

## Interspec-Unilab Interface 5.0

### Description and Setup of the Interspec-Unilab Interface

Interface Documentation

---

Preface

---

Table of Contents

---

---

Overview

---

**1**

---

Functional Description

---

**2**

---

Setting Up the Interface

---

**3**

## Guidelines

This manual contains notices intended to protect the products and connected equipment against damage. These notices are graded according to severity by the following texts:

---

### Caution

Indicates that if the proper precautions are not taken, this can result into property damage.

### Notice

Draws your attention to particularly important information on handling the product, the product itself or to a particular part of the documentation.

---

## Trademarks

SIMATIC® is a registered trademark of SIEMENS AG.

If third parties use, for their own purposes, other names than the names in this document that refer to trademarks, they might infringe upon the rights of the trademark owners.

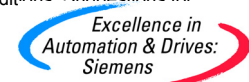
### Copyright © Siemens AG 2004 All rights reserved

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  
Bereich Automation and Drives  
Geschäftsgebiet Industrial Automation Systems  
Postfach 4848, D- 90327 Nuernberg  
Siemens Aktiengesellschaft

### Disclaimer of Liability

We have checked the contents of this manual for agreement with the hardware and software described. Since 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 2004  
Technical data subject to change.

# Preface

## Purpose

This Interface Documentation describes the functionality and architecture of the standard interface between the Siemens A&D AS MES specification management system, SIMATIC IT Interspec and the laboratory information management system, SIMATIC IT Unilab.

## Basic knowledge required

This guide is intended for Interspec-Unilab Interface users who are responsible for system configuration, such as application managers and system integrators (consultants). To be able to understand the concepts and examples discussed in this guide, the reader should at least have taken the Interspec-Unilab Interface Basic Training.

## Where is this manual valid?

This manual is valid for release 5.0 of Interspec-Unilab Interface.

## Online help

The online help of Interspec-Unilab Interface, which is integrated in the software, complements this Interface Documentation and provides you with detailed support for using the software.

## Finding your way

This document consists of 3 parts. The table below explains the structure of the document.

Part	Content
1	General overview of the system
2	Detailed description of the configuration of the interface and the data transfer
3	How you can set up the system

## Conventions

The table below describes the specific typographic conventions that are used throughout this manual:

Symbol/Convention	Indicates...
E.g.	Where examples are given.
<b>Text in bold</b>	The names of menus, commands, dialog boxes and toolbar buttons and, in general, all strings (e.g. <b>File</b> menu; <b>Save</b> command).
KEY1+KEY2	Shortcut keys, which permit rapid access to commands (e.g. CTRL+C).
UPPERCASE	The names of keyboard keys (e.g. RETURN key).
<i>Italics</i>	The names of parameters that must be replaced with a specific name or value.  E.g. <i>filename</i> indicates that the name of the file must be specified; <i>input</i> indicates that the corresponding value must be specified.
>	A succession of commands in which the command preceding the symbol must be selected before the command following it.

## Training Center

Siemens A&D AS MES offers a number of training courses to familiarize you with the SIMATIC IT product suite. To successfully achieve this goal, training consists of lessons in both theory and practice.

There are courses the whole year through, according to a program that is published well in advance of the first scheduled session.

The material on the basis of which our courses are conducted reflects the result of years of experience in process, LIMS and quality control and production management.

All courses are given by expert personnel that is aware of the developments and innovations in the Siemens A&D AS MES product suite.

Courses are held in English at the Siemens A&D AS MES Training Centers in Genoa, Italy and Ninove, Belgium.

Upon request, training courses can also be organized on the customer's premises.

For more information on the training course calendar, please visit our Technical Web site (<http://www.simaticit.com/> **Support > Training**).

## A&D AS MES Service & Support

A comprehensive **Software Maintenance** program is provided with SIMATIC IT products.

Software Maintenance includes the following services:

- Software Update Service (SUS): automatic distribution of upgrades and service packs
- Access to Technical Support Service (TSS): support on technical problems with SIMATIC IT software
- Access to Technical Support Web: a technical web site that includes information such as Frequently Asked Questions and technical documentation on SIMATIC IT products
- Access to SIMATIC IT Club: business web site to keep you informed of the latest news and initiatives on SIMATIC IT products

### Software Update Service (SUS)

This service automatically distributes new versions and service packs when they are released. Typically, when a version becomes available, the installation CD is distributed within a month. For each server that is covered by Software Maintenance, one copy of the installation CD is shipped

Hot fixes are not shipped and can only be downloaded from the Technical Support Web site.

### Technical Support Service (TSS)

SIEMENS has a dedicated technical support team for SIMATIC IT products.

Customers that have a Software Maintenance program, can directly access TSS.

To be able to access TSS, you need to register yourself as a user at the Technical Support Web site: <https://tss.simaticit.com> Complete the registration form with:

- Personal data
- Company and plant information
- The number to order licenses or Software Maintenance Renewal at the Siemens offices.

On the Technical Support Web site you can find:

- The service conditions
- The phone numbers
- The working hours
- The reaction times

## Technical Support Web (TSW)

Every customer with a registered TSS user can access the Technical Support Web site (<https://tss.simaticit.com>). TSW contains technical information such as:

- SIMATIC IT knowledge base: a technical support database that includes practical service solutions from Technical Support or the SIMATIC IT community
- SIMATIC IT software (e.g. hot fixes, software examples) and release notes that can be downloaded
- SIMATIC IT Cross-Industry libraries that can be downloaded (limited access to SIMATIC IT Certified Partners)
- SIMATIC IT Product Documentation that can be downloaded
- Frequently Asked Questions and useful tips.

## Club SIMATIC IT

Club SIMATIC IT is a private web site that contains sales and marketing information on the SIMATIC IT products.

SIMATIC IT partners and end users can access the Club SIMATIC IT web site (<https://club.simaticit.com>).

## A&D Central Support

Customers who use a SIMATIC IT product but do not have a direct contract with Siemens A&D AS MES, can contact Siemens A&D Central Support.

The table below contains the details to contact Siemens A&D Central Support:

<b>Worldwide (Nürnberg) Technical Support</b>	
24 hours a day, 365 days a year	
Phone:	+49 (0) 180 5050-222
Fax:	+49 (0) 180 5050-223
E-Mail:	adsupport@siemens.com
GMT:	+1:00
The languages of the SIMATIC Hotlines are generally German and English.	

A central dispatch service will then forward the request to the Siemens A&D AS MES Technical Support Team.

This kind of support is free of charge, but customers are only entitled to receive support for a total of one hour per issue.

# Table of Contents

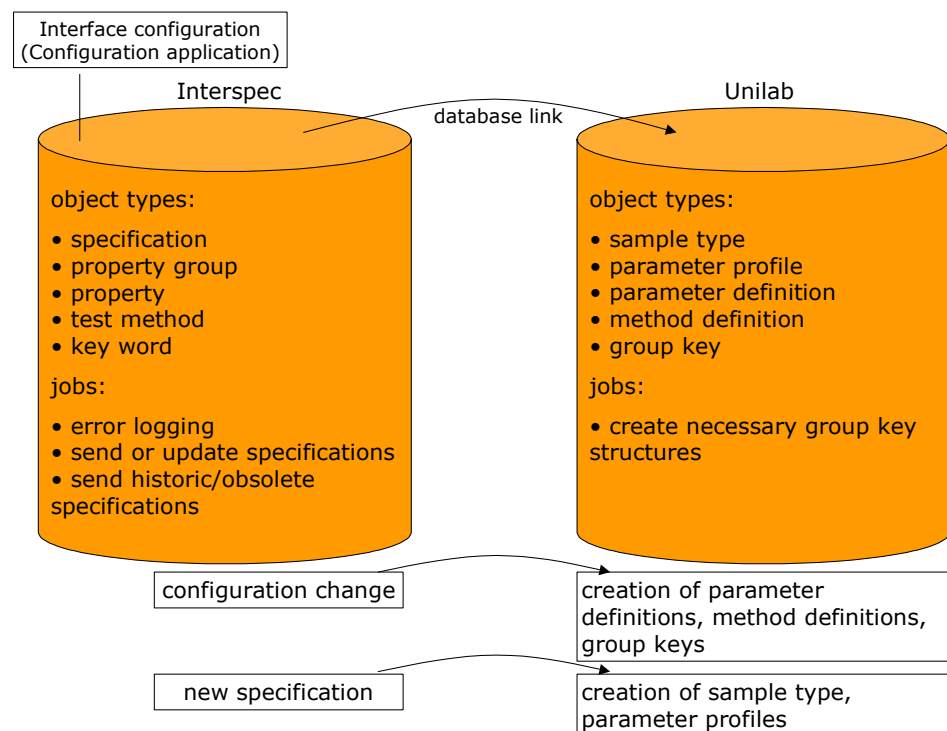
<b>1 Overview .....</b>	<b>9</b>
1.1 Schematically .....	9
1.2 Procedure .....	10
1.2.1 General .....	10
1.2.2 Version Alignment .....	11
1.2.3 Interfacing Interval .....	11
1.3 One Interspec Database vs. Multiple Unilab Databases .....	12
<b>2 Functional Description .....</b>	<b>13</b>
2.1 The Configuration of the Interface .....	13
2.1.1 Introduction .....	13
2.1.2 Configuration of the Plants .....	13
2.1.3 Configuration of the Display Formats .....	14
2.1.4 Configuration of the Key Words .....	24
2.1.5 Configuration of the Templates .....	25
2.1.6 Configuration of the Preference Settings .....	26
2.2 Data Transfer from Interspec to Unilab .....	30
2.2.1 Introduction .....	30
2.2.2 Configuration Data .....	30
2.2.3 Specification Data .....	33
2.2.4 Linked Specifications .....	38
<b>3 Setting Up the Interface .....</b>	<b>41</b>
3.1 What Do You Need? .....	41
3.2 Before Starting .....	41
3.3 Installation Procedure for the Unilab Database .....	42
3.3.1 Enabling the Database Jobs .....	42
3.3.2 Installing the Packages .....	42
3.3.3 Checking the Status of the Packages .....	43
3.3.4 Starting the Database Jobs .....	43
3.3.5 Installing the Views .....	43
3.3.6 Setting the Database Parameters .....	43
3.3.7 Creating the Templates .....	44
3.3.8 Adapting the Lifecycle [Optional] .....	44
3.4 Upgrade Procedure for the Unilab Database .....	45
3.4.1 Upgrading the Packages .....	45
3.4.2 Checking the Status of the Packages .....	45
3.4.3 Starting the Database Jobs .....	46
3.4.4 Upgrading the Other Objects .....	46
3.5 Installation Procedure for the Interspec Database .....	46
3.5.1 Installing the Database Link .....	46
3.5.2 Installing the Grants .....	47
3.5.3 Installing the Add-on Packages .....	47
3.5.4 Installing the Tables, Views, Sequences and Initial Data .....	48
3.5.5 Installing the Packages and Triggers .....	48
3.5.6 Checking the Status of the Packages .....	49
3.5.7 Starting the Database Jobs for the Interface .....	50
3.5.8 Modifying the Configuration .....	51
3.5.9 Installing the LIMS Language .....	52

3.6	Upgrade Procedure for the Interspec Database .....	52
3.6.1	Stopping the Database Jobs for the Interface .....	52
3.6.2	Installing the Add-on Packages .....	52
3.6.3	Upgrading the Packages .....	53
3.6.4	Checking the Status of the Packages.....	53
3.6.5	Starting the Database Jobs for the Interface.....	54
3.6.6	Upgrading the Other Objects.....	54



# 1 Overview

## 1.1 Schematically



## 1.2 Procedure

### 1.2.1 General

One of the most important rules is that Interspec will always be the master of the specifications where the following object mapping is applied:

Nbr	Interspec Object Type	Unilab Object Type
1	Specification	Sample type
2	Property group	Parameter profile
3	Property	Parameter definition
		Sample type group key
		Sample type standard attribute
		Sample type user-defined attribute
4	Test method	Method definition
5	Key word	Sample type group key

The Interspec release includes a screen where the user can map the different headers in an Interspec display format with the available fields in Unilab. When this mapping screen is changed, this change is applicable to all future specification transfers between Interspec and Unilab. The change is not applicable to the specifications which were already transferred from Interspec to Unilab, unless an update of this specification in Interspec results in a new transfer to Unilab. When the interface starts, it searches all specifications that became **Approved** in the last 14 days.

- When you manually transfer objects, you have to make sure that all configuration data has been sent before the specifications are transferred.
- When you run the database job, the interface makes sure that all necessary parameter profiles, parameters and methods are created before copying the values.

You can base the creation of these objects on a predefined template in Unilab, but if there is no template available, the hardcoded values will be used for the object properties. The description and ID are defined in Interspec. All transferred specifications are stored in a history table (**itlimsjob\_h**), which you can consult in Interspec.

The interface works on two levels:

- Configuration change in Interspec
- New specification in Interspec

## Configuration change in Interspec

A configuration change in Interspec creates in Unilab the corresponding:

- Parameter definitions (**utpr**)
- Method definitions (**utmt**)
- Links between method definitions and parameter definitions (**utprmt**)
- Sample type group key definitions (**utgkst**, **utstgkxx**)

## New specification in Interspec

A new specification in Interspec will create in Unilab the required:

- Sample type (**utst**)
- Links between parameter profiles and the sample type including the parameter profile itself and its specification data, parameters and methods to be used (**utpp**, **utstpp**, **utppsp[a|b|c]**, **utpppr**)
- Sample type group key values (**utstgk**)
- Sample type user-defined attribute values (**utstau**)
- User-defined attributes on the link between parameter profile and parameter (**utppprau**).

### 1.2.2 Version Alignment

The object revision in Interspec corresponds to the major object version in Unilab. If the major version does not exist yet, it is created. If the object with the specified major version is found in the Unilab database, a status check is done.

- If the object with the highest minor version is modifiable, this object will be adapted.
- If the object with the highest minor version is not modifiable, a new minor version will be created.

Keep in mind that the version of the sample type and parameter profile corresponds to the specification revision, while the version of the parameter definition and method definition corresponds to, respectively, the revision of the property and test method.

### 1.2.3 Interfacing Interval

All specifications with the status **Approved** are inserted into a working table (**itlimsjob**) that contains all specifications that have to be processed and all specifications that were already processed. It is not advisable to immediately send the specification to Unilab when the status is changed to **Approved**, because this would slow down the status change logic. As a result you need to implement a background process that reads this table and processes the specifications one by one.

### 1.3 One Interspec Database vs. Multiple Unilab Databases.

A company will often have only one Interspec database at corporate level, while Unilab databases often run per plant or set of plants. All specifications that need to be processed will be grouped by their associated plant and will be exported to the Unilab database that corresponds to that plant. Therefore, you need to implement a mapping table (**itlimsplant**) that stores the SQLNet connect string per plant. Because the procedure will make use of a database link, the database link will have to be created dynamically. If the connection to one of the databases is not successful, the specification/plant combination will not be indicated as processed and an error will be logged in the **iterror** table.

## 2 Functional Description

### 2.1 The Configuration of the Interface

#### 2.1.1 Introduction

The configuration includes the configuration of:

- Plants
- Display formats
- Key words
- Templates
- Preference settings

You can configure all these items, except the preference settings, in the **Configuration** application through the menu **Property > Display Format > Setup LIMS**.

The preference settings for the interface can be found on the menu **Other > Preferences**.

#### 2.1.2 Configuration of the Plants

Every plant in Interspec can have its own Unilab database, so you need to configure the Unilab database connect string and the language for each plant.

---

**Note**

It is not possible to configure an Unilab database for the plants **0** (Default plant), **DEV** (Development Plant) and **INTL** (International BOM Plant).

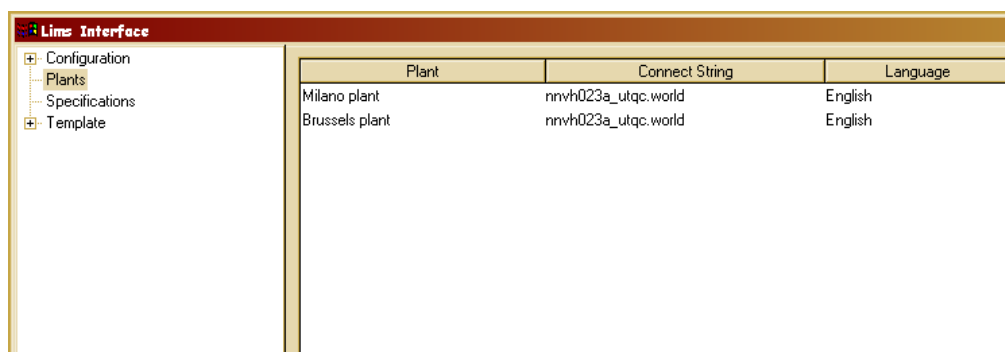
The ID of the objects that will be transferred to Unilab, is the description in the language LIMS, if this is configured (cfr. [Data Transfer from Interspec to Unilab](#)). When you use this mechanism of LIMS descriptions, you have to set the language of the plant to **LIMS**.

---

In the table **itlimspkey**, do not forget to configure the paths where the values of the pp\_keys can be found. For more information, see [Configuration of the Preference Settings](#).

## Example

The connect string and the language for a Unilab database can be configured as follows:

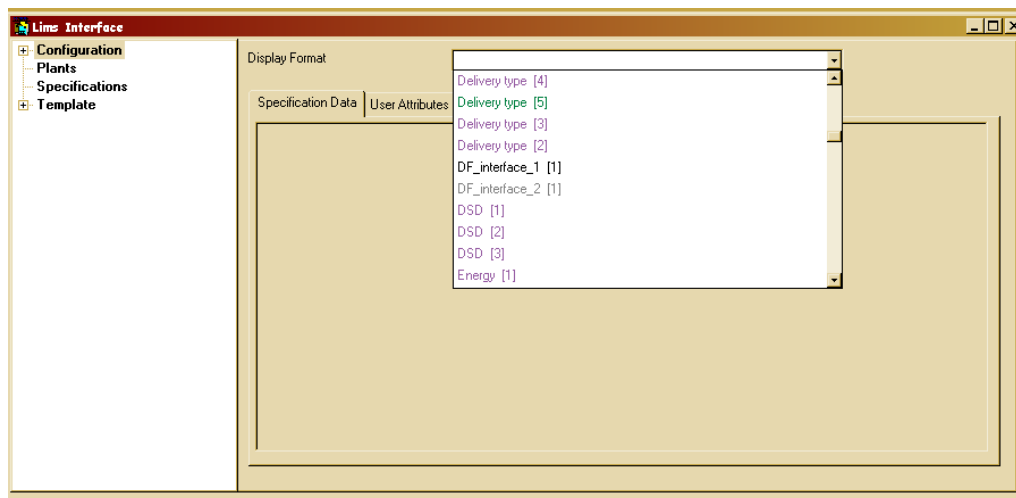


Plant	Connect String	Language
Milano plant	nnvh023a_utqc.world	English
Brussels plant	nnvh023a_utqc.world	English

### 2.1.3 Configuration of the Display Formats

#### Meaning of colors

When you choose a display format from the drop-down list, you can see that the application uses different colors.



The table below lists the four possible colors and their meaning.

Color	Meaning: The display format is...
purple	<ul style="list-style-type: none"> <li>Not set up for LIMS.</li> <li>Not used in a specification.</li> </ul>
green	<ul style="list-style-type: none"> <li>Not set up for LIMS.</li> <li>Used in a specification.</li> </ul>
grey	<ul style="list-style-type: none"> <li>Set up for LIMS.</li> <li>Not used in a specification.</li> </ul>
black	<ul style="list-style-type: none"> <li>Set up for LIMS.</li> <li>Used in a specification.</li> </ul>

## Configuration in four parts

The configuration of the display format can be divided into four parts:

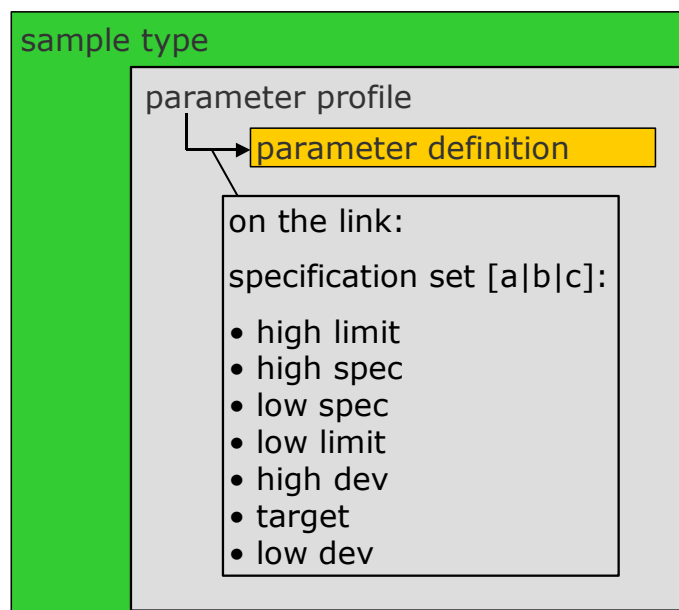
- Specification data
- User attributes
- Standard attributes
- Group keys

## Specification data

The link between a parameter definition and a parameter profile in Unilab has a number of characteristics. These characteristics are grouped into specification sets. One specification set consists of:

- Low limit
- High limit
- Low spec
- High spec
- Target
- Low dev
- High dev
- Rel high dev
- Rel low dev

The illustration below shows an overview for specification data.



Every link between a parameter definition and a parameter profile has 3 specification sets (A,B,C).

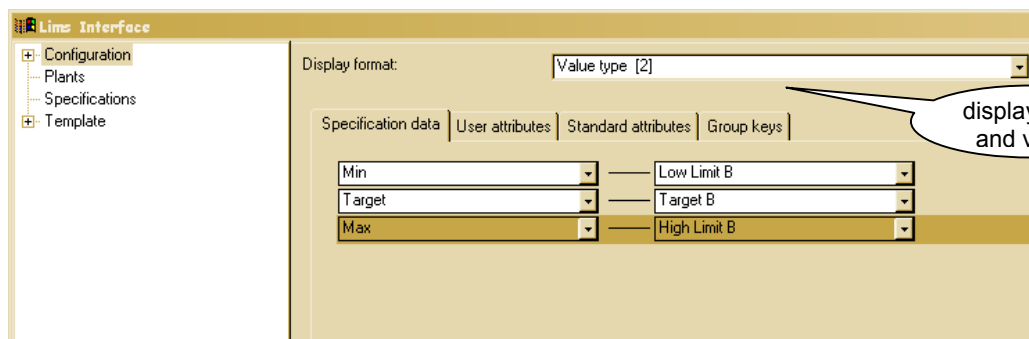
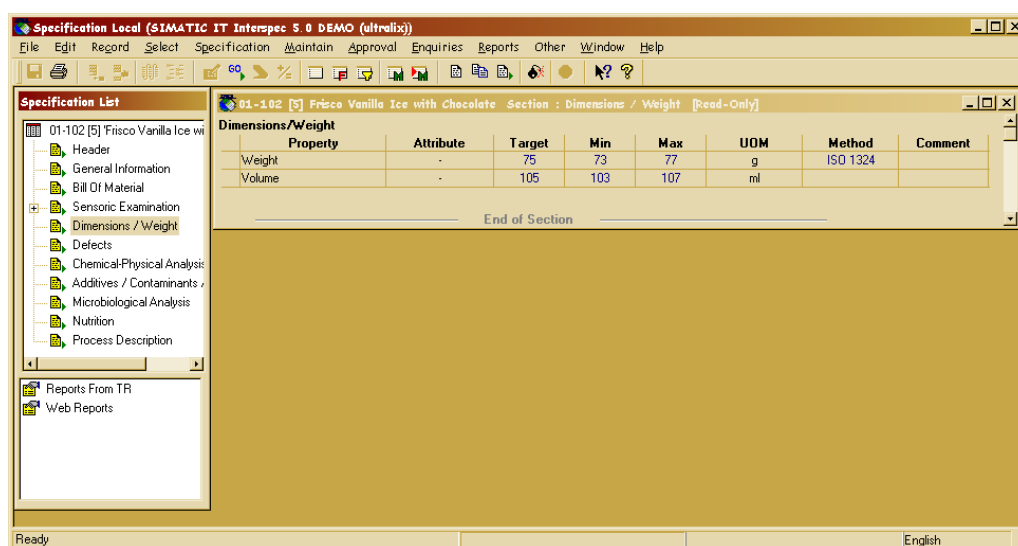
A display format in Interspec defines how a property group or property is displayed. The fields of the display format can be mapped with one of the Unilab fields of a specification set.

### Note

Only numeric specification data can be mapped with the Unilab specification sets.

### Example

You need to set up this configuration for each display format used in a property group/property that needs to be transferred to Unilab.





Parameter Profile <Dimensions/Weight> Dimensions/Weight

Name: Dimensions/Weight Plant:  Status: Approved

Description: Dimensions/Weight Product: 01-102 Life cycle: System LC

Version: 0005.00 Leverancier:  Klant:

☒ Current

Parameter	Parameter	Assign. Freq.	#P.	Method	# M.	Low Limit	Low Spec	Target	High Limit	High Spec
Weight	~Current~	Always...	1		0	73,00		75,00	77,00	
Volume	~Current~	Always...	1		0	103,00		105,00	107,00	

## User attributes

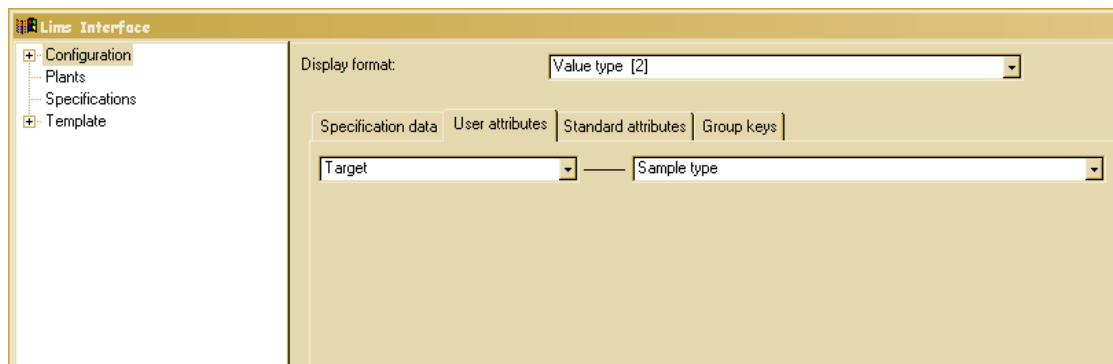
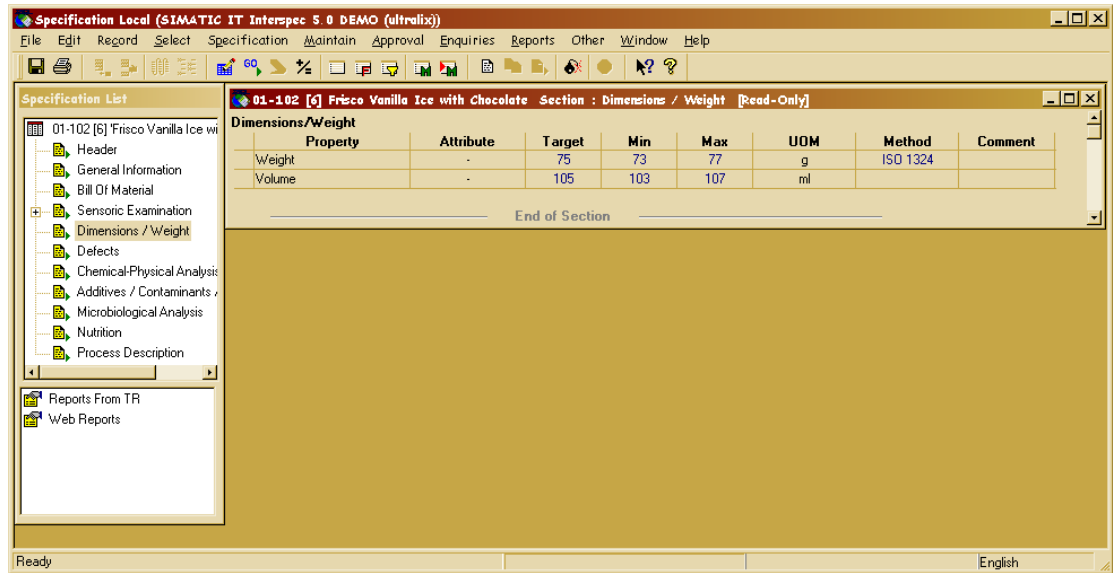
The user attributes can be configured on two levels:

- Sample type
- The link between a parameter definition and a parameter profile.

You can only configure the fields of the display format of the following types as a user attribute:

- UoM
- Numeric
- Character
- Boolean
- Date
- Association
- Test method
- Test detail

## Example



Name	Value
FatTarget	16
pHTarget	6.5
VolumeTarget	105
WeightTarget	75

☐ Inherit attributes

Assign...  
Launch  
Remove

OK Cancel Help

## Standard attributes

The Unilab standard attributes for a sample type that can be set via the interface are:

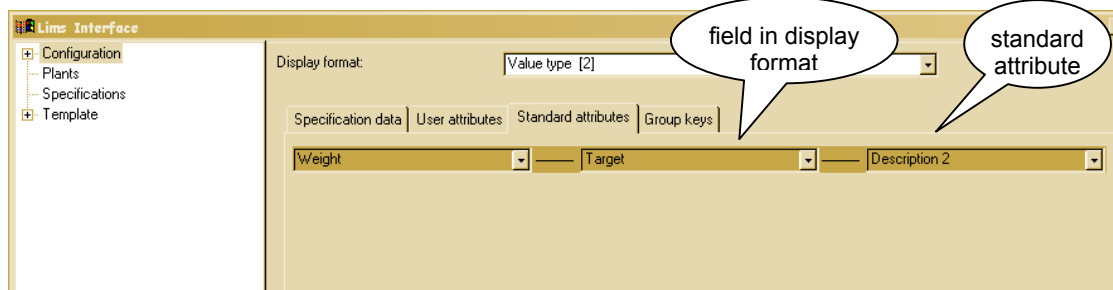
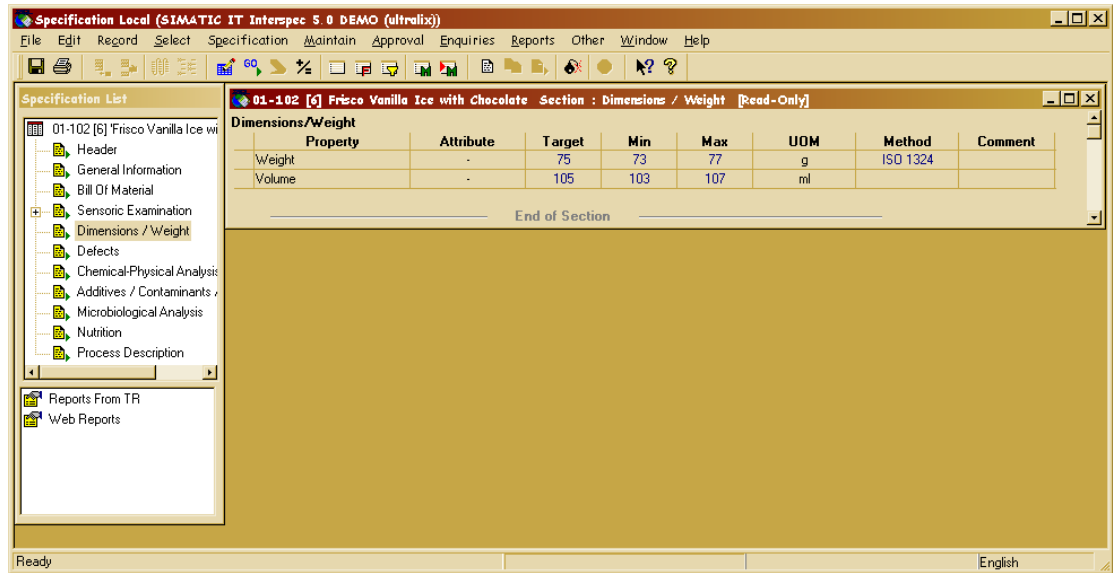
- Description
- Description 2
- Default priority
- Label format
- Shelf life unit
- Shelf life value

Every property of a property group of a specification that uses a display format that is set up for LIMS, can be configured as standard attribute. Besides the property, you also need to configure the field of the display format.

You can only configure the fields of the display format of the following types as a standard attribute:

- UoM
- Numeric
- Character
- Boolean
- Date
- Association
- Test method
- Test detail

## Example



**Properties of Sample Type <01-102>**

Standard Attributes | Planning | Access Rights | Audit Trail | Attributes | Group Keys

**Identification**

Description: Frisco Vanilla Ice with Chocolate      Status: In Editing

Description (2): 75      Life cycle: System LC

**Version control**

Version: 0006.00      Effective From: 26/11/2003 0:00:00

☐ Current      Effective Till: 26/11/2003 9:24:41

☐ Assign any parameter profile      Life cycle of sample: [dropdown]

☐ Can be used as a template      SOP: [text box]

☐ User ID must be confirmed      Default priority: [dropdown]

Shelf life limit: 0 Days      Sample code mask: Basic Mask

Label Format: [dropdown]

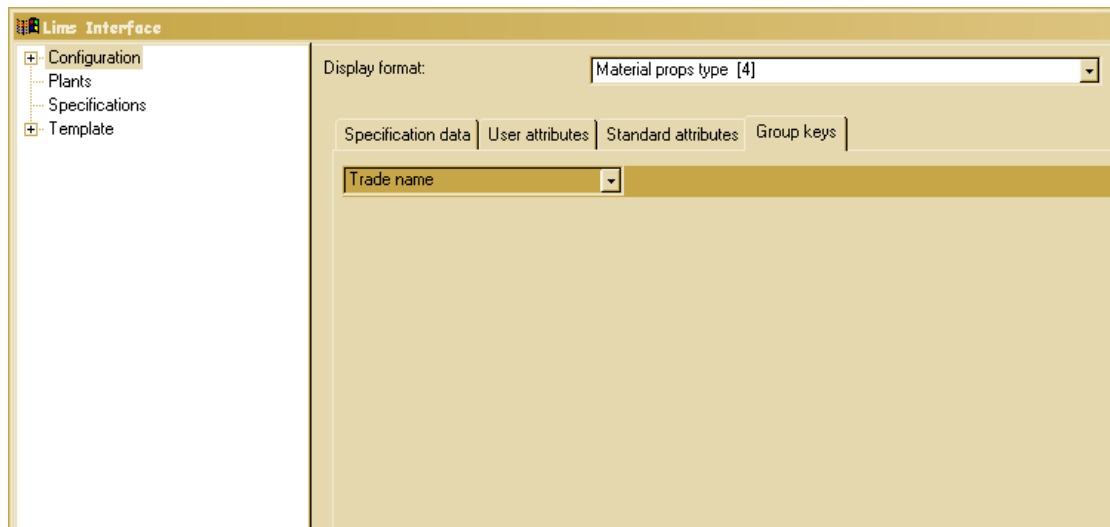
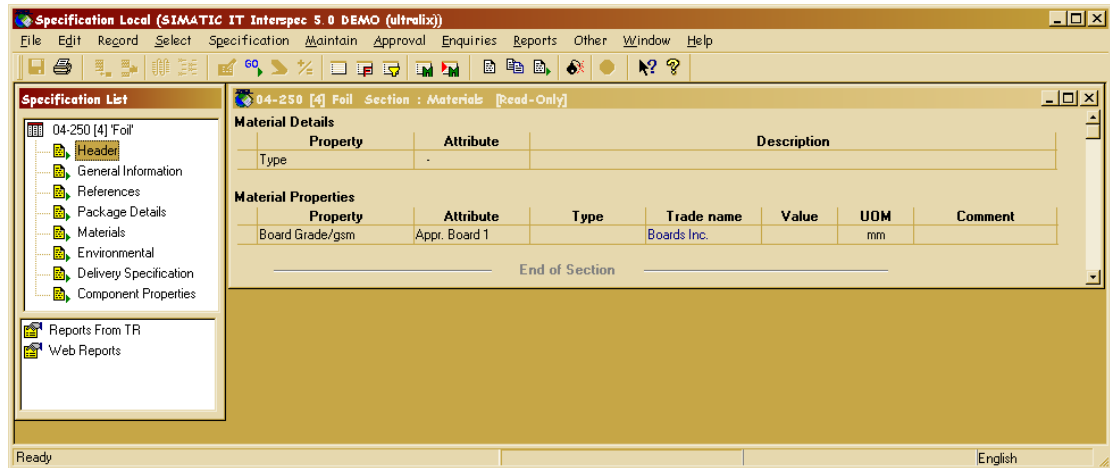
OK Cancel Help

## Group keys

You can only configure the fields of the display format of the following types as a group key of a sample type:

- UoM
- Numeric
- Character
- Boolean
- Date
- Association
- Test method
- Test detail.

## Example



Properties of Sample Type <04-250>

Standard Attributes | Planning | Access Rights | Audit Trail | Attributes | Group Keys

Name	Value
Locally_Sourced	No
Packaging_Level	Secondary
plant	BXL
plant	ML
Trade_name	Boards Inc.

☐ Inherit groupkeys

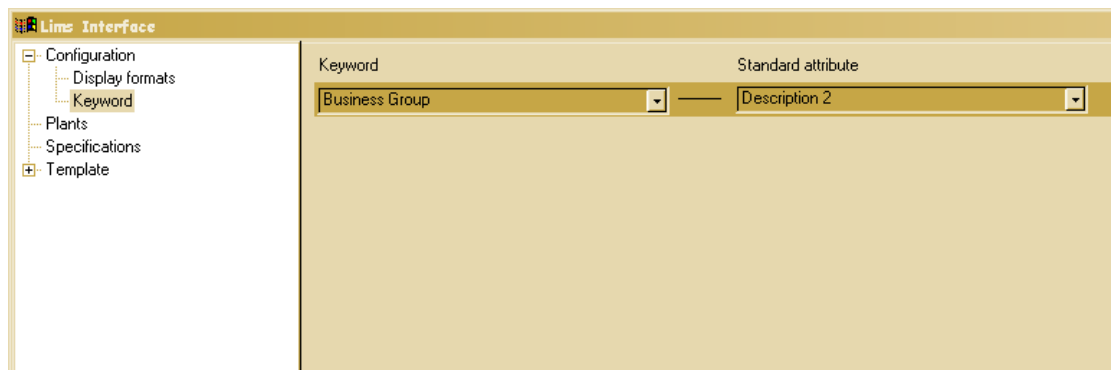
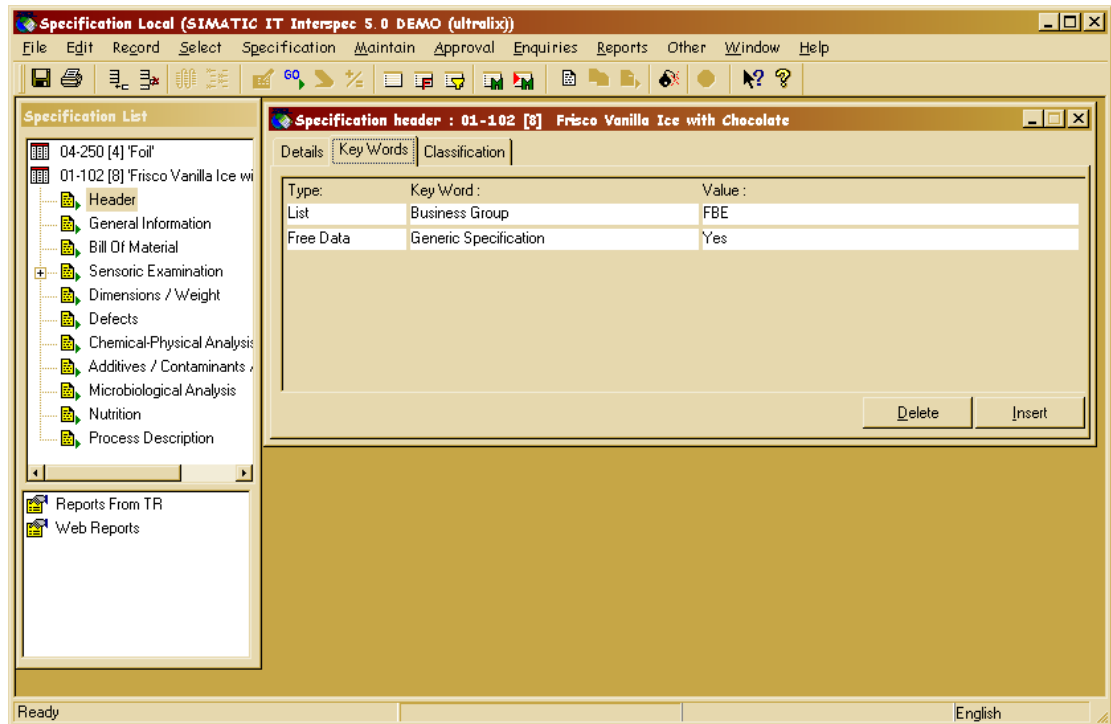
Assign...  
Remove

OK Cancel Help

## 2.1.4 Configuration of the Key Words

You can configure a key word in Interspec as a standard attribute.

### Example





**Properties of Sample Type <01-102>**

Standard Attributes | Planning | Access Rights | Audit Trail | Attributes | Group Keys

**Identification**

Description: Frisco Vanilla Ice with Chocolate      Status: In Editing

Description (2): FBE      Life cycle: System LC

**Version control**

Version: 0008.00      Effective From: 2/12/2003 0:00:00

☐ Current      Effective Till: 26/11/2003 10:53:54

☐ Assign any parameter profile      Life cycle of sample: [dropdown]

☐ Can be used as a template      SDP: [text box]

☐ User ID must be confirmed      Default priority: [dropdown]

Shelf life limit: 0 Days      Sample code mask: Basic Mask

Label Format: [dropdown]

OK Cancel Help

## 2.1.5 Configuration of the Templates

Because a lot of standard attributes of Unilab objects cannot be defined in Interspec, you can define a template in Unilab for each interfaced object. The interface can use this template together with the information from Interspec to create a new object. If there is no template available, hardcoded values will be used. Be aware of the fact that the templates are only used to inherit the object standard properties, and not the attributes, group keys, method cells, info profiles, or other configured items. A template can be defined for a:

- Sample type
- Parameter profile
- Parameter definition
- Method definition.

---

### Note

The templates are the same for all plants.

---

### Sample type

For each specification type you can configure a template for a sample type.

You can configure a default template, which will be used for all specifications of a certain type for which there is no template configured.

## Parameter profile

For each display format you can configure a template for a parameter profile.

You can configure a default template, which will be used for all properties group with a display format for which there is no template configured.

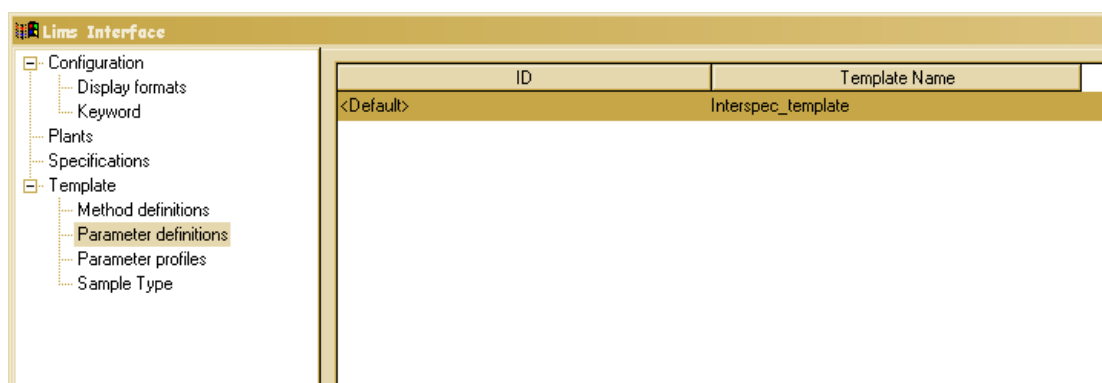
## Parameter definition

There can only be one template for a parameter definition.

## Method definition

There can only be one template for a method definition.

## Example



## 2.1.6 Configuration of the Preference Settings

The table below lists the preference settings.

Value	Description
<b>LIMS Enabled</b>	
1	The Unilab Interface is enabled. (default)
0	The Unilab Interface is disabled.
<b>Transfer Cfg MT</b>	
1	The test methods linked to a property in the Interspec configuration are transferred to Unilab, provided that the property meets the necessary requirements. (mt, prmt) (default)
0	The test methods linked to a property in the Interspec configuration are not transferred to Unilab. (mt, prmt)
<b>TMP Master PR</b>	
1	The creation of the parameter definition fails if the configured template does not exist. (default)
0	The parameter definition is created with default values if the template does not exist.

Value	Description
<b>TMP Master MT</b>	
1	The creation of the method definition fails if the configured template does not exist. (default)
0	The method definition is created with default values if the template does not exist.
<b>TMP Master ST</b>	
1	The creation of the sample type fails if the configured template does not exist. (default)
0	The sample type is created with default values if the template does not exist.
<b>TMP Master PP</b>	
1	The creation of the parameter profile fails if the configured template does not exist. (default)
0	The parameter profile is created with default values if the template does not exist.
<b>Master Standard Au</b>	
PROPERTY	If a property and a key word are mapped with the same standard attribute of a sample type, the value of the property will become the value of the standard attribute. If the property has no value or the property is not configured, the value of the key word will become the value of the standard attribute. (default)
KEY WORD	If a property and a key word are mapped with the same standard attribute of a sample type, the value of the key word will become the value of the standard attribute. If the key word has no value, the value of the property will become the value of the standard attribute.
<b>MT Based On PR</b>	
1	A method definition is created with the same ID, description as the parameter definition. (default)
0	Methods are based on Interspec methods.
<b>Transfer PPR MT</b>	
1	In Interspec a test method used as part of a property value in a property group, will be interfaced and become a method as part of the parameter profile – parameter link in Unilab.(ppprmt) (default)
0	In Interspec a test method used as part of a property value in a property group, will not be interfaced to become a method as part of the parameter profile – parameter link in Unilab.
<b>Inherit Linked Spc</b>	
1	The parameter profiles coming from linked specifications are a combination of the test plan of the generic specification and the test plan defined in the linked specifications. Use this option when the company's production is based on orders. E.g. for customer specifications, only after the production it is decided to which customer the product will be delivered. Therefore the lab must execute every test (depending on the cost) a customer could ask for. (default)
0	The generic test plan will only be transferred to the generic specification/sample type. The parameter profiles coming from linked specifications will only contain the tests as defined in the linked specification.

Value	Description
<b>KW ID Generic Spc</b>	
<key_word_ID>	The ID of the key word that indicates a generic specification. (default value = 0)
<b>Transfer All Cfg</b>	
1	All configuration data (methods, properties, links between methods and properties, properties+attributes, links between methods and properties+attributes) has to be transferred to Unilab.
0	Only the configuration data (methods, properties, links between methods and properties, properties+attributes, links between methods and properties+attributes) that is used in a specification that is in the queue to transfer (table <b>itlimsjob</b> ) has to be transferred to Unilab. (default)

The table **itlimsppkey** is a kind of extension of the normal preference settings table. It contains the paths where the values for the corresponding Unilab pp\_keys can be found.

These settings are stored in a separate table to avoid interference with the preference table. There are 5 settings per Unilab database, so if there are 3 Unilab databases, the number of settings quickly increases to 15.

Column **db** contains the name of the Unilab database link and in column **pp\_key\_seq** the sequence number of the pp\_key is stored. For the column **path**, following values are possible:

<b>Path</b>	
sample type	For the pp_key <b>Product</b> , no path has to be specified, because the sample type corresponds to the part_no of the specification.
plant	You do not need to specify the path for pp_key <b>Plant</b> because: <ul style="list-style-type: none"> <li>The pp_key <b>Plant</b> is not set in Unilab.</li> <li>The plants of a specification can simply be found by querying a table, not by following a certain path.</li> </ul>
<section_ID># <subsection_ID># <property_group_ID># <property_ID># <type of display format column>	For a linked specification, the corresponding pp_key has to be filled in with the correct value. The setting defines the path where this value can be found.
0	This key is not defined on the Unilab database. (default)

## Example

Configure a frame for the customer specification and fill in a value for the customer code. Execute the following sql statement:

```
SQL> SELECT fr.*
      FROM frame_prop fr, section sc, sub_section ssc,
           property_group pg, property sp
      WHERE fr.frame_no      = '<frame_no>'
      AND fr.revision        = <frame_revision>
      AND fr.section_ID      = sc.section_ID
      AND fr.sub_section_ID  = ssc.sub_section_ID
      AND fr.property_group  = pg.property_group
      AND fr.property        = sp.property
      AND sc.description     = '<section_descr>'
      AND ssc.description    = '<sub_section_descr>'
      AND pg.description     = '<property_group_descr>'
      AND sp.description     = '<property_descr>';
```

with <xxx> replaced by the correct values. If there is no sub\_section, set the value to **(none)**.

For example, this will result in value **700480#0#700462#702823#char\_6**:

- **700480** = value of column section\_ID
- **0** = value of column sub\_section\_ID
- **700462** = value of column property\_group
- **702823** = value of column property
- **char\_6** = column\_name of the column where the customer code is stored.

## 2.2 Data Transfer from Interspec to Unilab

### 2.2.1 Introduction

There are two types of data you can send to Unilab:

- Configuration data
- Specification data.

### 2.2.2 Configuration Data

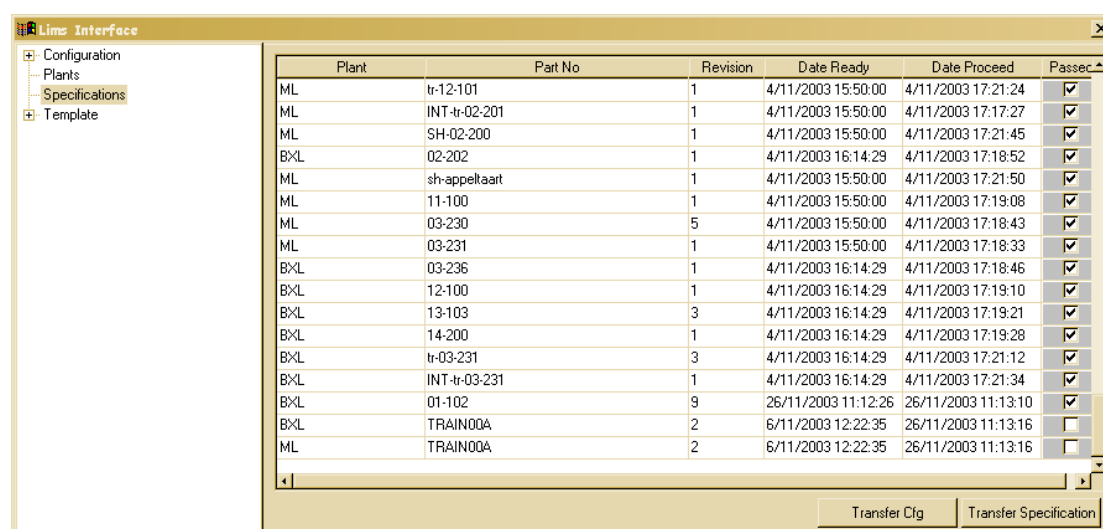
#### What?

The configuration data consist of:

- Parameter definitions
- Method definitions
- Links between method definitions and parameter definitions
- Sample type group key definitions.

Click the **Transfer Cfg** button in the LIMS interface window to send this kind of data.

#### Example



Plant	Part No	Revision	Date Ready	Date Proceed	Passec
ML	tr-12-101	1	4/11/2003 15:50:00	4/11/2003 17:21:24	<input checked="" type="checkbox"/>
ML	INT-tr-02-201	1	4/11/2003 15:50:00	4/11/2003 17:17:27	<input checked="" type="checkbox"/>
ML	SH-02-200	1	4/11/2003 15:50:00	4/11/2003 17:21:45	<input checked="" type="checkbox"/>
BXL	02-202	1	4/11/2003 16:14:29	4/11/2003 17:18:52	<input checked="" type="checkbox"/>
ML	sh-appeltaart	1	4/11/2003 15:50:00	4/11/2003 17:21:50	<input checked="" type="checkbox"/>
ML	11-100	1	4/11/2003 15:50:00	4/11/2003 17:19:08	<input checked="" type="checkbox"/>
ML	03-230	5	4/11/2003 15:50:00	4/11/2003 17:18:43	<input checked="" type="checkbox"/>
ML	03-231	1	4/11/2003 15:50:00	4/11/2003 17:18:33	<input checked="" type="checkbox"/>
BXL	03-236	1	4/11/2003 16:14:29	4/11/2003 17:18:46	<input checked="" type="checkbox"/>
BXL	12-100	1	4/11/2003 16:14:29	4/11/2003 17:19:10	<input checked="" type="checkbox"/>
BXL	13-103	3	4/11/2003 16:14:29	4/11/2003 17:19:21	<input checked="" type="checkbox"/>
BXL	14-200	1	4/11/2003 16:14:29	4/11/2003 17:19:28	<input checked="" type="checkbox"/>
BXL	tr-03-231	3	4/11/2003 16:14:29	4/11/2003 17:21:12	<input checked="" type="checkbox"/>
BXL	INT-tr-03-231	1	4/11/2003 16:14:29	4/11/2003 17:21:34	<input checked="" type="checkbox"/>
BXL	01-102	9	26/11/2003 11:12:26	26/11/2003 11:13:10	<input checked="" type="checkbox"/>
BXL	TRAIN00A	2	6/11/2003 12:22:35	26/11/2003 11:13:16	<input type="checkbox"/>
ML	TRAIN00A	2	6/11/2003 12:22:35	26/11/2003 11:13:16	<input type="checkbox"/>

## Parameter definitions

A parameter definition in Unilab is a property in Interspec. All the properties that have the following characteristics are transferred to Unilab:

- The property is not **Historic**.
- The property belongs to a property group.
- The property group is not **Historic**.
- The property group has a display format that is set up for LIMS.
- The display format is used in a specification that is in the queue to transfer (table **itlimsjob**). This condition is only valid if preference setting **Transfer All Cfg = 0**.

The next list contains some general principles and remarks:

- The parameter definitions are based on a template (if there is one configured).
- The ID of the parameter definition is the description in language LIMS of the property (limited to 20). If no LIMS description is found, the normal description is taken.
- The description of the parameter definition is the description of the property (limited to 40).
- The parameter definition has an attribute with the ID of the property.
- The transfer of parameter definitions is handled one by one. This means that if the transfer of one of the parameter definitions fails, the other parameter definitions can be transferred.
- If the description of a property in Interspec changes, the description in Unilab will be modified as well.
- A parameter definition that is based on a template inherits all properties from the template with the exception of the flag that indicates whether the parameter definition can be used as template.
- A property that becomes **Historic** in Interspec has no influence on the Unilab configuration.

## Method definitions

A method definition in Unilab is a test method in Interspec. All test methods that have the following characteristics are transferred to Unilab:

- The test method is not **Historic**.
- The test method belongs to a property.
- The property is not **Historic**.
- The property belongs to a property group.
- The property group is not **Historic**.
- The property group has a display format that is set up for LIMS.
- The display format is used in a specification that is in the queue to transfer (table **itlimsjob**). This condition is only valid if preference setting **Transfer All Cfg = 0**.

The next list contains some general principles and remarks:

- The method definitions are based on a template (if there is one configured.).
- The ID of the method definition is the description in language LIMS of the test method (limited to 20). If no LIMS description is found, the normal description is taken.
- The description of the method definition is the description of the test method (limited to 40).
- The method definition has an attribute with the ID of the test method.
- The transfer of method definitions is handled one by one.
- If the description of a test method in Interspec changes, the description in Unilab will be modified as well.
- A method definition that is based on a template inherits all properties from the template with the exception of the flag that indicates whether the method definition can be used as template.
- A test method that becomes **Historic** in Interspec has no influence on the Unilab configuration.

### Links between method definitions and parameter definitions

All links between method definitions and parameter definitions that have the following characteristics are transferred to Unilab:

- The test method is not **Historic**.
- The test method belongs to a property.
- The property is not **Historic**.
- The property belongs to a property group.
- The property group is not **Historic**.
- The property group has a display format that is set up for LIMS.
- The display format is used in a specification that is in the queue to transfer (table **itlimsjob**). This condition is only valid if preference setting **Transfer All Cfg = 0**.

The next list contains some general principles and remarks:

- The links between method definitions and parameter definitions are not based on a template.
- The transfer of the link between a method definition and a parameter definition is handled one by one.
- If the parameter definition or the method definition does not exist in Unilab, it is not possible to transfer the link.
- A property or test method that becomes **Historic** in Interspec has no influence on the Unilab configuration.
- The version of the method definition on the link between method definition and parameter definition is **~Current~**.



## Sample type group key definitions

A sample type group key definition in Unilab is a key word in Interspec. All the key words that have the following characteristics are transferred to Unilab:

- The key word is not **Historic**.
- The key word is assigned to a specification.

The next list contains some general principles and remarks:

- The ID of the sample type group key definition is the description of the key word (limited to 20, and with the characters ' ', '-', and '.' replaced by '\_'). It is not supported to use reserved words of the database as key word description, because this will generate an error during the creation of the sample type group key structures.).
- If a key word is assigned to a specification that cannot be transferred to Unilab, the key word will be transferred anyway.
- A key word that becomes **Historic** in Interspec has no influence on the Unilab configuration.
- If a specification has to be transferred short after the transfer of the configuration data, it is recommended to manually run the job that creates the group key structures. If not, it is possible that the transfer mechanism tries to save group key values for a group key that has no group key table yet. This will generate an error.

### 2.2.3 Specification Data

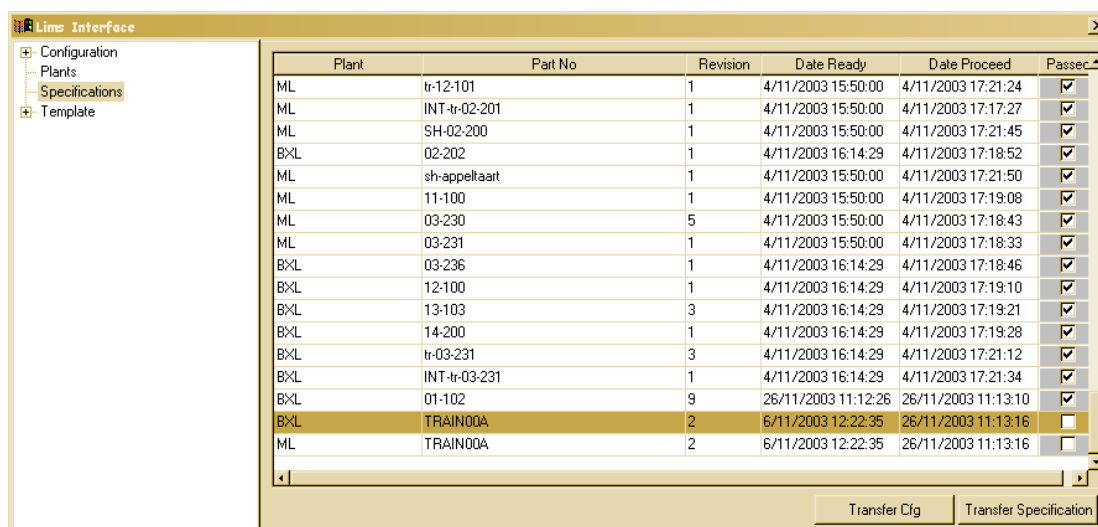
#### What?

A specification in Interspec is a sample type in Unilab. For each sample type the following data must be transferred to Unilab:

- Sample types
- Sample type group keys
- Sample type attributes
- Parameter profiles and the links between parameter profiles and sample types
- Links between parameter definitions and parameter profiles
- Attributes on the links between parameter definitions and parameter profiles
- Linked specifications

## Example

To transfer a specification to Unilab, you need to select it in the LIMS interface window, before you click the button **Transfer Specification**.



## Sample types

A sample type in Unilab is a specification in Interspec. All the specifications that have the following characteristics are transferred to Unilab:

- The specification must have the status **Approved**.
- The specification is linked to a plant that has a Unilab database.
- The specification has a property group that uses a display format that is set up for LIMS.

The next list contains some general principles and remarks:

- The sample types are based on a template (if there is one configured).
- The ID of the sample type is the part number of the specification (limited to 20 characters).
- A specification (except a linked specification) that becomes **Historic** in Interspec makes the corresponding sample type in Unilab **Obsolete**. Also the parameter profiles assigned to the specification become **Obsolete**, and all links between parameter profiles and the sample type are removed.
- The effective date of the specification in Interspec is copied to the effective from date of the sample type in Unilab.

## Sample type attributes

A sample type attribute in Unilab is a property in Interspec. All the properties that have the following characteristics are transferred to Unilab:

- The property belongs to a property group.
- The property group is used in the specification.
- The property group has a display format that is set up for LIMS.
- The fields in the display format are configured as user attributes for a sample type.

The next list contains some general principles and remarks:

- The ID of the sample type attribute is the description of the property. To this description, the description of the attribute is added, if there is one. Finally the header name of the display format column is attached to this string (limited to 20).
- It is possible to configure the interface in a way that a sample type attribute must get the value of more than one property. However this is not supported. The interface will reject the values and an error will be logged in **iterror**.

## Sample type group keys

A sample type group key in Unilab is a key word or a property in Interspec:

- All the key words that have the following characteristic are transferred to Unilab:
  - The key word is assigned to the specification.
- All the properties that have the following characteristics are transferred to Unilab:
  - The property belongs to a property group.
  - The property group is used in the specification.
  - The property group has a display format that is set up for LIMS.
  - There are fields of the display format configured as sample type group keys.

The next list contains some general principles and remarks:

- The ID of the sample type group key is the description of the key word (limited to 20, and with the characters ‘ ‘, ‘-’, and ‘.’ replaced by ‘\_’). It is not supported to use reserved words of the database as key word description, because this will generate an error when the sample type group key structures are created.
- The sample type group key definition for a property is not part of the configuration data but is transferred together with the specification data.
- It is possible to configure the interface in a way that the sample type group key must get its value from a property and a key word. The interface does not support this kind of operation and will reject the values. Instead an error will be logged in **iterror**.
- If a group key is configured as single-valued, it will be transferred to Unilab only once. When the group key is allowed to have more than one value in Unilab, it can be transferred more than once to Unilab. Keep in mind that all values need to be different (cfr. standard Unilab). When a group key is created by the Interspec-Unilab interface, by default it is allowed to have multiple values.

- For all plants that are configured for the specification, a sample type group key **Plant** is assigned with the name of the plant as value.

Furthermore, for all different pp\_key values that exist in the sample type, a sample type group key is assigned. The group key name is the pp\_key name, the value is the pp\_key value.

## Parameter profiles and the links between parameter profiles and sample types

A parameter profile in Unilab is a property group that is a part of a section of a specification in Interspec. All the property groups of the specification that have the following characteristics are transferred to Unilab:

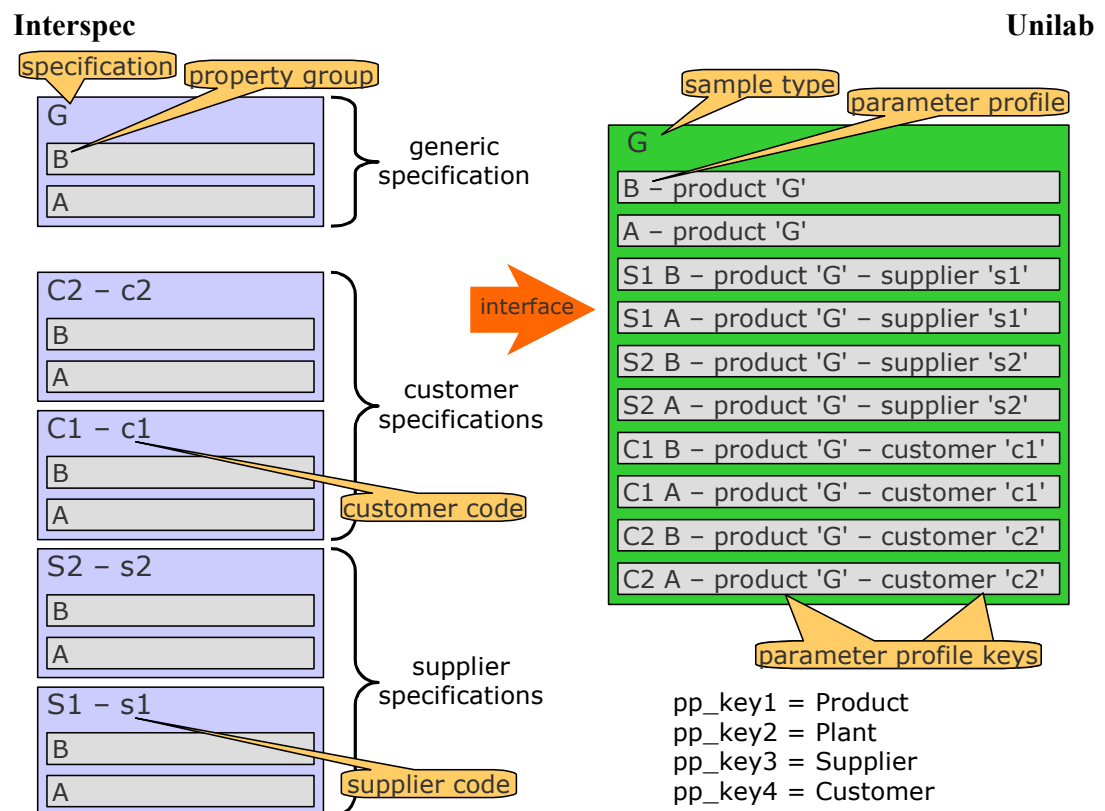
- The property group is used in the specification.
- The property group has at least one property.
- The property group has a display format that is set up for LIMS.
- There are fields of the display format configured as specification data or as user attributes of the parameter profile.

The next list contains some general principles and remarks:

- The parameter profiles are based on a template (if there is one configured).
- The ID of the parameter profile is the description in language LIMS of the property group (limited to 20). If no LIMS description is found, the normal description is selected.
- The description of the parameter profile is the description of the property group (limited to 40).
- The parameter profile has an attribute with the ID of the property group.
- The transfer of the parameter profiles is not the same for the linked specifications (cfr. [Linked Specifications](#)).
- If the sample type or the parameter profile does not exist in Unilab, it is not possible to transfer the link.
- A property group that becomes **Historic** has no influence on the Unilab configuration. If a property group is removed from the specification in Interspec, then the corresponding link between the parameter profile and the sample type in Unilab will be removed.
- The pp\_keys will be filled with the following values:
  - Pp\_key **Product**: the sample type ID
  - Pp\_key **Plant**: no value. The user has to create plant-specific parameter profiles.
  - User-defined pp\_keys: the value that was found by following the path as set in the preference setting **Path pp\_key<x>**.
- It is not mandatory to configure certain pp\_keys, but it is strongly recommended. If no pp\_keys exist, strange and unwanted side-effects may occur. For example, a user transfers different specifications (sample type in Unilab) with the same property group (parameter profile in Unilab). Multiple versions of the same parameter profile will be generated, instead of generating different parameter profiles with the same parameter profile name but with a different pp\_key **Product**.

- The version of the parameter profile on the link between parameter profile and sample type is **<x>.\***, with **<x>** the major version.
- The parameter profiles are assigned to the sample type according to the following sorting rules:
  - generic parameter profiles come first
  - linked parameter profiles are sorted by pp\_key index
  - linked parameter profiles within one pp\_key group are sorted alphabetically by part\_no (first part of parameter profile description = part\_no of the linked specification)
  - the order of linked parameter profiles within one part\_no group (generic or linked) is the order as in Interspec.

Example:



### Links between parameter definitions and parameter profiles

All links between parameter definitions and parameter profiles that have the following characteristics are transferred to Unilab:

- The property is assigned to a property group that is used in the specification.
- The property group has a display format that is set up for LIMS.
- There are fields of the display format configured as specification data or as user attributes of the parameter profile.

The next list contains some general principles and remarks:

- If the parameter profile or the parameter definition does not exist in Unilab, it is not possible to transfer the link.
- For every link between a parameter definition and a parameter profile the specification data (always the absolute values) are also transferred.
- The parameter definitions are part of the configuration data. (cfr. [Configuration Data](#)).
- The version of the parameter and method definition on the link between parameter definition and parameter profile is **~Current~**.
- A property group or property that becomes **Historic** has no influence on the Unilab configuration. If a property is removed from the property group in Interspec, then the corresponding link between the parameter definition and the parameter profile in Unilab will be removed.

## 2.2.4 Linked Specifications

### What are linked specifications?

The example of customer and supplier specifications is a good starting point.

In general, companies often have a corporate specification for a particular Raw Material or Finished Product. However, to save money, a company might want to have a slightly different set of validation specifications per customer/supplier. In principle a possible solution could be to create a specification per customer/supplier, but that would mean that all the Raw Materials and Finished Products were bought or sold under a different item code in the ERP system. A lot of generic properties would have to be maintained per customer/supplier although they always have the same value.

The solution in Interspec to maintain these kind of specifications is the following:

All property groups or free texts that can be overruled in a specification for a Raw Material or Finished Product are put as optional properties or sections into one frame. This frame also contains a section with the **Attached Specification** data type. A second frame will be created for all other data types that are generic across all customers/suppliers.

In the specification module, a generic specification will be created based on the frame that contained all generic data types and per customer/supplier a separate specification will be created based on the frame that only contained the properties and texts that can be overruled. If people want to overrule values in the generic specification, they can add the optional properties and free texts to the customer/supplier specification. In the **Attached Specification** section of the customer/supplier specification the generic specification will be attached.

A standard report has been created that puts the two specifications together into one report, the **General Specification Report**. This report will have some built-in logic that merges the duplicate properties or free texts where the customer/supplier specific values will be the master. This revision displayed in the report will be a combination of the customer/supplier specification with the generic specification.

## General

A generic specification can have more than one linked specification. Each linked specification has its generic specification as attached specification. For every linked specification the following data is transferred:

- The parameter profiles and the link between the parameter profiles and the sample type (sample type that represents the generic specification).
- The link between the parameter definition and the parameter profile.

The next list contains some general principles and remarks:

- A generic specification can be recognized by the key word specified in preference **KW ID Generic Spc**. The key word value has to be set to **Yes**.
- In Unilab, to be able to detect the sample types that are created from a generic specification in Interspec, a sample type group key will be assigned. The name of the group key will be the same as the Interspec key word (e.g. **Generic specification**), and will be stored in the system setting **STGK ID Generic St**. The value of this setting will be set dynamically on transferring a specification from Interspec.
- Sample type group keys are not transferred for linked specifications. Sample type attributes for linked specifications become attributes of the link between the parameter profile and the sample type.
- The description of a parameter profile of a linked specification is the description of the property group (limited to 40), preceded by the part\_no of the linked specification.
- For a linked specification there may be only one attached specification with the key word **Generic specification = Yes**
- The effective date of the linked specification in Interspec is copied to the effective\_from date of the parameter profile in Unilab
- A warning is given when an attempt is made to transfer a linked specification before its generic specification has been transferred
- A linked specification that becomes **Historic** in Interspec makes the corresponding parameter profiles in Unilab **Obsolete**. All links between the parameter profiles and their sample type are removed.

## How will linked specifications be handled in Unilab?

The interface will translate the generic specifications into a sample type. Its property groups become parameter profiles with the pp\_key **Product** specified.

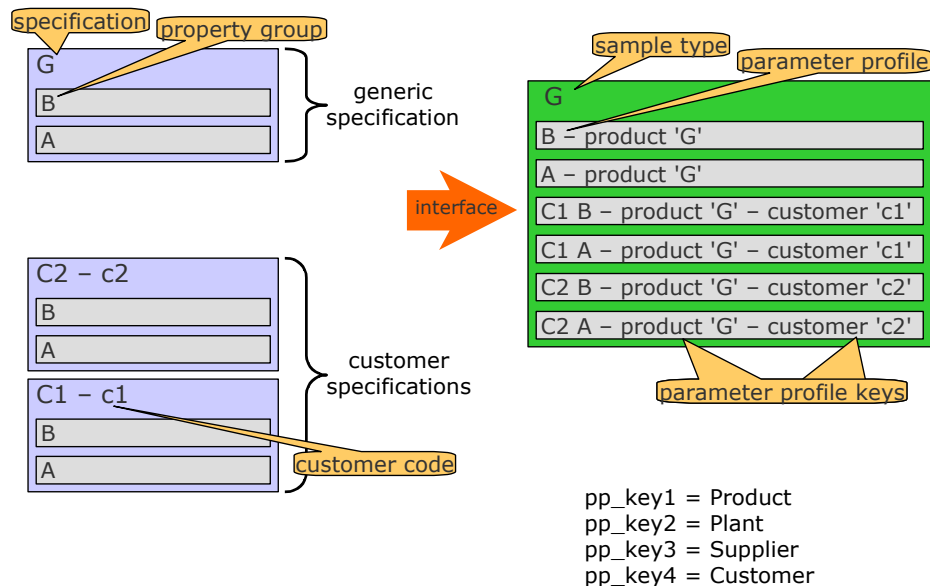
A linked specification does not become a sample type. All property groups of linked specifications will be translated into parameter profiles of the generic sample type with pp\_key **Product** filled in. Also the pp\_key corresponding to the type of the linked specification will be filled in, e.g. pp\_key **Customer** will be set for a customer specification. The value of that pp\_key is the value of a certain property of the linked specification. The path where the value can be found, is defined in the table **itlimspkey** (cfr. [Configuration of the Preference Settings](#)).

## Example

The example below is for a linked specification of the type **customer**.

### Interspec

### Unilab



## What are the advantages of such a configuration?

- A new revision of the generic specification does not force you to release (actually: push through the approval cycle) a new revision of the linked specification if you make use of the phantom logic in Interspec. However, the revision printed on the report will be increased and when this new revision is transferred to Unilab, only the generic parameter profiles are assigned, not the parameter profiles that come from the linked specification.
- A new revision of the linked specification does not force you to release (actually: push through the approval cycle) a new revision of the generic specification if you make use of the phantom logic in Interspec. However, the revision printed on the report will be increased.
- A new linked specification for the same product does not force you to release (actually: push through the approval cycle) a new version of the generic specification. If the linked specification is an attachment of the generic specification, this can create a problem.



## 3 Setting Up the Interface

In the procedures below, the string *<interface\_directory>* should be replaced by the appropriate directory containing the interface data.

### 3.1 What Do You Need?

The following versions of the Siemens A&D AS MES products have to be installed correctly before the rest of the installation can be executed.

- SIMATIC IT Unilab V5.0 SP2 (or SP3) with 21CFR11 enabled
- SIMATIC IT Interspec V5.1.

### 3.2 Before Starting

Because you can choose the administrator, it is not possible to include this hard-coded in the scripts. Therefore you should make some modifications before you can start the setup.

Do not forget to make the scripts writable before you try to modify them !

- Replace the string **@DBANAME@** by the local Interspec DBA name in the following directories:
  - *<interface\_directory>*\Creation\speCX DB
  - *<interface\_directory>*\Upgrade\speCX DB
- Replace in the following directories the string **@DBANAME@** by the local Unilab DBA name:
  - *<interface\_directory>*\Creation\Unilab DB
  - *<interface\_directory>*\Upgrade\Unilab DB

### 3.3 Installation Procedure for the Unilab Database

To be able to execute the following procedure, make sure your location is the `<interface_directory>\Creation\Unilab DB` directory.

#### 3.3.1 Enabling the Database Jobs

After a standard installation of Unilab, you still have to make an Interspec-specific job available, because this job is not automatically available when the Unilab database is installed.

Adapt the stored procedures **StartAllDbJobs** and **StopAllDbJobs** of package **cxapp**: remove the comment indicators in the lines where the Interspec job is handled. You need to start one job to automatically create the group key structures when necessary.

#### 3.3.2 Installing the Packages

Proceed as follows to install the packages:

1. Log on as administrator in the Unilab database with sqlplus.
2. Run the script to stop all jobs:

```
SQL> BEGIN cxapp.StopAllDbJobs; END;
```

3. Run the script to install the packages.

```
SQL> @install_pack.sql;
```

Explicitly ignore **ORA-01432: public synonym to be dropped does not exist** errors. The script will perform its job.

The package **pa\_specxinterface** will be compiled.

4. In order to solve Bug 5452, the Unilab package undiff had to be modified. This modification will be included in standard Unilab from Unilab V6.1, but for earlier versions the package is supplied with the interface installation, and has to be compiled explicitly.

```
SQL> @..\..\undiff.sql;
```

### 3.3.3 Checking the Status of the Packages

After a well-executed installation, all packages should be valid. To be absolutely safe, an extra check will be executed, so all packages that are marked as invalid by Oracle will be recompiled.

Proceed as follows to check the status of the packages:

1. Compile the invalid packages, if there are some:

```
SQL> @unilab_compinv.sql;
```

2. In case there actually are some invalid packages, the script should be executed a second time to make sure that the first execution has made them valid:

```
SQL> @unilab_compinv.sql;
```

The output of this second execution should contain **no invalid packages**.

You may get the error message **ORA-01555: snapshot too old: rollback segment number <string> with name "<string>" too small** while running the script. This is due to internal Oracle procedures. Just restart the script until no more error messages are displayed.

### 3.3.4 Starting the Database Jobs

Proceed as follows to start the database jobs:

1. Log on as administrator to the Unilab database with sqlplus.
2. Run the script to start all jobs:

```
SQL> BEGIN cxapp.StartAllDbJobs; END;
```

### 3.3.5 Installing the Views

Proceed as follows to install the views:

1. Log on as administrator to the Unilab database with sqlplus.
2. Run the script to install the necessary upgrades:

```
SQL> @install_vw.sql;
```

### 3.3.6 Setting the Database Parameters

Proceed as follows to set the database parameters:

1. Log on as administrator to the Unilab database with sqlplus.
2. Execute following command to set database parameter **remote\_dependencies\_mode**:

```
SQL> alter system set remote_dependencies_mode = signature;
```

### 3.3.7 Creating the Templates

Template objects can be created in Unilab to copy the value of some standard attributes of the transferred sample types, parameter profiles, parameter definitions and method definitions:

Proceed as follows to create the templates:

1. Create a sample type.
2. Create a parameter profile.
3. Create a parameter definition.
4. Create a method definition.

Make sure these template objects have a current version. For more information on the templates in Interspec, see [Configuration of the Templates](#).

### 3.3.8 Adapting the Lifecycle [Optional]

After being transferred to Unilab, the object status is **In Editing**, not **Approved**. The Unilab user has to manually approve the object. This behavior is not always desired, because sometimes a lot of data is transferred. The system lifecycle can be adapted in order to immediately approve transferred objects. An exception on this rule is the generic specification. If a sample type, coming from a generic specification is automatically set to **Approved**, a new minor version of the sample type will be created every time a linked specification gets transferred.

It is possible in Unilab to create new minor versions of parameter profiles and sample types. If a new version is transferred from Interspec, it is possible to automatically create new versions of these derived parameter profiles and sample types. Normally, these are created by a difference mechanism triggered by an event rule (for more info, cfr. the documentation of Enterprise LIMS).

The list of possible transitions from status @~ should look like this:

@~	->	@A:	condition <b>NewObjFromInterspc</b> action <b>SynchronizeAll</b>
@~	->	@E:	condition <b>NewGenStFromInterspc</b> action <b>SynchronizeAll</b>
@~	->	@E:	no condition no action

If also the new minor versions created in Unilab by the action **SynchronizeAll** should be changed to **Approved**, then the user has to modify the package **undiff** accordingly.

### 3.4 Upgrade Procedure for the Unilab Database

This procedure is meant for an upgrade from interface V4.03.

To be able to execute the following procedure, make sure your location is the `<interface_directory>\Upgrade\Unilab DB directory`.

#### 3.4.1 Upgrading the Packages

Proceed as follows to upgrade the packages:

1. Log on as administrator to the Unilab database with sqlplus.
2. Run the script to stop all jobs:

```
SQL> BEGIN cxapp.StopAllDbJobs; END;
```

3. Run the script to install the packages.

```
SQL> @upgrade_pack.sql;
```

Explicitly ignore **ORA-01432: public synonym to be dropped does not exist** errors. The script will perform its job.

The package **pa\_specxinterface** will be compiled.

4. In order to solve Bug 5452, the Unilab package undiff had to be modified. This modification will be included in standard Unilab from Unilab V6.1, but for earlier versions the package is supplied with the interface installation, and has to be compiled explicitly.

```
SQL> @..\..\undiff.sql;
```

#### 3.4.2 Checking the Status of the Packages

After a well-executed upgrade, all packages should be valid. To be absolutely safe, an extra check will be executed, so all packages that are marked as invalid by Oracle will be recompiled.

Proceed as follows to check the status of the packages:

1. Compile the invalid packages, if there are some:

```
SQL> @unilab_compinv.sql;
```

2. In case there actually are some invalid packages, the script should be executed a second time to make sure that the first execution has made them valid:

```
SQL> @unilab_compinv.sql;
```

The output of this second execution should contain **no invalid packages**.

You may get the error message **ORA-01555: snapshot too old: rollback segment number <string> with name "<string>" too small** while running the script. This is Oracle-related. Just restart the script until no more error messages are displayed.

### 3.4.3 Starting the Database Jobs

Proceed as follows to start the database jobs:

1. Log on as administrator to the Unilab database with sqlplus.
2. Run the script to start all jobs:

```
SQL> BEGIN cxapp.StartAllDbJobs; END;
```

### 3.4.4 Upgrading the Other Objects

Proceed as follows to upgrade the other objects:

1. Log on as administrator to the Unilab database with sqlplus.
2. Run the script to install the necessary upgrades:

```
SQL> @upgrade_objects.sql;
```

## 3.5 Installation Procedure for the Interspec Database

To be able to execute following procedure, make sure your location is the `<interface_directory>\Creation\speCX DB` directory.

### 3.5.1 Installing the Database Link

Before the database link and the tables can be installed, check whether it is possible to connect to the Unilab databases from the Interspec database server. If it is not possible, modify the file **tnsnames.ora** on the Interspec database server. Add a new entry for every Unilab database.

Proceed as follows to install the database link:

1. Log on as administrator to the Interspec database with sqlplus.
2. Modify the script **install\_dblink.sql**. Change the following items:
  - `<user>`: Unilab administrator
  - `<pswd>`: corresponding password
  - `<connectionstring>`: connection string of the Unilab database
  - `<dbname>`: name of the Unilab database, that consists of `<servername>_<sid>`, eg. `nnvh023a_utqc`. This name has to match the string that is configured for the plants (cfr. [Configuration of the Plants](#))
  - Create a link for all Unilab databases for which specifications must be exported. (Copy the statement in **install\_dblink.sql** and modify the corresponding fields)

3. Run the script to install the database link:

```
SQL> @install_dblink.sql;
```

4. Check whether the database link is created correctly:

```
SQL> SELECT name, host, userid, password
FROM sys.link$
WHERE name LIKE '%LNK_LIMS%';
```

This example shows how to test database link **LNK\_LIMS**. Execute analogous statements for **LNK\_LIMS\_H** and the links to the Unilab databases.

5. If the creation was not successful, check whether the parameter **global\_names** is false. If true, you cannot choose the name of the database link

```
SQL> SELECT name, value
FROM v$parameter
WHERE name = 'global_names';
```

6. Check whether it is possible to execute a command that uses the database link:

```
SQL> SELECT SYSDATE
FROM dual@LNK_LIMS;
```

Again, this example shows how to test database link **LNK\_LIMS**. Execute analogous statements for **LNK\_LIMS\_H** and the links to the Unilab databases.

### 3.5.2 Installing the Grants

Proceed as follows to install the grants:

1. Log on as user **sys** to the Interspec database with sqlplus.
2. Run the script to stop the specserver:

```
SQL> @stopserv.sql;
```

3. Run the script:

```
SQL> @install_grant.sql;
```

4. Run the script to start the specserver:

```
SQL> @strtserv.sql;
```

### 3.5.3 Installing the Add-on Packages

Proceed as follows to install the add-on packages:

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to install the add-on packages:

```
SQL> @install_packaddon.sql;
```

Explicitly ignore **ORA-01432: public synonym to be dropped does not exist** errors. The script will perform its job. The packages **pa\_terror** and **pa\_tools** will be compiled, and their public synonyms will be created.

### 3.5.4 Installing the Tables, Views, Sequences and Initial Data

Proceed as follows to install the tables, views, sequences and initial data:

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to install the tables, the views, the sequences and the initial data:

```
SQL> @install_tblvwseqdata.sql;
```

Explicitly ignore **ORA-00942: table or view does not exist**, **ORA-01432: public synonym to be dropped does not exist**, and **ORA-02289: sequence does not exist** errors. The script will perform its job.

The following table gives an overview of the database objects that are installed:

Object type	Name
table	itlimsplant
	itlimstmp
	itlimsiob
	itlimsseq
	itlimsconflv
	itlimsconfkw
view	ivpapr
	ivlimsiob
sequence	limspa seq
	limssc seq
	limsssc seq

### 3.5.5 Installing the Packages and Triggers

Proceed as follows to install the packages and triggers:

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to install the packages and the triggers.

```
SQL> @install_packtrg.sql;
```

Explicitly ignore **ORA-01432: public synonym to be dropped does not exist** errors. The script will perform its job.



The following table gives an overview of the database objects that are installed:

Object type	Name
package	pa_lims
	pa_limscfg
	pa_limsspc
	pa_limsspc2
	pa_liminterface
trigger	tr_lims_plant
	tr_lims_pa
	tr_lims_sc
	tr_lims_soc
	tr_lims_ssc

To guarantee the consistency of the database, some extra scripts have to execute the trigger actions on the existing data.

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to insert data into table **itlimsjob**:

```
SQL> @install_trgdata_job.sql;
```

3. Run the script to insert data into table **itlimsseq**:

```
SQL> @install_trgdata_seq.sql;
```

### 3.5.6 Checking the Status of the Packages

After a well-executed installation, all packages should be valid. To be absolutely safe, an extra check will be executed, so all packages that are marked as invalid by Oracle will be recompiled.

Proceed as follows to check the status of the packages:

1. Compile the invalid packages, if there are some:

```
SQL> @interspec_compinv.sql;
```

2. In case there actually are some invalid packages, the script should be executed a second time to make sure that the first execution has made them valid:

```
SQL> @interspec_compinv.sql;
```

The output of this second execution should contain **no invalid packages**.

You may get the error message **ORA-01555: snapshot too old: rollback segment number <string> with name "<string>" too small** while running the script. This is due to internal Oracle procedures. Just restart the script until no more error messages are displayed.

### 3.5.7 Starting the Database Jobs for the Interface

Two database jobs are necessary and a third job is optional.

The first mandatory job, **pa\_limsspc.f\_transferallhistobs**, executes the necessary actions when a specification in Interspec becomes **Historic** or **Obsolete**. The second mandatory job, **pa\_terror.p\_errorlogging**, takes care of the error logging.

The optional job, **pa\_limsinterface.p\_transfercfgandspc**, will automatically transfer all objects if the user does not want to transfer the configuration objects and the specifications manually. If this behavior is not desired, the job **pa\_limsinterface.p\_transfercfgandspc** has to be stopped manually afterwards.

#### Execute the installation procedure

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to start the jobs:

```
SQL> @interspec_start_jobs.sql;
```

3. Check whether the jobs are installed:

```
SQL> SELECT what
      FROM dba_jobs
      WHERE what IN ('pa_limsinterface.p_transfercfgandspc;',
                    'pa_limsspc.f_transferallhistobs;',
                    'pa_terror.p_errorlogging;');
```

#### Updating the interval of the jobs

The script below can change the interval of the jobs running on the Interspec database.

Proceed as follows to update the interval of the jobs:

1. Check whether it is necessary to change the interval:

```
SQL> SELECT what, interval
      FROM dba_jobs
      WHERE what IN ('pa_limsinterface.p_transfercfgandspc;',
                    'pa_limsspc.f_transferallhistobs;',
                    'pa_terror.p_errorlogging;');
```

2. Modify the script **interspec\_update\_jobs.sql**. Change **<NEW INTERVAL>** to a new interval for the job to run.
3. Run the update script:

```
SQL> @interspec_update_jobs;
```

### 3.5.8 Modifying the Configuration

When the interface is configured to support linked specifications (linked as attached specifications in interspec) then you have to execute the following steps.

#### Check list for the configuration of preference KW ID Generic Spc

1. Check the value of preference **KW ID Generic Spc** (via menu **Other > Preferences**), to see whether a key word exists that indicates which specifications are generic.
2. If a key word ID is found, then the key word can be found through:

```
SQL> SELECT kw_id, description
      FROM itkw
      WHERE kw_ID = <key word ID>;
```

Replace *<key word ID>* by the returned value of the previous query.

The value of the column **description** is the key word used in Interspec to indicate generic specifications.

If there is no key word, you can execute the following procedure to configure one.

#### Modifying the preference KW ID Generic Spc

Proceed as follows to modify the configuration of **KW ID Generic Spc**:

1. Create a key word that indicates a generic specification e.g. **Generic specification**.
2. Get the ID of the specific key word:

```
SQL> SELECT kw_id
      FROM itkw
      WHERE description = 'Generic specification';
```

The value of the column **kw\_ID** is the ID of the key word.

3. Modify the preference **KW ID Generic Spc**.

#### Modifying the preferences for the pp\_key paths

Insert into the **itlimsppkey** table the preferences for the paths. For each Unilab database, 5 lines should be inserted, one for each possible Unilab pp\_key.

Following statement can be used to insert the records. The *<db\_link\_name>* has to be replaced by the name of the correct database link, and *<x>* by the correct pp\_key index. To define the correct value for *<path>*, see [Configuration of the Preference Settings](#).

```
SQL> INSERT INTO itlimsppkey(db, pp_key_seq, path)
      VALUES ('<db_link_name>', '<x>', '<path>');
SQL> COMMIT;
```

### 3.5.9 Installing the LIMS Language

The IDs of the Unilab parameter profiles, parameter and method definitions are the descriptions in language **LIMS** of the Interspec objects. Following steps have to be executed:

Create a language **LIMS** via the Interspec configuration module.

## 3.6 Upgrade Procedure for the Interspec Database

This procedure is meant for an upgrade from interface V4.03.

To be able to execute following procedure, make sure your location is the `<interface_directory>\Upgrade\speCX DB` directory.

### 3.6.1 Stopping the Database Jobs for the Interface

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to start the jobs:

```
SQL> @interspec_stop_jobs.sql;
```

3. Check if the jobs are stopped. The following query should not return any row:

```
SQL> SELECT what
      FROM dba_jobs
      WHERE what IN ('pa_limsInterface.p_TransferCfgAndSpc;',
                    'PA_LIMSSPC.f_TransferAllHistObs;',
                    'pa_iterror.p_ErrorLogging;');
```

### 3.6.2 Installing the Add-on Packages

Proceed as follows to upgrade the add-on packages:

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to upgrade the add-on packages:

```
SQL> @upgrade_packaddon.sql;
```

Explicitly ignore **ORA-01432: public synonym to be dropped does not exist** errors. The script will perform its job.

The packages **pa\_iterror** and **pa\_tools** will be compiled.

### 3.6.3 Upgrading the Packages

Proceed as follows to upgrade the packages:

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to install the packages.

```
SQL> @upgrade_packtrg.sql;
```

Explicitly ignore **ORA-01432: public synonym to be dropped does not exist** errors. The script will perform its job.

The following database objects will be compiled:

Object type	Name
package	pa_lims
	pa_limscfg
	pa_limsspc
	pa_limsspc2
	pa_limsinterface
trigger	tr_lims_plant
	tr_lims_pa
	tr_lims_sc
	tr_lims_spc
	tr_lims_ssc

### 3.6.4 Checking the Status of the Packages

After a well-executed upgrade, all packages should be valid. To be absolutely safe, an extra check will be executed, so all packages that are marked as invalid by Oracle will be recompiled.

Proceed as follows to check the status of the packages:

1. Compile the invalid packages, if there are some:

```
SQL> @interspec_compinv.sql;
```

2. In case there actually are some invalid packages, the script should be executed a second time to make sure that the first execution has made them valid:

```
SQL> @interspec_compinv.sql;
```

The output of this second execution should contain **no invalid packages**.

You may get the error message **ORA-01555: snapshot too old: rollback segment number <string> with name "<string>" too small** while running the script. This is due to internal Oracle procedures. Just restart the script until no more error messages are displayed.

### 3.6.5 Starting the Database Jobs for the Interface

#### Execute the installation procedure

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to start the jobs:

```
SQL> @interspec_start_jobs.sql;
```

3. Check whether the jobs are installed:

```
SQL> SELECT what
       FROM dba_jobs
       WHERE what IN ('pa_limsinterface.p_transfercfgandspc;',
                     'pa_limsspc.f_transferallhistobs;',
                     'pa_iterror.p_errorlogging;');
```

#### Updating the interval of the jobs

If on the installation of the interface, the interval of the jobs was adapted, this modification has to be executed again.

Proceed as follows to update the interval of the jobs:

1. Check whether it is necessary to change the interval:

```
SQL> SELECT what, interval
       FROM dba_jobs
       WHERE what IN ('pa_limsinterface.p_transfercfgandspc;',
                     'pa_limsspc.f_transferallhistobs;',
                     'pa_iterror.p_errorlogging;');
```

2. Modify the script **interspec\_update\_jobs.sql**. Change **<NEW INTERVAL>** to a new interval for the job to run.
3. Run the update script:

```
SQL> @interspec_update_jobs;
```

### 3.6.6 Upgrading the Other Objects

Proceed as follows to upgrade the other objects:

1. Log on as administrator to the Interspec database with sqlplus.
2. Run the script to install the necessary upgrades:

```
SQL> @upgrade_objects.sql;
```