You can find a sample SIMIT 7 project on the SIMIT software CD in the folder *Sample Projects\Migration* that contains this setup as a macro component. If required, you can copy this macro component to your migrated SIMIT 7 projects.

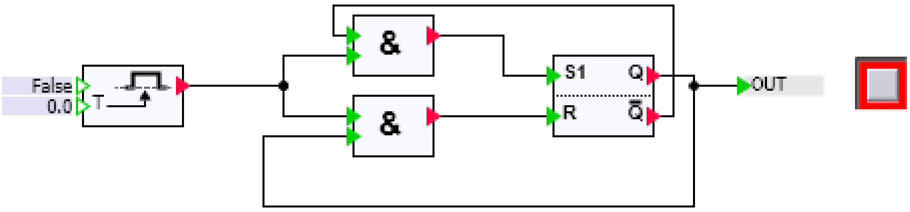
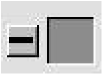


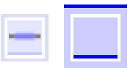
Figure 4-4: Setup to show color change

4.7.11 Button with override

Widget in SIMIT 5

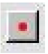


Controls in SIMIT 7



4.7.12 Switch

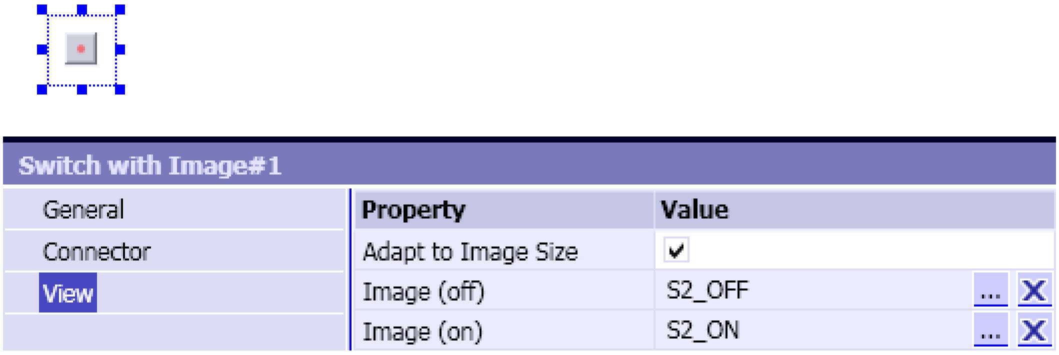
Widget in SIMIT 5



Control in SIMIT 7

This widget will be replaced by a downscaled *Switch* control.

If you need to exactly match the look from SIMIT 5, just use a *Switch with Image* as shown in Figure 4-5. The SIMIT 7 project on the SIMIT software CD in the folder *Sample Projects\Migration* provides this control for use in your migrated SIMIT 7 projects.



Switch with Image#1

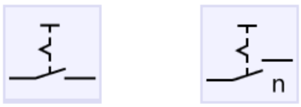
	Property	Value
General	Adapt to Image Size	☑
Connector	Image (off)	S2_OFF
View	Image (on)	S2_ON

Figure 4-5: Switch with Image

4.7.13 Image Button

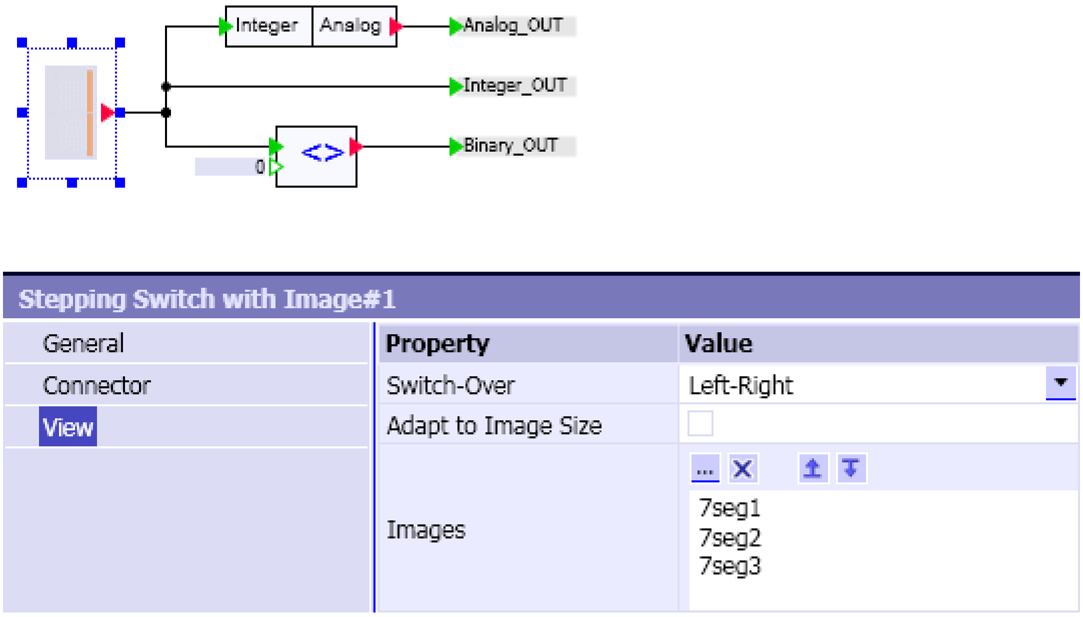
Widget in SIMIT 5

Control in SIMIT 7



When connected to a binary signal, the *Switch with Image* control will be used. When connected to an analog signal, the *Stepping Switch with Image* control will be used.

If both an analog and a binary signal is used there is no matching control in SIMIT 7 available. If required, just use a setup as shown in Figure 4-6 to build this behavior. You will find this setup also in the sample project *Migration.simarc* on the SIMIT software CD.



Stepping Switch with Image#1

	Property	Value
General	Switch-Over	Left-Right
Connector	Adapt to Image Size	□
View	Images	7seg1 7seg2 7seg3

Figure 4-6: Stepping switch with different variable types

The sensitive area for toggling values via mouse click are predefined, they are however parametrizable for the controls in SIMIT 7 as shown in Figure 4-7.

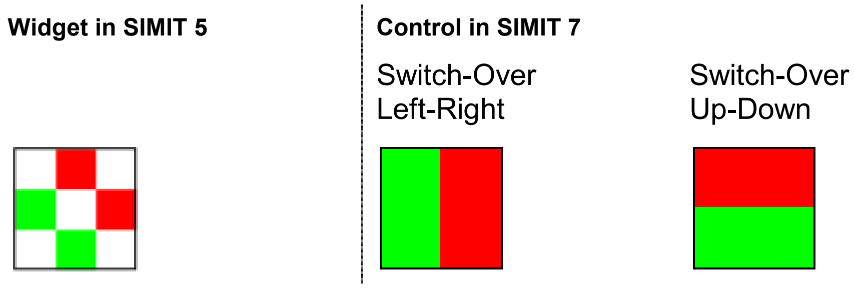


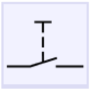
Figure 4-7: Sensitive areas for toggling values

Please check if the sensitive areas match your demands and adjust them if necessary.

4.7.14 Pushbutton with Image

Widget in SIMIT 5

Control in SIMIT 7



4.8 Rule based implicit connections

SIMIT 5 projects allow for implicit connections to be established based on rules which are defined in configuration files. This strategy does not exist in SIMIT 7. If required, just use the easy-to-use way in SIMIT 7 to newly establish implicit connections.

5 MIGRATING GATEWAYS

Gateways are not migrated together with the project but need to be newly created in SIMIT 7. Just establish new gateways in your SIMIT 7 project and check their signals for inconsistencies as shown in the following.

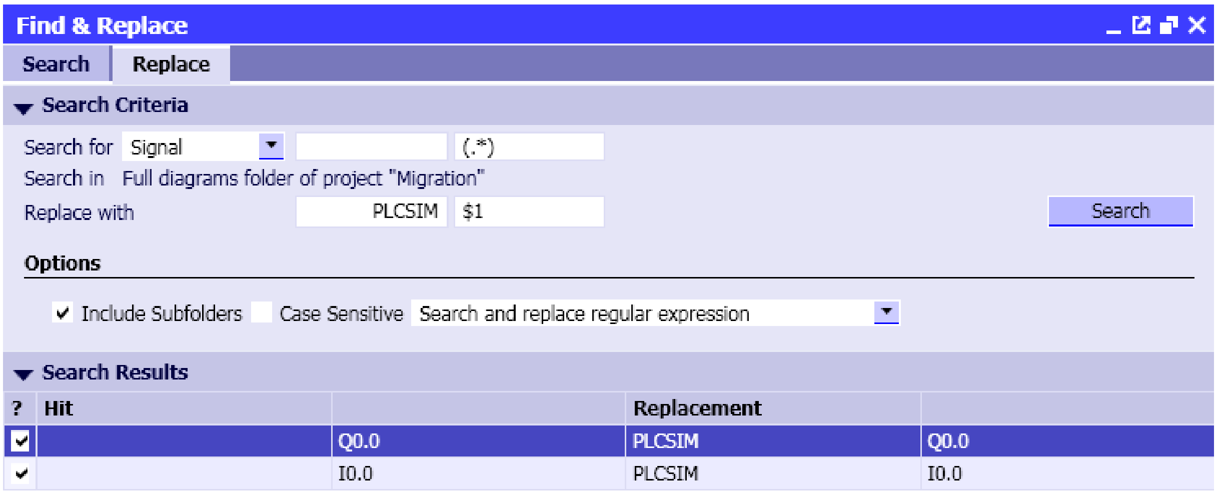
5.1 Migrating the names of gateway signals

In SIMIT 5 a gateway signal has both its symbolic name and its absolute address as signal name. In SIMIT 7 any gateway signal is known by its source and name. The gateways name defines the signal source, the symbolic name or the absolute address define the signals name.

5.1.1 Adding the souce

When adding a gateway in SIMIT 7 any gateway signal will automatically know the gateway to be its source. When migrating diagrams and operating windows the sources name will be missing in the resulting diagrams. The *Search and Replace* feature allows you to easily add the sources name: Use *Search for Signal* and choose the *Search and replace regular expression* option.

If you did not provide a prefix for the SIMIT 5 gateway, use settings as shown in Figure 5-1. If you did provide a prefix, use settings as shown in Figure 5-2. In both cases *PLCSIM* was used as both the gateways name and the prefix.



Search for: Signal | | (.*)

Search in: Full diagrams folder of project "Migration"

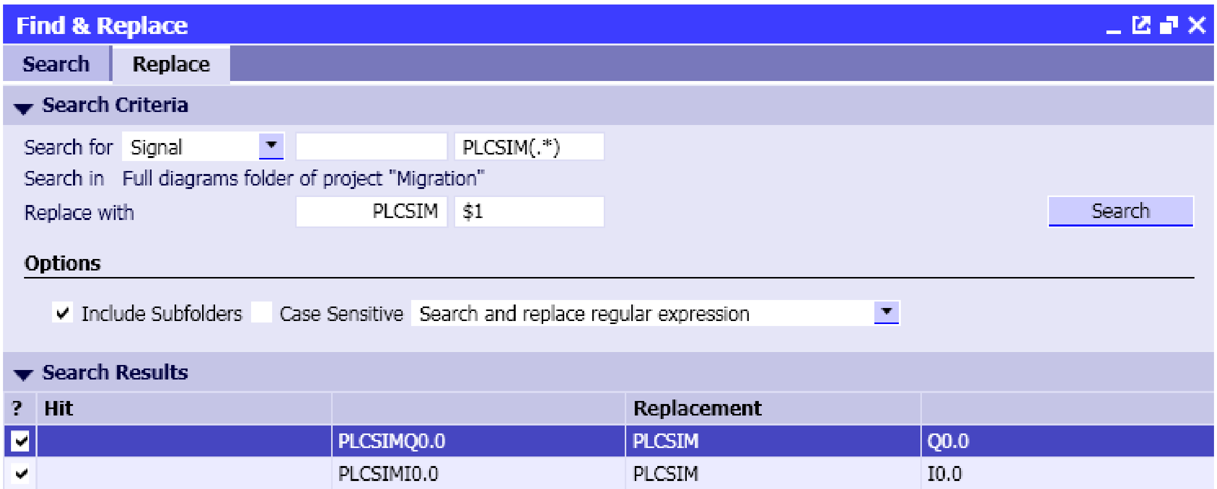
Replace with: PLCSIM | $1

☑ Include Subfolders ☐ Case Sensitive Search and replace regular expression

?	Hit		Replacement	
☑		Q0.0	PLCSIM	Q0.0
✓		I0.0	PLCSIM	I0.0

Figure 5-1: Replacement if no prefix was provided

Copyright © Siemens AG, 2013



Search for: Signal | | PLCSIM(.*)

Search in: Full diagrams folder of project "Migration"

Replace with: PLCSIM | $1

☑ Include Subfolders ☐ Case Sensitive | Search and replace regular expression

?	Hit		Replacement	
☑		PLCSIMQ0.0	PLCSIM	Q0.0
✓		PLCSIMI0.0	PLCSIM	I0.0

Figure 5-2: Replacement if prefix *PLCSIM* was provided

5.1.2 Symbolic and absolute address

In SIMIT 7 a gateway signal that was assigned a symbolic name can be accessed using this symbolic name only. If in SIMIT 5 you used the absolute address to access the signal although the signal was provided a symbolic name, you need to either delete the symbolic name in the gateway or change signal access to use the symbolic name.

CAUTION

If an input- or output connector in SIMIT 7 is assigned a signal that cannot be found, SIMIT 7 will produce a warning message, not an error. Hence simulation can be started anyway.

To avoid problems, manually call the *Consistency Check* once after migration and check warnings.

5.2 Migrating data types of gateway signals

SIMIT 5 treated all non-binary signals as analog signals which were mapped to the data type *double*. SIMIT 7 will classify all non-binary signals according to their type in SIMATIC. Table 5-1 shows the mapping to an analog or integer data type.

Signaltyp in SIMATIC	Datentyp in SIMIT 7
BOOL	Binary
BYTE	Integer
WORD	Integer, if there is **no** standardization specified for this signal Analog, if there is a standardization specified for this signal
INT	Integer
DWORD	Integer
DINT	Integer
REAL	Analog

Table 5-1: Data types of gateway signals

This enhancement will show effects in a migrated project for gateway signals that were mapped to *Integer* type. On diagrams resulting from migrated diagrams and operating windows these gateway signals will usually be connected to analog inputs and outputs of components and controls.

The *Consistency Check* will show such signals as *Connector with signal of wrong type.* Please remove these inconsistencies by using an appropriate converter component, e.g. by using an *Analog2Integer* component for integer inputs or an *Integer2Analog* component for integer outputs.

5.3 Migrating data record communication

In SIMIT 7 communication of data records in the Profibus DP gateway has become more flexible and easier to use. If in SIMIT 5 data record signals were created to communicate data records to a component, SIMIT 7 provides the special *Unit* connector to establish a connection between a component and a gateway.

This only affects component types *SIWAREXU1* and *SIWAREXU2*. For detail with respect to converting data record communication for these components please see these components online help or the Standard Library Manual.

Copyright © Siemens AG, 2013

6 MIGRATING MACRO COMPONENTS

To migrate macro components from SIMIT 5 to SIMIT 7 just convert them into diagrams, convert a project containing these diagrams and convert the resulting diagrams back into macro components.

In order to correctly migrate projects that use macro components you first need to migrate those macro components themselves to SIMIT 7 and place them in the *Global Macros* section. These macro components need to be arranged in a hierarchy that exactly matches the folder structure as it was used in SIMIT 5.

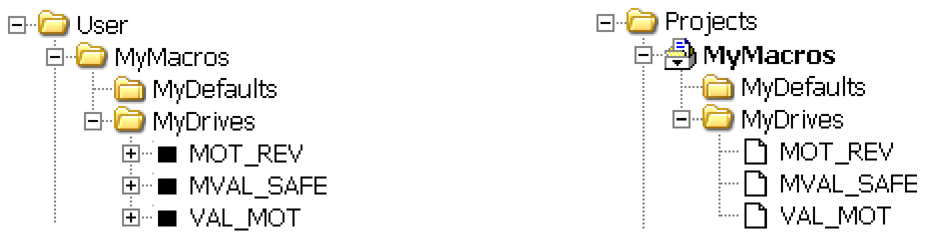
6.1 Migrating macro components step-by-step

Please follow these steps to migrate macro components:

Step 1:

Create a new project in SIMIT 5 and create a new diagram for each macro component.

If you placed your macro components in the SIMIT 5 *User* section using a directory structure, use the same directory structure in your project and assign each diagram the name the corresponding macro has as shown in Figure 6-1.



- User
  - MyMacros
    - MyDefaults
    - MyDrives
      - MOT_REV
      - MVAL_SAFE
      - VAL_MOT

- Projects
  - **MyMacros**
    - MyDefaults
    - MyDrives
      - MOT_REV
      - MVAL_SAFE
      - VAL_MOT

Macro components
in the library

Project with diagrams
for the migration
of the macro components

Figure 6-1: Project used to migrate macro components

Step 2:

From each macro component copy its content onto the corresponding diagram. Observe that macro connectors and their connections will vanish. Figure 6-2 shows the result for the VAL_MOT macro.

Copyright © Siemens AG, 2013

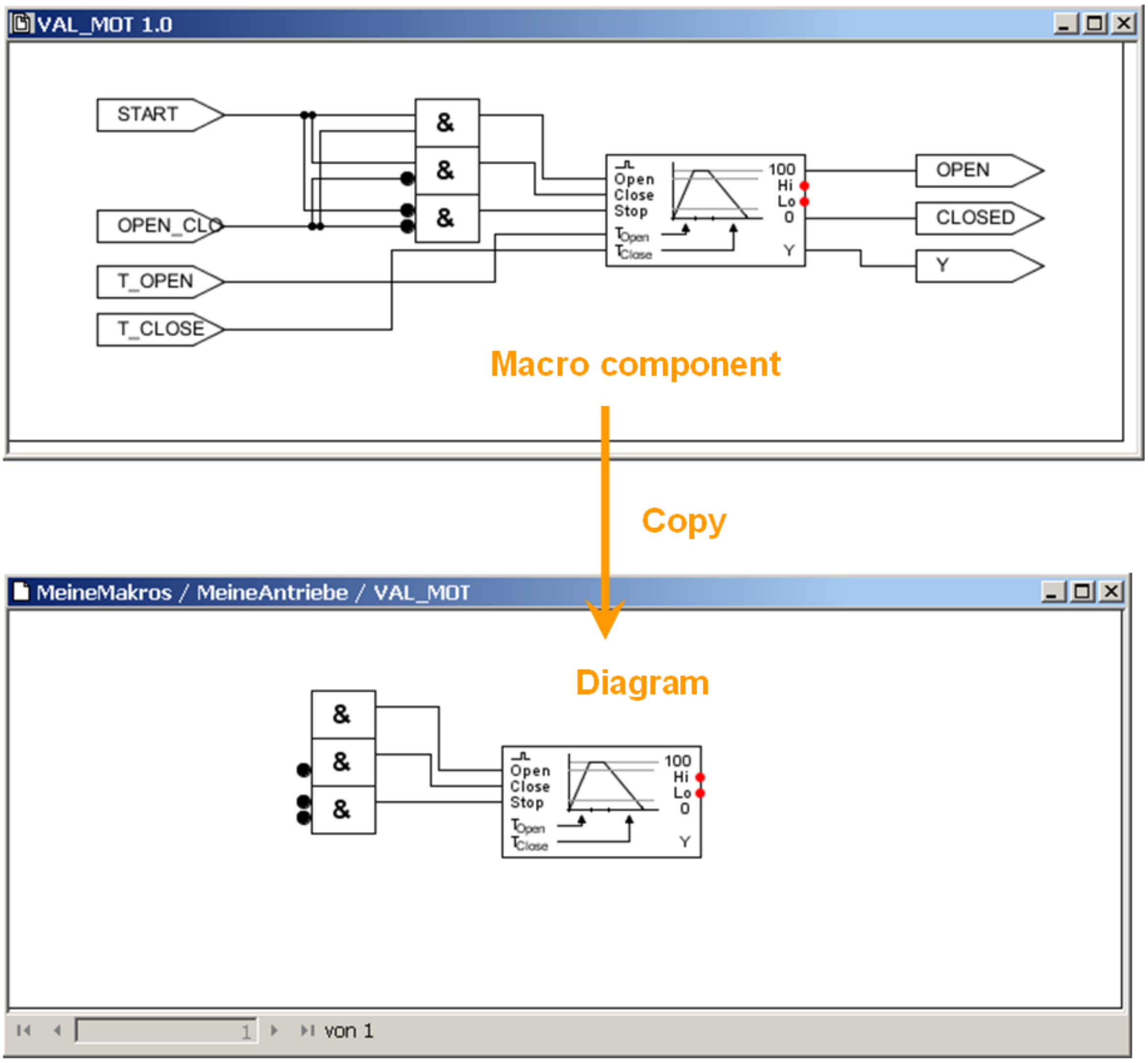


Figure 6-2: Macro component as diagram in SIMIT 5

Step 3:

Export the SIMIT 5 project.

Step 4:

Migrate the project in SIMIT 7 using the menu *Project* | *migrate* (see section 2.2). The SIMIT 7 project now contains the migrated diagrams. Figure 6-3 shows the resulting diagram for the VAL_MOT macro.

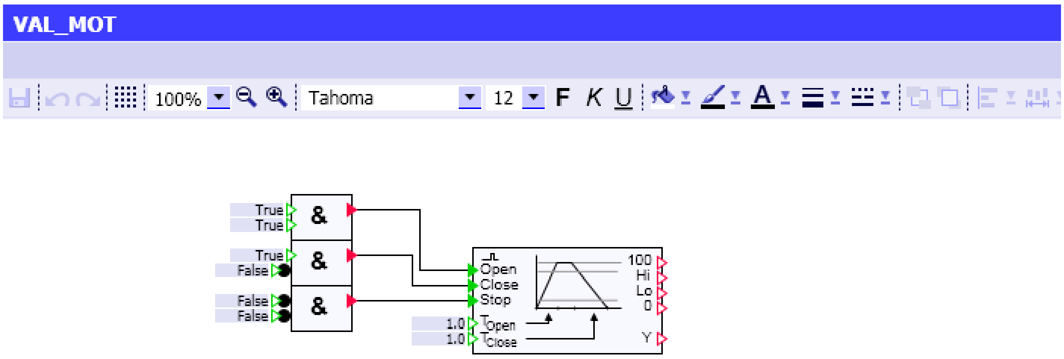


Figure 6-3: Macro component as diagram in SIMIT 7

Step 5:

Create a new editable macro for each diagram using the same name as in SIMIT 5 and copy the content from the diagram into the macro you are just editing.

Step 6:

Finish each macro component by defining its connection points within the input and output section and by establishing connections as in the SIMIT 5 macro component. Please use the same connection point names as in SIMIT 5.

NOTE

Please note that in SIMIT 7 connection names of a macro must not contain special characters such as the +-sign.

In case special characters have been used in the connection names of macro components in SIMIT5 this will lead to incompatibility when migrating projects that contain such macro components. In this case pease rework your macro components and projects that make use of them in SIMIT 5 prior to migration so that special characters are removed from connection names.

Please note that in SIMIT 7 the order in which connection points are definied in the input and output section also defines the order in which these connection points appear on the macros symbolic view while in SIMIT 5 connection points are shown in alphabetical order. Furhermore you can use separators in SIMIT 7, Figure 6-4 shows the resulting macro component VAL_MOT.

Copyright © Siemens AG, 2013

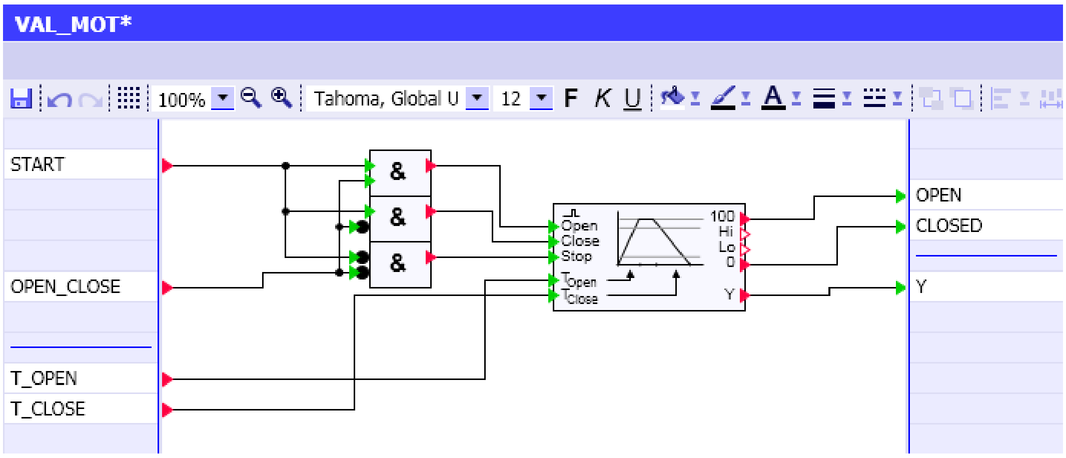
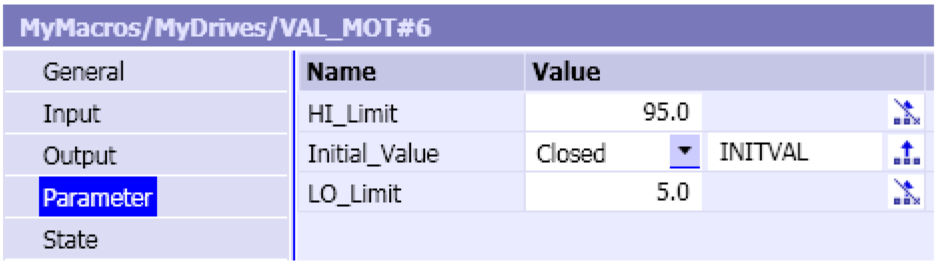


Figure 6-4: Macro component in SIMIT 7

Finally definie parameters for each macro component as required. Figure 6-5 shows the parameter *Initial_Value*. This parameter is definied as a macro components parameter named INITVAL.


MyMacros/MyDrives/VAL_MOT#6

- General
- Input
- Output
- Parameter
- State

Name	Value		
HI_Limit	95.0		
Initial_Value	Closed	INITVAL	
LO_Limit	5.0		

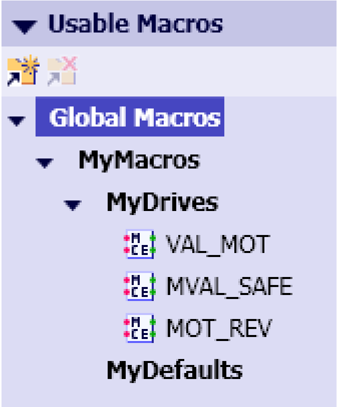
Figure 6-5: Defining a macro components parameter in SIMIT 7

In case you used inputs of components to parametrize a macro component in SIMIT 5 you need to use the *AConst*, *BConst* or *IConst* component to define these inputs as parameters. For details please see the Standard Library Manual.

Step 7:

Now create the same directory structure in *Global Macros* within the *Usable Macros* section of the task card *Macros* as used in the SIMIT 5 *User* area. Then drag and drop each macro component from the *Editable Macros* section into the appropriate folder in *Global Macros*. The result of migrating the example from Figure 6-1 is shown in Figure 6-6.

Copyright © Siemens AG, 2013



- Usable Macros
  - Global Macros
    - MyMacros
      - MyDrives
        - VAL_MOT
        - MVAL_SAFE
        - MOT_REV
      - MyDefaults

Figure 6-6: Usable macro components in SIMIT 7

Having migrated these macro components you can now migrate projects from SIMIT 5 that use these macro components.

6.2 Migrating diagrams with macro components

In SIMIT 5 macro components used on diagrams are identified using their library path, name and version. In order to correctly assign macro components when migrating diagrams in SIMIT 7, macro components in SIMIT 7 need to be located in the same directory structure as used in SIMIT 5. This procedure is shown in section 6.1. SIMIT 5 projects which contain macro components that are correctly migrated can then be migrated as shown in chapter 3.

After migration the diagram as shown in Figure 6-7 has become the diagram as shown in Figure 6-8.

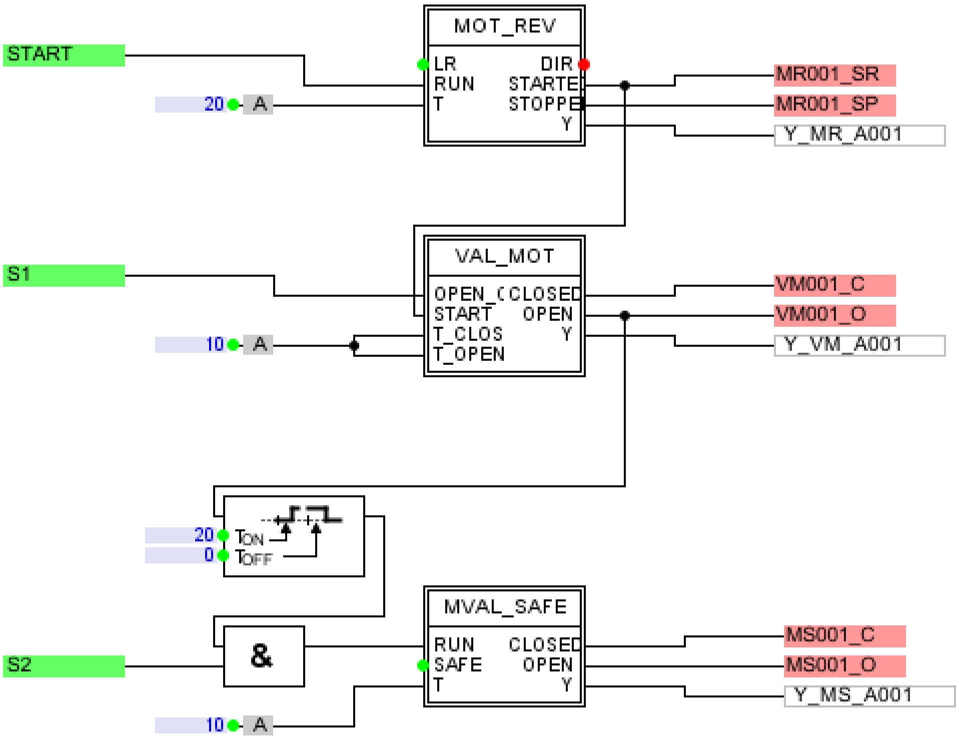


Figure 6-7: Diagram with macro components in SIMIT 5

Copyright © Siemens AG, 2013

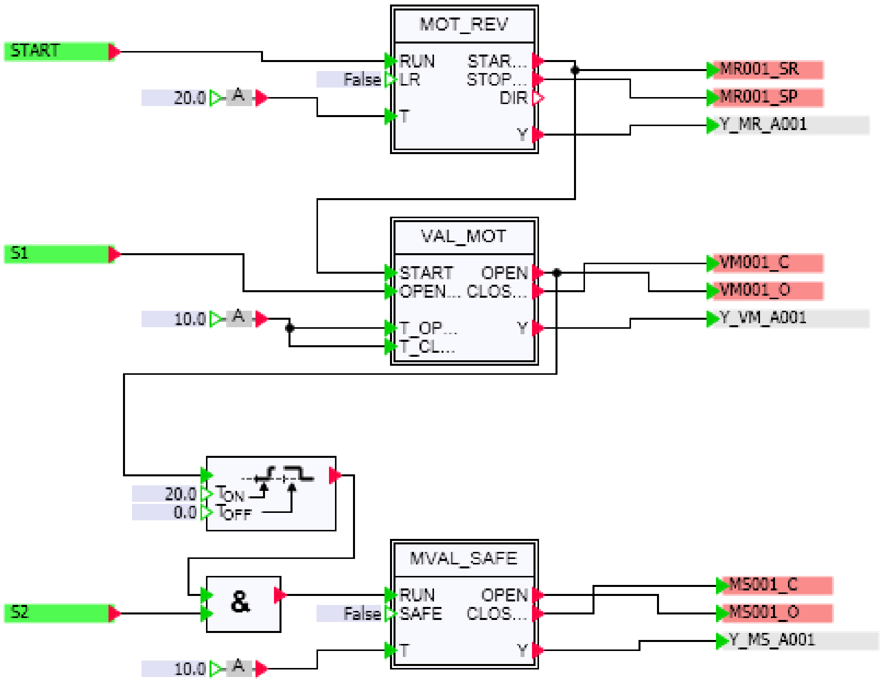


Figure 6-8: Migrated diagram with macro components in SIMIT 7

Each macro component in the SIMIT 5 diagram has been replaced with a corresponding macro component in SIMIT 7. Layout and width of the macro components will be copied.

6.3 Problems arising when migrating

Migrating diagrams with macro components assumes that the macro component and its connection points on the diagram can be identified in SIMIT 7. In the following potential problem cases are described.

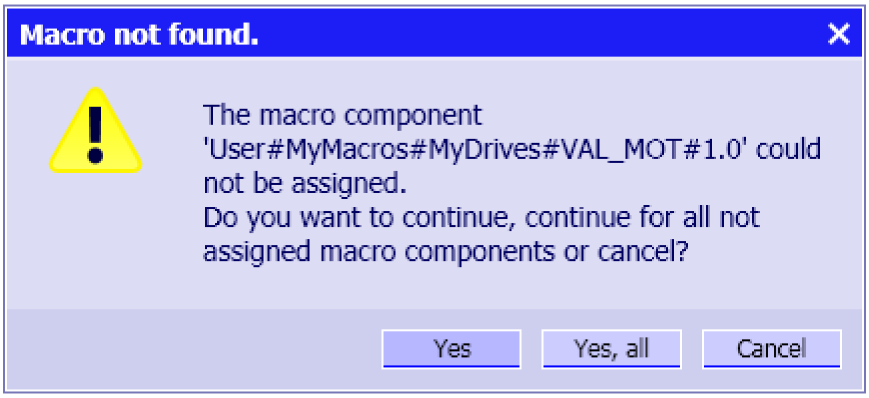
6.3.1 Mismatched connection name

If a macro components connection point in SIMIT 7 has not been assigned the same name as in SIMIT 5, no connections will be established for this connection point when migrating diagrams. Hence this connection point will show no connections in the resulting diagram.

6.3.2 Missing macro component

If a macro component in SIMIT 7 has been assigned a different name as in SIMIT 5 or if a macro component has not been placed in the same directory structure as in SIMIT 5, diagrams using this macro component cannot be completely migrated. The resulting diagrams will lack a macro component, i.e. the macro component and its connections are erased. This also is the case if a macro component used in a SIMIT 5 project is not at all present in SIMIT 7.

Upon the first occurence of this case a dialog will show as in Figure 6-9. You may abort migration or continue ignoring this case or all similar future cases.



The macro component 'User#MyMacros#MyDrives#VAL_MOT#1.0' could not be assigned.
Do you want to continue, continue for all not assigned macro components or cancel?

Figure 6-9: Error dialog when migrating diagrams with macro components

6.3.3 Macro versions in SIMIT 5

In SIMIT 7 a macro component has no version any more. Since macro components are matched based on their name when migrating a project, this mapping can only take place for one single version of a macro component.

This provides an automatic update of all versions of a macro to the migrated version.

In case you have used the macro components with the same name but different versions in your SIMIT 5 project and you want to maintain this distinction, you first need to make sure that these macro components differ either in name or in directory structure. To do so, just assign different names to the macro components in SIMIT 5 or place them into different folders within the library and then replace your macro components in all diagrams. Then migrate macro components and diagrams as usual.

Copyright © Siemens AG, 2013

7 MIGRATING TEMPLATES

Templates created with SIMIT 5 can be migrated to SIMIT 7 based on the migration of diagrams and operating windows.

7.1 Migrating templates step-by-step

To migrate SIMIT 5 templates to SIMIT 7 please follow these steps:

Step 1:

Create a project in SIMIT 5 and import all templates to be migrated into this project.

Step 2:

Export the SIMIT 5 project.

CAUTION

Please note that only SIMIT projects can be migrated that were exported with SIMIT 5.4 **SP1**!

Step 3:

Migrate the exported project into SIMIT 7 using menu *Project* | *migrate* (see chapter 2.2). The migrated project now contains the migrated diagrams and operating windows in form of diagrams.

You may need to rework these diagrams, for detail see chapter 3.

Step 4:

Open the Project Manager and drag the diagrams into the *User Templates* section in the *Templates* taskcard.

Step 5:

Now open each template in the template editor and define placeholders if required. In case the template used inputs or outputs for gateway signals or implicit connections were used, the migration did not provide a placeholder to the signal source. Please enter appropriate placeholders here.

Also please see specifics concerning migration of placeholders as described in chapter 7.2.2.

7.2 Specifics when migrating templates

Apart from specifics concerning migration of diagrams and operating windows you need to take into account additional issues when migrating templates as described in the following.

7.2.1 Adapting profiles

Instanciation of templates based on tables along with the profiles P1, P2 and P3 are available in SIMIT 7 also. Profiles are labled as in Table 7-1.

Profile in SIMIT 5	**Profile in SIMIT 7**
P1	Fixed placeholders V1', ,V2', ...
P2	Placeholders defined in 1st row
P3	Placeholders listed individually

Table 7-1: Names of profiles

Profiles P1ML, P2ML and P3ML are not available in SIMIT 7 any more. Tables using any of these profiles need to be converted into a format supported by SIMIT 7.

7.2.2 Placeholders for enumeration parameters

If a template has a parameter defined as a variable that is set by means of a enumeration, these parameters will be set to the first value in the enumeration in the migrated template. You need to newly define these parameters as placeholders.

7.2.3 Placeholders for numeric parameters

If a parameter does not allow a text value, placeholders entered in that parameter cannot be migrated. In this case the parameter will be set to its default value as defined in the component type, the placeholder needs to be newly assigned.

7.2.4 Placeholders for input defaults

Since an input default does not allow a text value, placeholders entered in that input default cannot be migrated. In this case the input will be set to its default value as defined in the component type, the placeholder needs to be newly assigned.

8 MIGRATING SCRIPTS

In SIMIT 7 the additional module ACI (Automatic Control Interface) has been fundamentally revised. Scripts that were created for use with SIMIT 5 need to be reworked for use with SIMIT 7.

Please note the following aspects:

- In SIMIT 7 there is no `RESET`-command any more.
- In SIMIT 7 there are no `ASSERT`- and `CHECKBUFFER`-commands any more.
- In SIMIT 7 the simulation is not logged any more and there is no more post-processing.
  As an alternative the scripting mechanism in SIMIT 7 provides the command`_printlog` which can be used by any component to directly write to the log-file of a running script. The component type needs to be adapted accordingly.
- Scripts in SIMIT 7 are type safe. As a consequence, some statements that were still accepted in SIMIT 5 are not legal any more in SIMIT 7:
  - A binary variable cannot be assigned a numerical value any more.
  - Comparison operators ’**>**’, ’**<**’, ’**>=**’, ’**<=**’ as well as calculation operators ’**+**’, ’**-**’, ’*****’, ’**/**’ operate on numerical values only, not on binary values.
  - Comparison operators ’**==**’ and ’**!=**’ operate on either binary values or numerical values, you cannot compare a binary value to a number.
  - Negation ’**!**’ operates on binary values only.
- Boolean constants are written “True“ and “False“, not “TRUE“ and “FALSE“ anymore.
- The decimal point always is a dot ’.’
- Variables within a `PRINTF`-command are written with a comma ’,’ as separator.

In order to keep effort for migrating a SIMIT 5 script as small as possible, some enhancements to the new script syntax are accepted in order to remain compatible to SIMIT 5:

- When using an option dialog the mode may be omitted. In this case a Yes-No-Cancel dialog is displayed like in SIMIT 5, i.e.

  DIALOG “Ready ?“

  yields the same result as

  DIALOG “Ready ?“ YESNOCANCEL
- An assignment can be written in the form

  “Variable“ = Value

  but can also be written by use of the `SET-VAR`-command:

  SET-VAR “Variable“ (Value)

  Please note that the value needs to be written in brackets.
- Within an `IF`-condition the `THEN`-keyword may be omitted provided that the expression is written in brackets, i.e. instead of

```
IF Expression THEN
    Block
ELSE ... or ENDIF
```

you may also write

```
IF (Expression)
    Block
ELSE ... or ENDIF
```

Copyright © Siemens AG, 2013