Chapter 12

The Telelogic Tau Public Interface

This chapter introduces the Telelogic Tau Public Interface and describes a message based interface by which external tools and applications could be integrated with the Telelogic Tau tools.

Three conceptually different facilities are provided:

- By controlling the behavior of the Telelogic Tau tools and by modifying the data the Telelogic Tau tools handle.
- By listening to important events of status change in the Telelogic Tau tools.
- By integrating an external application with SDL simulators.

The Telelogic Tau Public Interface uses the PostMaster as transport. This chapter assumes the reader to be familiar with the PostMaster; see <u>chapter 11, *The PostMaster*</u>.

In *chapter 13, Using the Telelogic Tau Public Interface*, you may find examples of how to use the Public Interface.

General Concepts

Introduction

Significant steps have been made to increase the openness of the Telelogic Tau tools by the introduction of the *Telelogic Tau Public Interface*. Openness is provided both in terms of diverse ways of controlling the Telelogic Tau tools and the way data handled by the Telelogic Tau tools are made visible.

Application Areas

Two different kinds of usage of the Public Interface could be foreseen:

- As an alternate or complementary means to access functionality provided by the Telelogic Tau tools, compared to the normal graphical interface. Such usage could be to integrate simulators generated by the Telelogic Tau tools with dedicated user interfaces, or to use the Telelogic Tau tools batch facilities using make or print in a large "build" environment.
- Integrating the Telelogic Tau tools with other tools forming a larger environment.

The Public Interface

The public interface to the Telelogic Tau tools has been made possible due to the modularity of the Telelogic Tau tools and thanks to the well defined interfaces by which the Telelogic Tau tools has been built.

The public interface is made available to the Telelogic Tau tools users using two different mechanisms.

• Via a programming interface, providing message based services and notifications. The rest of this section provides a description of these services and notifications and their usage.

- Via applications executed via the OS command line interface. By using this interface:
 - Important services could be executed running the Telelogic Tau tools in a *batch* mode.
 - A Service Encapsulator could be run, which allows message based services to be issued. This tool is also made available as a an example application. For more information, see <u>"The Service Encapsulator" on page 537</u>.

The PostMaster

The PostMaster forms an integral mechanism to integrate tools in the Telelogic Tau tools environment (a reference to the PostMaster is provided in <u>chapter 11, *The PostMaster*</u>).

It is also used as a low level mechanism to physically integrate the Telelogic Tau tools with external tools. The PostMaster provides three basic facilities:

- It maintains a list on known tools. Such tools could be connected to the PostMaster and thereby taking part of message sending
- It provides means to broadcast messages to a list of subscribers
- It contains functionality to send a message to a certain recipient.

These facilities are used to implement the concepts provided by the Telelogic Tau tools services. These are built up of <u>Services</u> and <u>Notifications</u>. A special case of service is <u>Communication with SDL Simulators</u>.

It is important to note that the interface provided could be extended for client-client usage outside the Telelogic Tau tools as long as the conventions described in this document are followed.

Services

Provides access to functionality within the Telelogic Tau tools. Such functions could be used by external tool to control the Telelogic Tau tools, to integrate the Telelogic Tau tools with external tools or have the Telelogic Tau tools take part in a larger environment.

Services adopts the client-server concept where a *request* is sent by a client to a server, which returns with a *reply*. Service request and service reply take both advantage of message sending from one tool to another.

Notifications

These are broadcast messages which are spontaneously emitted when a significant event takes place in the Telelogic Tau tools. Often these notifications are caused by a service being successfully processed.

Communication with SDL Simulators

When external applications are to communicate with SDL Simulators, the message SESDLSIGNAL is used. It is asynchronously sent/broadcaster into the system. All simulators subscribes on this message. As a parameter to this message is the receiver (in the context of SDL) provided.

Overview of Available Services

The following services are provided by the Telelogic Tau tools for external usage. A service is normally supported by a specific tool or a few number of tools within the Telelogic Tau tools environment. Some of the services correspond to a menu choice or an operation performed by a user using the tools's graphical interface.

In the following services, the <u>Configuration Services</u> and the <u>System</u> <u>File Services</u> are applicable to Telelogic Tau, and the <u>ITEX Services</u> are only applicable to the TTCN suite. All other services are only applicable to the SDL suite.

Service	Servers	Graphical correspondence
Start Tool	PostMaster	Start a new tool
Stop Tool	All tools	Exit menu choice
Get Tool Type	PostMaster	N/A
Get Tool Pid	PostMaster	N/A
Add Tool	PostMaster	N/A
Add Tool Subscription	PostMaster	N/A

Configuration Services

System File Services

Service	Servers	Graphical correspondence
List System Files	Organizer	N/A
New System	Organizer	New quick button
Open System	Organizer	Open quick button
Save System	Organizer	Save quick button
Add Existing	Organizer	Add Existing menu choice

Link File Services

Service	Servers	Graphical correspondence
Add Local Link File	Organizer	N/A
Merge Local Link File	Organizer	N/A

ITEX Services

Service	Servers	Graphical correspondence
Convert to GR	ITEX	N/A
Opened Documents	ITEX	N/A
Fetch Buffer Identifier Given the Database	ITEX	N/A
Fetch Buffer Identifier Given MP File Path	ITEX	N/A
Convert to MP	ITEX	N/A
Convert Selection to MP	ITEX	N/A
Merge Document	ITEX	N/A
Analyze Document	ITEX	N/A
Close Document	ITEX	N/A
Save Document	ITEX	N/A
Selector	ITEX	N/A
SelectAll	ITEX	N/A
DeselectAll	ITEX	N/A
IsSelected	ITEX	N/A
Get Modify Time	ITEX	N/A
Get Path	ITEX	N/A
Get MP Path	ITEX	N/A
Find Table	ITEX	N/A
Close Table	ITEX	N/A

Service	Servers	Graphical correspondence
Get Table State	ITEX	N/A
Get Row Number	ITEX	N/A
Select Row	ITEX	N/A
Clear Selection	ITEX	N/A
Rows Selected	ITEX	N/A

Menu Manipulation Services

Service	Servers	Graphical corre- spondence
Add Menu	Organizer SDLE MSCE OME TE Coverage Viewer Index Viewer Type Viewer Tree Viewer File Viewer Preference Manager SimUI/ValUI	N/A
Delete Menu	As for Add Menu above.	N/A
Clear Menu	As above.	N/A
Add Item to Menu	As above.	N/A
Add Item to Menu – Organizer	Organizer	N/A
Add Item to Menu <u>– Text Editor</u>	TE	N/A
Add Item to Menu <u>– Graphical Edi-</u> <u>tors</u>	SDLE MSCE OME	N/A

Logging Services

Service	Servers	Graphical correspondence
Start MSC Log	PostMaster	N/A
Stop MSC Log	PostMaster	N/A

SDT Reference Services

Service	Servers	Graphical correspon- dence
Show Source	Organizer	Show source menu choice
Obtain GR Reference	SDLE MSCE OME	N/A

Editor – Diagram Services

Service	Servers	Graphical correspondence
Load Diagram	SDLE MSCE OME TE	Open menu choice
Unload Diagram	SDLE MSCE OME TE	Close menu choice
Show Diagram	SDLE MSCE OME TE	Diagrams menu
Save Diagram	SDLE MSCE OME TE	Save menu choice
Create SDL Diagram	SDLE	New menu choice

Service	Servers	Graphical correspondence
Create MSC Diagram	MSCE	New menu choice
Create OM Diagram	OME	New menu choice
Create Text Diagram	TE	New menu choice

Editor – Object Services

Service	Servers	Graphical correspondence
Select Object	SDLE MSCE OME	Selecting an object
Show Object	SDLE MSCE OME	Selecting an object
Insert SDL Object	SDLE	Select object in symbol menu
Insert MSC Object	MSCE	Select object in symbol menu
Remove Object	SDLE MSCE OME	Clear menu choice
Get Object Text	SDLE MSCE OME	N/A

Editor – Object Attribute Services

Service	Servers	Graphical correspondence
<u>Display Key</u>	SDLE MSCE OME	N/A
List Key	SDLE MSCE OME	N/A

Service	Servers	Graphical correspondence
Create Attribute	SDLE MSCE OME	N/A
Update Attribute	SDLE MSCE OME	N/A
Read Attribute	SDLE MSCE OME	N/A
Delete Attribute	SDLE MSCE OME	N/A

Information Server Services

Service	Servers	Graphical correspondence
Load Definition File	Information Server	N/A

SDL Editor Services

Service	Servers	Graphical correspondence
GRPR	SDLE	N/A
<u>Tidy Up</u>	SDLE	Tidy Up menu choice

SC Editor Services

Service	Servers	Graphical correspondence
Get Diagram Info	OME	N/A

MSC Editor Services

Service	Servers	Graphical correspondence
MSC GRPR	MSCE	Generate MSC PR menu choice

HMSC Editor Services

Service	Servers	Graphical correspondence
HMSC GRPR	MSCE	Generate MSC PR menu choice

CIF Services

Service	Servers	Graphical correspondence
Create SDL Diagram	SDLE	N/A
Create SDL Page	SDLE	N/A
Insert SDL Object	SDLE	N/A
Create OM Diagram	OME	N/A
Create OM Page	OME	N/A
Insert OM Object	OME	N/A

Text Editor Services

Service	Servers	Graphical correspondence
Show Position	TE	N/A
Select Text	ТЕ	N/A

Client Interface

A client which would like to make an integration using these facilities should consult the following interface:

- A programming interface to communicate with the PostMaster
- A description of the provided services
- A description of the available notifications

External Client types

Clients connecting to the PostMaster must be known by the PostMaster. This can be accomplished in a number of ways:

1. By defining a configuration file containing the extended configuration, name this file to post.cfd and to set the environment variable POSTPATH to include the directory where the file is stored.

A change in this file is kind of static nature, since the configuration files are read every time the PostMaster is started.

2. By modifying the configuration dynamically. Requires a PostMaster to be running. The PostMaster provides services for dynamically changing a configuration, adding tools and adding subscriptions. This is easily done via a script and does only affect the current session.

New tool types or message types added to the PostMaster tool and subscription list should be set in a range not conflicting with Telelogic Tau tools. New tools and messages should use values above 100000. Exact numbering should follow the scheme used by the Telelogic Tau tools:

Item	Description
Tools	Use values in steps of 1000
Service Request	Base the value on the tool providing the service and add local base of 100. Then sequentially number the services
Service Reply	Base the value on the corresponding service request and add 100.
Notifications	Base the value on the tool broadcasting the noti- fication and add the messages sequentially.

Message Based Services

Introduction

Using a message based service requires the client to be connected to the PostMaster. To invoke a service, the application sends a service request message to the tool providing the service. A service request message will normally cause a reply message to be sent to the client. An exception is if the server unexpectedly terminates while processing the service.

The client could choose whether to wait for the reply message (emulating a remote procedure call) or to continue working and intercept the reply message via a callback routine.

The client does not need to subscribe on service request messages or service reply messages. However the client itself must be present in the configuration list.

A server can only process one service at time. If additional services are requested from that particular server when being occupied processing the first service, the server is said to be *busy* and a service error reply is returned to the requester. Other kinds of messages received by server during the processing of a service request are queued up and will be processed as soon as the service has completed.

Service Request

A *service request* normally takes advantage of the <u>SPSendToTool</u> function. If more than one instance of a tool is active the function <u>SPSendToPid</u> will be more appropriate. The reply is fetched with the <u>SPRead</u> function. The client must however verify that the read message is the reply message. The macro SP_MATCHREPLY could be used to determine whether the received message is the desired reply.

Service Reply

The first parameter in the *service reply* informs about the result of the service request. The following codes are used:

Status Code	Value	Description
ОК	0	The service was successfully processed. Optional parameters are provided in the re- ply message.
Busy	1	The server is <i>busy</i> and cannot process the service. The service request is aborted. An additional message might be provided.
ErrorString	2	The server failed to process the service. The remaining parts of the message contain an error message.
ErrorCode	3	The server failed to process the service, the next parameter in the service reply is a code indicating the error.

In the detailed description of each service, the reply format is only specified for the normal case returning <u>OK</u>. For services replying errors in the <u>ErrorString</u> format, the text may be context sensitive and only major error causes are specified.

Error Handling

Error handling takes place at two different levels.

1. The service request failed to be issued.

Service Request message couldn't be sent. In such case the <u>SPSendToTool</u>, <u>SPSendToPid</u> returns false and the variable sperrno indicates the error. It also means that a service reply message will not be sent.

2. While processing the service.

In this case the service request reaches the server but cannot be processed or the processing of the service fails. In such a case, an error code is provided in the service reply message. The error is either provided as a code or as explanatory text.

How to decode the service reply is described in the section below.

Common Errors

These errors are common to all services. They all use the <u>ErrorString</u> format. Where applicable, an additional explanatory text is added to the strings below:

Bad parameters Server busy Service not supported Server locked

SDT Reference Errors

When a service request refers to an SDT reference, the following errors may occur:

```
Reference must contain paranthesis
Reference must start with #SDTREF
Invalid reference
Garbage after reference
Embedded value not terminated
Illegal value in paranthesis after token <token-nr>
Junk after embedded value
Incomplete reference lacking trailing paranthesis
Unsupported reference type <token>
Symbol must be integer >= 0
Malformed coordinate
```

For a reference to the syntax of references, see <u>chapter 19, SDT Refer</u>ences.

Notifications

Certain services will broadcast notifications as the service is processed in order to inform other participants than the service issuer that an operation has been performed by the server or that the state of the server has changed.

Message Parameters

Normally a message (a service request, a service reply, a notification or other) uses one or more parameters. Generally all such parameters are stored in a PostMaster message's data part.

- Parameters are normally separated by a blank (' ') character.
- A parameter that has an additional '*' character is a shorthand for none or a number of this type, each one separated by a blank ' ' or a newline '\n'.
 - The character '+' might also appear as an alternative. The difference is that one or more is expected.
- A preceding attribute tells how many items to expect.

The following parameter types are used:

Туре	Description
integer	32 bit integer in ASCII form.
bool	Logical. True = 1, False = 0
string	ASCII string. If the string contains one or more spaces, it is surrounded by double quotes (i.e. "The string"). If a quote character appears in the string, it is preceded by a backslash ('\'). A backslash character is doubled ('\\'). An empty string is also double quoted ("").
QString	ASCII string surrounded by double quotes (i.e. "The string"). If a quote character appears in the string it is preceded by a backslash. ('\').
ByteString	Byte string. Is always preceded by an attribute telling the length of the byte string.

Files

- The files sdt.h and itex.h provides two lists of the available services and notifications in terms of message definitions. It also defines a list (see <u>"Adding Tools and Messages" on page 492 in chapter 11, *The PostMaster*) providing a textual definition of messages.
 </u>
- The file sdtsym.h provides the definitions necessary when working with editors and defines values for diagram types, pages types and symbol types.
- In Windows, the file post.dll is the DLL that contains the external PostMaster interface. It needs either reside in the same directory as the application that is utilizing it, or in a directory that is in the current PATH.

The files are found in the directory \$sdtdir/INCLUDE (on UNIX), or \$SDTDIR%\include (in Windows).

Interpretation of a Service Description

Below is an explanation of the different sections found in the service descriptions in <u>"Tool Services" on page 539</u>.

Description

A brief textual description of the service.

Tools Supporting the Service

The tool type(s) providing the service. Corresponds to definitions in the file sdt.h/itex.h; see <u>"Files" on page 535</u>.

Service Request

Message name. Corresponds to a definition in the file sdt.h/itex.h; see <u>"Files" on page 535</u>.

The service request is presented in a table of service parameters, with the appearance below. The parameters must appear in the order they are listed in the table. Usually, a parameter cannot be omitted. If it is optional, this is explicitly mentioned. See also <u>"Message Parameters" on page 534</u>.

Parameter	Туре	Description
symbolic parameter name	parameter type	Brief description of the parameters.

Service Reply

Message name. Corresponds to a definition in the file sdt.h/itex.h; see <u>"Files" on page 535</u>.

The table below is similar to the service request parameter table above. A service reply always contains a status code in the first parameter. Additional parameters are only valid if the status code was OK (0). See also <u>"Message Parameters" on page 534</u>.

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Errors additional to the common errors are listed in this section; see "Common Errors" on page 533.

Emitted Notifications

Notifications emitted as a result of the service being successfully processed; see <u>"Notifications" on page 522</u>.

The Service Encapsulator

The Service Encapsulator application enables access to Telelogic Tau tools services from the Operating System command line prompt.

Basically the tool connects to the Postmaster, sends a service request message and waits for an answer. When the answer arrives, it is printed on standard output. Finally the application exists.

The Service Encapsulator is also available in source code form to show how the Postmaster's <u>Functional Interface</u> could be used. A description of the internal design of the Service Encapsulator is found in <u>chapter 13</u>, <u>Using the Telelogic Tau Public Interface</u>. The source code of the Service Encapsulator is found in

```
$telelogic/examples/public_interface (on UNIX)
```

<installation directory>\examples\public_interface
(in Windows)

The Service Encapsulator binary is invoked by:

On UNIX: \$telelogic/bin/serverpc <tool> <service><params>

In Windows: <installation directory>\sdtbin\serverpc
<tool> <service> <params>

where <tool> is the tool that should perform the service, and <service> is the service itself.

The <tool> and <service> arguments could either be entered as a symbolic value or as the assigned integer value. These definitions are found in sdt.h

If the service takes parameters, these should be provided in <params>

Care should be taken in order to enter parameters correctly. In particular if the service uses quoted string parameters:

- **On UNIX**, surrounding quotes should be doubled since the shell from which the tool is invoked, consumes one pair of quotes.
- In Windows, surrounding quotes should be prefixed with a backslash (\) since the "DOS" shell from which the tool is invoked, consumes the quotes otherwise.

The tool allows carriage return "n" and line-feed "r" to be used.

The tool does not allow <params> to contain binary data. Therefore, the SDL suite services accepting binary data must only contain ASCII characters.

The tool returns 0 on success and -1 if an error occur. Such errors correspond to errors when calling Postmaster functions, see <u>SPInit</u>, <u>SPSendToTool</u>, <u>SPRead</u> in <u>Functional Interface</u> for possible errors.

Tool Services

Configuration Services

Start Tool

Description

This service will start the specified tool. **On UNIX**, start means that a new UNIX process is "fork'ed" and "exec'ed". **In Windows**, start means that the applications is started using the Windows API function CreateProcess.

The Service adds "-post" to the argv[1] variable.

The parameters are recognized by the started tool in the argv[] variable starting from index 2.

The started tool inherits the environment used by the PostMaster.

The service behaves somewhat differently from other services since it is performed both in the PostMaster and in the PostMaster library in the started tool. The service is initiated in the PostMaster creating the new tool, but the service is recognized to be completed when the started tool calls the PostMaster function <u>SPInit</u>. This call causes a SESTARTNOTIFY message to be sent. This message is recognized by the PostMaster which upon reception of the SESTARTNOTIFY also sends a SESTARTREPLY to the issues of the start service.

The service is also different in its nature since it will cause a time-out to expire if the started tool does not call the <u>SPInit</u> function within a certain time limit.

The started tool must have an entry in the PostMaster configuration and the associated file containing the tool to start must exist.

The service will return to the caller:

- When the SESTARTREPLY message is received by the caller.
- If the tool is only available in one instance and is already instantiated. In this case the process id of the already instantiated tool is returned.

Chapter 12 The Telelogic Tau Public Interface

• If the tool cannot be started or if the tool is not recognized by the PostMaster within a time-out. (The started tool calls <u>SPInit</u>.)

The following MSC diagram shows the protocol when a tool is started normally (the Organizer starts the editor).



Figure 164: Protocol to start a tool

The SESYNCCONNECT and the SESYNCCONNECTREPLY messages are not present in the **Windows** implementation.

Tools Supporting the Service

SET_POST

Service Request

SESTART

Parameter	Туре	Description
toolType	integer	Tool type to be started
params	string	Optional. Parameters for the started tool. More than one parameter allowed. If separat- ed with blanks, each item will be insert- ed into the argv[] string list for the started application. The PostMaster will add a parameter -post as the first pa- rameter in the argv[] list.

Service Reply

SESTARTREPLY

Parameter	Туре	Description
status	integer	Service reply status.
pId	integer	A PId identifying the tool type of which the corresponding type is started.

Errors

```
No such tool
Max instances of tool
Already started
File not found
+ Operating System file access and process creation
error messages
```

Emitted Notifications

Notification	Description
SESTARTNOTIFY	Broadcast by the started application when calling <u>SPInit</u> .

Stop Tool

Description

The tool disconnects from the Postmaster and then terminates.

Tools Supporting the Service

All tools

Service Request

SESTOP

Parameter	Туре	Description
force	bool	If <i>false</i> , the tool is allowed to reject the request. If <i>true</i> , the tool is expected to terminate

Service Reply

SESTOPREPLY

Parameter	Туре	Description
status	integer	Service reply status
cancelled	bool	<i>True</i> if the tool did not accept the Stop request.

Errors

_

Telelogic Tau 4.5 User's Manual

Get Tool Type

Description

Returns the process type for a given process id.

Tools Supporting the Service

SET_POST

Service Request

SEGETTOOLTYPE

Parameter	Туре	Description
pId	integer	A PId identifying the tool type of which the corresponding type is wanted.

Service Reply

SEGETTOOLTYPEREPLY

Parameter	Туре	Description
status	integer	Service reply status.
noOfToolType	integer	Number of tools corresponding to the PId value. If the requested tool type was not found in the configura- tion, noOfToolType is set to 0.
toolType	integer	The type of tool. If the requested PId is not found, toolType 0 is returned

Errors

_

July 2003

Get Tool Pid

Description

Returns the process id for the requested tool type. If multiple instances of the tool type exist, all PId values are returned.

Tools Supporting the Service

SET_POST

Service Request

SEGETTOOLPID

Parameter	Туре	Description
toolType	integer	An identifier for the tool type of which the corresponding PIds) are wanted.

Service Reply

SEGETTOOLPIDREPLY

Parameter	Туре	Description
status	integer	Service reply status.
noOfPid	integer	Number of PId values corresponding to the toolType. If the requested tool type was not found in the configuration, noOfPid is set to 0.
pId	integer	A PId value corresponding to toolType

Errors

_

Add Tool

Description

Dynamically adds a tool type to the PostMaster configuration.

Tools Supporting the Service

SET_POST

Service Request

SEADDTOOL

Parameter	Туре	Description
toolType	integer	An identifier for the tool. Should be a value within the allowed range.
filename	string	The filename of the executable tool. Should be a pure filename and must not include a directory path. The file should be stored on a directory that is included in the directories designated by the envi- ronment variable POSTPATH.

Service Reply

SEADDTOOLREPLY

Parameter	Туре	Description
status	integer	Service reply status.
exist	bool	Returns false if the tool was inserted, true if it already existed.

Errors

Operation failed

Add Tool Subscription

Description

Dynamically adds a message to a tool's subscription list.

Tools Supporting the Service

SET_POST

Service Request

SEADDTOOLSUBSCRIPTION

Parameter	Туре	Description
tooltype	integer	The tool type to which a message sub- scription should be added.
message	integer	The message to add to the subscription list.

Service Reply

SEADDTOOLSUBSCRIPTIONREPLY

Parameter	Туре	Description
status	integer	Service reply status.
exist	bool	Returns false if the message was in- serted, true if it already existed.

Errors

No such tool Operation failed

System File Services

List System Files

Description

Lists the files currently held in the Diagram Structure chapter in the Organizer. The files will be returned in the order they are found in the Organizer (i.e. top-down, left-right).

Tools Supporting the Service

SET_ORGANIZER

Service Request

SELISTSYSTEMFILES

Parameter	Туре	Description
-	-	N/A.

Service Reply

SELISTSYSTEMFILESREPLY

Parameter	Туре	Description
status	integer	Service reply status.
noOfFiles	integer	Number of returned files.
file(s)	string	A list of files, with a complete directory path. Each file specification is ended by a newline.

Errors

_

New System

Description

Clear the Organizer content and create a new system. Corresponds to the Organizer menu choice <u>New</u>.

Tools Supporting the Service

SET ORGANIZER

Service Request

SENEWSYSTEM

Parameter	Туре	Description
forceQuit	bool	If <i>true</i> , quit modified diagrams. If <i>false</i> and there were one or more modified diagrams, the service is denied.

Service Reply

SENEWSYSTEMREPLY

Parameter	Туре	Description
status	integer	Service reply status.
cancelled	bool	True, if the service was cancelled.

Errors

Service request denied. No license available.

Open System

Description

Opens a system file and displays the file contents in the Organizer. Corresponds to the Organizer menu command <u>Open</u>.

Tools Supporting the Service

SET_ORGANIZER

Service Request

SEOPENSYSTEM

Parameter	Туре	Description
filename	string	The system file to open.
forceQuit	bool	If true, quit modified diagrams. If false and modified diagram(s) exist(s), the service is denied.

Service Reply

SEOPENREPLY

Parameter	Туре	Description
status	integer	Service reply status.
toolType	integer	The type of tool. If the requested PId is not found, toolType 0 is returned.

Errors

Service request denied. No license available Cannot open, system modified Error opening system A message box displays the error

Save System

Description

Corresponds to the Organizer menu command <u>Save</u>. Unconnected diagrams are saved in default file names; for more information, see <u>"Save</u> in file" on page 62 in chapter 2, *The Organizer*.

Tools Supporting the Service

SET ORGANIZER

Service Request

SESAVEALL

Parameter	Туре	Description
systemStructureOnly	bool	True, both the system files and diagram files are saved; False, only diagram files are saved.

Service Reply

SESAVEALLREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Service request denied. No license available Error save all A message box displays the error

Add Existing

Description

Adds an existing document to the Organizer and optionally displays it in an editor. The type of document to be added and its corresponding editor depends on the extension of the given filename. This is the same mechanism as in the GUI-based equivalent. Corresponds to the Organizer menu choice <u>Add Existing</u>.

Tools Supporting the Service

SET_ORGANIZER

Service Request

SEADDEXISTING

Parameter	Туре	Description
filename	string	The document to add in the Organizer.
selected Filename	string	If there is a document connected to this file in the Organizer, the document will be selected before the <i>Add Existing</i> opera- tion. In this case, the added document will be inserted before the selected document. An empty string means no selection.
start Editor	bool	If <i>true</i> , the added document will be popped up in an editor.
expand Substruct ure	bool	If <i>true</i> , the substructure diagrams to the added SDL diagram are added as well.

Service Reply

SEADDEXISTINGREPLY

Parameter	Туре	Description
status	integer	Service reply status.
ok	bool	Returns <i>true</i> if the operation could be performed.

Errors

Chapter 12 The Telelogic Tau Public Interface

Link File Services

Add Local Link File

Description

Prepares a personal link file for a user. Any changes in endpoint and link information after this service is called are saved into this file. The master link file can be read only for the user at this time.

Tools Supporting the Service

SET_ORGANIZER

Service Request

SEADDLOCALLINKFILE

Parameter	Туре	Description
filename	string	File name.

Service Reply

SEADDLOCALLINKFILEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Telelogic Tau 4.5 User's Manual

Merge Local Link File

Description

Merges the changes saved into the personal link file into the master link file. After a successful merge, the local link file information is cleared.

Tools Supporting the Service

SET_ORGANIZER

Service Request

SEMERGELOCALLINKFILE

Parameter	Туре	Description
-	-	N/A.

Service Reply

SEMERGELOCALLINKFILEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

-

ITEX Services

Convert to GR

Description

Converts a MP file to TTCN-GR. One document file is generated in the specified destination directory. The document filename is generated by ITEX. The name of the destination directory is given. The parameter *ignore page number* indicates that any included page numbers in the MP file will be ignored.

The generated names are displayed in the Organizer log. For more information, see <u>"Importing a TTCN-MP Document" on page 1149 in</u> chapter 25, *The TTCN Browser (on UNIX)*.

Tools Supporting the Service

IET_BASE

Service Request

IEBXCONVERTTOGR

Parameter	Туре	Description
filename	QString	The name of the MP file
destination	QString	The name of the destination directory
ignore page number	bool	Ignoring the included page numbers

Service Reply

IEBXCONVERTTOGRREPLY

Parameter	Туре	Description
status	integer	Service reply status
itexfile	QString	The filename of the generated document.

Errors

The MP file does not exist Illegal MP file The file has no read permission Unable to generate document files in the destination
Opened Documents

Description

Retrieves information about open documents known to ITEX. The EBNF follows:

```
DocInfoList ::= empty | DocInfoList `<NewLine>'
DocInfo
DocInfo ::= documentidentifier `:' buffid `:'
itexfile `:' backupfile
```

Tools Supporting the Service

IET_BASE

Service Request

IEBXOPENEDDOCUMENTS

Service Reply

IEBXOPENEDDOCUMENTSREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the oper- ation does not fail).
infolist	EBNF	The list of open documents identifiers, buffer identifiers and database file.

Errors

Fetch Buffer Identifier Given the Database

Description

Given a database, this function fetches the document buffer identifier if the document is open.

Tools Supporting the Service

IET_BASE

Service Request

IEBXGETBUFFIDFROMPATH

Parameter	Туре	Description
itexfile	QString	The database path in the local system syntax.

Service Reply

IEBXGETBUFFIDFROMPATHREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the oper- ation does not fail).
buffid	integer	The document's buffer identifier

Errors

Unable to find database

Fetch Buffer Identifier Given MP File Path

Description

Given a MP file path this function fetches the document buffer identifier if there is any database generated from this MP and the document in the database is open.

Tools Supporting the Service

IET BASE

Service Request

IEBXGETBUFFIDFROMMPPATH

Parameter	Туре	Description
mpfile	QString	The MP file path in the local system syntax.

Service Reply

IEBXGETBUFFIDFROMMPPATHREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the oper- ation does not fail).
buffid	integer	The document's buffer identifier

Errors

Unable to find database

Convert to MP

Description

Generates a MP file for the given system node. This converts the entire document source (all TTCN objects) and does not consider the selection. The document must be connected but not necessarily open.

For more information, see <u>"Exporting a TTCN Document to TTCN-MP"</u> on page 1144 in chapter 25, *The TTCN Browser (on UNIX)*.

Tools Supporting the Service

IET_BASE

Service Request

IEBXCONVERTTOMP

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
standard flag	bool	Indicates if the MP shall be TTCN stan- dard. (True if standard MP is required.)
filename	QString	The MP filename

Service Reply

IEBXCONVERTTOMPREPLY

Parameter	Туре	Description
status	integer	Service reply status

Errors

Invalid buffer identifier The document is not connected Unable to write file

Convert Selection to MP

Description

Generates a MP file for the given system node. It converts only the selected parts of the document which means that the document needs both to be connected and open.

For more information, see <u>"Exporting a TTCN Document to TTCN-MP" on page 1144 in chapter 25, *The TTCN Browser (on UNIX)*.</u>

Tools Supporting the Service

IET BASE

Service Request

IEBXCONVERTSELTOMP

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
standard flag	bool	Indicates if the MP shall be TTCN stan- dard. (True if standard MP is required.)
filename	QString	The MP filename

Service Reply

IEBXCONVERTSELTOMPREPLY

Parameter	Туре	Description
status	integer	Service reply status

Errors

Invalid buffer identifier The document is not connected The document is not open Unable to write file Chapter 12 The Telelogic Tau Public Interface

Merge Document

Description

This function is used to merge one document into another document. Both the source and the destination documents must be available to the ITEX environment, i.e. both documents must be loaded. Furthermore a selection must exists in the source document.

For more information, see <u>"Merging TTCN Documents" on page 1137</u> in chapter 25, *The TTCN Browser (on UNIX)*.

Tools Supporting the Service

IET BASE

Service Request

IEBXMERGEDOCUMENT

Parameter	Туре	Description
srcbuffid	integer	The source document's buffer iden- tifier
destbuffid	integer	The destination document's buffer identifier

Service Reply

IEBXMERGEDOCUMENTREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

Unable to merge documents

Analyze Document

Description

Analyze the given system node.

For more information, see <u>chapter 27</u>, <u>Analyzing TTCN Documents (on</u> <u>UNIX</u>), or <u>chapter 32</u>, <u>Analyzing TTCN Documents (in Windows)</u>.

Tools Supporting the Service

IET_BASE

Service Request

IEBXANALYZE

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
forced analysis	bool	Set if forced analysis should be in effect
verbose	bool	Indicates if verbose mode is to be on
errorlimit	integer	Max number of errors before aborting

Service Reply

IEBXANALYZEREPLY

Parameter	Туре	Description
status	integer	Service reply status

Errors

Invalid buffer identifier The document is not connected

Close Document

Description

This function closes the document and all open tables in it.

Tools Supporting the Service

IET_BASE

Service Request

IEBXCLOSEDOCUMENT

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXCLOSEDOCUMENTREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

Invalid buffer identifier Unable to close document

Save Document

Description

Given the buffer identifier of the document the corresponding system node is saved in the given filename.

Tools Supporting the Service

IET_BASE

Service Request

IEBXSAVE

Parameter	Туре	Description
buffid	integer	The document reference
filename	QString	The filename where the document source must be saved.

Service Reply

IEBXSAVEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

No such tool Invalid buffer identifier Couldn't write file

Selector

Description

Given restrictions and a database this function selects all objects which fulfill the restrictions. For more information, see <u>"Using More Complex</u> Selections" on page 1117 in chapter 25, *The TTCN Browser (on UNIX)*.

Tools Supporting the Service

IET_BASE

Service Request

IEBXSELECTOR

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
namerestr	string	The name restriction.
typerestr	string	The content restriction.
contentrestr	string	The type restriction.
selectormode	string	The selector mode (restrict, ex- tend or replace REL where REL is references, references_recursive or referenced_by)
analysestatus	string	The analysis status (not_analyzed, error_analyzed or ok_analyzed)

Service Reply

IEBXSELECTORREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

```
No such tool
Invalid buffer identifier
```

SelectAll

Description

Select all objects in the document.

Tools Supporting the Service

IET_BASE

Service Request

IEBXSELECTALL

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXSELECTALLREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

Chapter 12 The Telelogic Tau Public Interface

DeselectAll

Description

Remove all selections in the document.

Tools Supporting the Service

IET_BASE

Service Request

IEBXDESELECTALL

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXDESELECTALLREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

IsSelected

Description

This function checks if there is any object selected in the document.

Tools Supporting the Service

IET_BASE

Service Request

IEBXISSELECTED

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXISSELECTEDREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
selectstate	bool	Indicates if there is any selected objects in the document.

Errors

Get Modify Time

Description

This function fetches the modify time of a document.

Tools Supporting the Service

IET_BASE

Service Request

IEBXGETDOCUMENTMODIFYTIME

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXGETDOCUMENTMODIFYTIMEREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
modifytime	QString	The modify time of the document.

Errors

Get Path

Description

Given a document buffer identifier this function fetches the corresponding database path.

Tools Supporting the Service

IET_BASE

Service Request

IEBXGETPATH

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXGETPATHREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
itexfile	QString	The database path in the local system syntax.

Errors

Chapter 12 The Telelogic Tau Public Interface

Get MP Path

Description

Given a document bufferid this function fetches the corresponding MP file path.

Tools Supporting the Service

IET_BASE

Service Request

IEBXGETMPPATH

Parameter	Туре	Description
buffid	integer	The document's buffer identifier

Service Reply

IEBXGETMPPATHREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
mpfile	QString	The MP file path in the local system syn- tax.

Errors

Find Table

Description

Given the buffer identifier of the selected document and a table identifier, ITEX searches for the table. The found table is displayed in the Table Editor.

Tools Supporting the Service

IET BASE

Service Request

IEBXFINDTABLE

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableid	QString	The name of the table to find

Service Reply

IEBXFINDTABLEREPLY

Parameter	Туре	Description
status	integer	Service reply status

Errors

Invalid buffer identifier Unconnected document

Chapter 12 The Telelogic Tau Public Interface

Close Table

Description

Close the given table.

Tools Supporting the Service

IET_BASE

Service Request

IEBXCLOSETABLE

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableident	QString	The table identifier.

Service Reply

IEBXCLOSETABLEREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

```
Invalid buffer identifier
No such object
```

Get Table State

Description

This function returns the status of a given table. The status a table indicates if the table is open or close.

Tools Supporting the Service

IET_BASE

Service Request

IEBXGETTABLESTATE

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableident	QString	The table identifier.

Service Reply

IEBXGETTABLESTATEREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
tablestate	QString	Returns open, close, row_in_open or row_in_close.

Errors

Invalid buffer identifier No such object

Get Row Number

Description

Given the name of a row this function returns the number (position) of it in the table. The row can be a row in a single or multiple table.

Tools Supporting the Service

IET_BASE

Service Request

IEBXGETROWNUMBER

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableident	QString	The table identifier.
rowname	identifier	The name of the row.

Service Reply

IEBXGETROWNUMBERREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
rownumber	integer	The number of the given row. The first row has number one.

Errors

Invalid buffer identifier No such object No such row

Select Row

Description

This function modifies the selection status of a given row in a table.

Tools Supporting the Service

IET_BASE

Service Request

IEBXSELECTROW

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableident	QString	The table identifier.
rownumber	integer	The number of the row.
selectstate	bool	The select status of the row to be modified.

Service Reply

IEBXSELECTROWREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
exist	bool	Returns false if the message was in- serted, true if it already existed.

Errors

Invalid buffer identifier No such object No such rownumber

Clear Selection

Description

This function removes all row selections in a table.

Tools Supporting the Service

IET_BASE

Service Request

IEBXCLEARSELECTION

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableident	QString	The table identifier.

Service Reply

IEBXCLEARSELECTIONREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).

Errors

```
Invalid buffer identifier
No such object
```

Rows Selected

Description

Given the buffer identifier and the table identifier, this function returns the numbers of the selected rows in a table (the first row has number 1).

Tools Supporting the Service

IET_BASE

Service Request

IEBXROWSSELECTED

Parameter	Туре	Description
buffid	integer	The document's buffer identifier
tableident	QString	The table identifier.

Service Reply

IEBXROWSSELECTEDREPLY

Parameter	Туре	Description
status	integer	Service reply status (zero if the opera- tion does not fail).
nonezero*	integer	A list of selected row numbers. The list is empty if no row is selected.

Errors

Invalid buffer identifier No such object

Menu Manipulation Services

Introduction

All graphical tools in the SDL suite support customizable menus. These user-defined menus will be appended to the menu bar of the tool. The exact location will be defined by the tool, depending on the abilities of the graphical framework the tool is built upon.

The intention is to have an external tool to configure a tool in order to provide the necessary UI. Telelogic Tau tools could also take advantage of these extendable menus.

The following operations are available through a set of well defined PostMaster messages:

- Creating a menu
- Creating a menu choice
- Deleting a menu

Each menu choice can be associated to any of the following:

- A PostMaster message
- An operating system command

Add Menu

Description

Adds a new menu to the menu bar.

Tools Supporting the Service

SET_ORGANIZER SET_SDLE SET_MSCE SET_OME SET_TE SET_SIMULATOR_UI¹ SET_FILEVIEWER SET_COVERAGEVIEWER SET_XREFVIEWER SET_TYPEVIEWER SET_TREEVIEWER SET_PREFERENCES SET_HELPVIEWER

Service Request

SEMENUADD

Parameter	Туре	Description
menuName	string	Name of the menu to add.

Service Reply

SEMENUADDREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

-

^{1.} In Windows, the Simulator/Validator UI's can not read menu-definition files.

Delete Menu

Description

Removes the menu and its menu choices from the menu bar.

Tools Supporting the Service

SET_ORGANIZER SET_SDLE SET_MSCE SET_OME SET_TE SET_SIMULATOR_UI¹ SET_FILEVIEWER SET_COVERAGEVIEWER SET_XREFVIEWER SET_TYPEVIEWER SET_TREEVIEWER SET_PREFERENCES SET_HELPVIEWER

Service Request

SEMENUDELETE

Parameter	Туре	Description
menuName	string	Name of the menu to delete.

Service Reply

SEMENUDELETEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

_

^{1.} In Windows, the Simulator/Validator UI's can not read menu-definition files.

Clear Menu

Description

Clears the menu bar from a menu item.

Tools Supporting the Service

SET_ORGANIZER SET_SDLE SET_MSCE SET_OME SET_TE SET_SIMULATOR_UI¹ SET_FILEVIEWER SET_COVERAGEVIEWER SET_XREFVIEWER SET_TYPEVIEWER SET_TREEVIEWER SET_PREFERENCES SET_HELPVIEWER

Service Request

SEMENUCLEAR

Parameter	Туре	Description
menuName	string	Name of the menu to clear.

Service Reply

SEMENUCLEARREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

_

^{1.} In Windows, the Simulator/Validator UI's can not read menu-definition files.

Chapter 12 The Telelogic Tau Public Interface

Add Item to Menu

Description

Adds a menu choice to the specified menu. The menu choice could either perform an OS command or issue a PostMaster notification when selected. The OS command to perform or the message to broadcast could be sensitive on a selected symbol.

The description of the service parameters below is generic to all tools supporting the service. Some tools have special interpretations of some parameters. Some tools also allow *format codes* to be used in the command string or as message parameter, providing additional context sensitive information. Both these tool-specific issues are described separately in the following sections:

- <u>"Add Item to Menu Organizer" on page 586</u>
- "Add Item to Menu Text Editor" on page 590
- "Add Item to Menu Graphical Editors" on page 591

Tools Supporting the Service

SET_ORGANIZER SET_SDLE SET_MSCE SET_OME SET_TE SET_SIMULATOR_UI¹ SET_FILEVIEWER SET_COVERAGEVIEWER SET_XREFVIEWER SET_TYPEVIEWER SET_TREEVIEWER SET_PREFERENCES SET_HELPVIEWER

^{1.} In Windows, the Simulator/Validator UI's can not read menu-definition files.

Service Request

SEMENUADDITEM

Parameter	Туре	Description
menuName	string	The menu in which an item should be added.
menuItem	string	Name of menu item to add.
separator	bool	If a separator should precede the item.
statusText	string	The status text to show when the menu item is selected.
notUsed1 lastAction ProprietaryKey (tool-specific)	integer	The interpretation of this pa- rameter is tool-specific; see the separate tool descriptions later. If not used, this parameter should be set to 0.
notUsed2 AttributeKey (tool-specific)	integer	The interpretation of this pa- rameter is tool-specific; see the separate tool descriptions later. If not used, this parameter should be set to 0.
scope	integer	Indicates when the menu item should be dimmed. The possi- ble values are tool-specific; see the separate tool descrip- tions later. If not used, this parameter should be set to:
		ALWAYS (0) The menu choice will al- ways be available, inde- pendently of the selection in the tool's active win- dow.

Parameter	Туре	Description
confirmText	string	If no text is provided, no con- firmation is assumed. A non- empty text denotes confirma- tion; a two button dialog will be issued with the choices <i>OK</i> and <i>Cancel</i> . The dialog text is defined in confirmText. In an associated user editable field, the expanded action to perform is displayed.
actionType	integer	The value controls the last part of the message which is vari- ant. 0 = PostMaster message 1 = OS command.

• If actionType above sets the variant part to be a **PostMaster message**, the last two parameters are:

Parameter	Туре	Description
message	integer	Interpreted as the PostMaster message to send.
params	string	Interpreted as the parameters to the PostMaster message. Tool-specific context sensitive format codes are eval- uated; see the separate tool descriptions later.

• If actionType above sets the variant part to be an **OS command**, the last two parameters are:

Parameter	Туре	Description
OSblock	bool	Whether to wait for the command to fin- ish before giving control to the user.

Parameter	Туре	Description
OScommand	string	The OS command to perform. Tool-spe- cific context sensitive format codes are evaluated; see the separate tool descrip- tions later.

Service Reply SEMENUADDITEMREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

-

Add Item to Menu – Organizer

Description

Identical to the description of <u>Add Item to Menu</u>, except that the semantics of the parameters and supported format codes for the service differ.

Tools Supporting the Service

SET_ORGANIZER

Service Request

SEMENUADDITEM

Format Codes

The following format codes are recognized. There are format modifiers for some of the basic formats:

- B the base filename
- D the directory part of the full filename
- R the filename, relative to the *source directory*

Format code	Description
%b %Bb, %Db %Rb	The file name of the system file associated to the Or- ganizer window.
%f %Bf %Df %Rf	The name of the file that contains the selected object.
%F %BF %DF %RF	All files in the substructure of the selected object (including the selected object).
%1 %Bl %Dl %Rl	The name of the link file loaded in the Organizer.
%0 %B0 %D0 %R0	The name of the Control Unit file for the selected object.

Format code	Description
%r	Perform the command recursively on all files in the substructure of the selected object (including the selected object).
%u	The source directory.
۶v	The target directory.

Parameters

Apart from providing a set of format codes, the Organizer gives special interpretation to the lastAction and scope parameters.

Parameter	Туре	Description
lastAction	integer	Controls what happens when a dynamic menu command has been performed. Only avail- able when actionType is OS Command. This attribute should be one of the following:
		NOTHING (0) Perform no action.
		CHECK FILE (1) A check file operation is performed on the selected object when the OS com- mand is completed. If the OS command is non- blocking, the command is performed as soon as the OS command has been is- sued.
		CHECK_FILE_ON_RECURSIVE (2) This flag works as the CHECK_FILE flag but is used only with a %r com- mand.

Parameter	Туре	Description
		CM_UPDATE (4) Perform a CM_UPDATE operation on the closest *.scu file.
		SHOW_LOG (8) Show the Organizer Log window (where textual output from the OS com- mand is presented).

Parameter	Туре	Description
scope	integer	This attribute could be one of the following:
		ALWAYS (0) The menu choice will al- ways be available, inde- pendently of the selection in the tool's active win- dow.
		ONE SELECTED OBJECT (1) The menu choice is avail- able if one object is select- ed.
		SELECTED_OBJECT_NOT_IN_ EDITOR (4) The menu choice is avail- able if one object is select- ed and the selected object is not loaded in an editor.
		VALID_SELECTED_OBJECT (5) The menu choice is avail- able if one object is select- ed and the object is not marked invalid.
		SELECTED_GROUP_NOT_IN_E DITOR (6) The menu choice is avail- able if one object is select- ed and no diagram of the document group for the selected object is loaded in an editor.

Chapter 12 The Telelogic Tau Public Interface

Add Item to Menu – Text Editor

Description

Identical to the description of <u>Add Item to Menu</u>, except that the semantics of the parameters and supported format codes for the service differ.

Tools Supporting the Service

SET_TE

Service Request

SEMENUADDITEM

Format Codes

The following format codes are recognized.

Format code	Description
%f	The filename of the document.
₩u	The source directory
ទំន	The selected text
۶S	The entire text of the document.
Add Item to Menu – Graphical Editors

Description

Identical to the description of <u>Add Item to Menu</u>, except that the semantic of the parameters and supported format codes for the service differ.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEMENUADDITEM

Format Codes

The following format codes are recognized.

Format code	Description
%a	The absolute name of the file associated with the se- lected object. Extracted from the SDT reference.
%b	The absolute name of the file associated to the win- dow.
%C	If the selected object uses extended data, the comment is extracted.
%d	If the selected object uses extended data, the data part is extracted.
%e	The text in the object. (Only in the SDL Editor.)
%f	The name of the file that contains the object.
%g	The SDT reference to the object.
\$p	The page name currently shown in the window (not applicable to the MSC Editor).
%S	The name of the file shown in the window.
%t	The page name for the object (not applicable to the MSC Editor). Extracted from the SDT reference.

Chapter 12 The Telelogic Tau Public Interface

Parameters

Apart from providing a set of format codes, the graphical editors give special interpretation to the lastAction, ProprietaryKey and AttributeKey parameters.

Parameter	Туре	Description
ProprietaryKey	integer	The keys ProprietaryKey and AttributeKey are used to determine whether or not a menu choice should be avail- able (i.e. dimmed or not). See <u>"Editor – Object Services" on</u> <u>page 608</u> for more informa- tion.
AttributeKey	integer	See description of ProprietaryKey.

Parameter	Туре	Description
scope	integer	This attribute should be one of the following:
		ALWAYS (0) The menu choice will al- ways be available, inde- pendently of the selection in the tool's active win- dow. The keys ProprietaryKey and AttributeKey are han- dled as don't care.
		ONE_SELECTED_OBJECT ^a (1) The menu choice is avail- able only if exactly one object is selected. The keys ProprietaryKey and AttributeKey are handled as don't care.
		MATCHING_KEYS (2) The menu choice is avail- able only if at least one of the selected objects has an attribute that matches the keys above.
		MATCHING_KEYS_ONE_SELEC TED_OBJECT (3) ^{a_} The menu choice is avail- able only if exactly one object is selected and it has an attribute that matches the keys above.

a. Each tool should define and adopt conventions for when exactly one object only is considered as selected. For instance, selecting a task symbol in an SDL Editor also selects the from and to lines. However, attaching information to these lines does not fill any meaningful purpose; the SDL Editor considers the situation as if one object only (i.e. the task symbol) was selected.

Logging Services

Start MSC Log

Description

Starts the logging of messages sent by tools connected to the PostMaster. The format used by the log is event-oriented MSC/PR.

Tools Supporting the Service

SET_POST

Service Request

SESTARTTRACE

Parameter	Туре	Description
filename	string	The name of the file on which the log should be stored. If the parameter is omitted, the default log file post.mpr will be used. If the parameter is set to "- e", logging is done on standard error.
logMode	integer	If not set, only public messages are logged. If set to a value > 2 all messages are logged.

Service Reply

SESTARTTRACEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Stop MSC Log

Description

Stops the logging of messages sent by PostMaster tools.

Tools Supporting the Service

SET_POST

Service Request

SESTOPTRACE

Parameter	Туре	Description
-	-	N/A.

Service Reply

SESTOPTRACEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

-

July 2003

SDT Reference Services

Show Source

Description

Selects the SDT reference in an editor. A reference could be:

- An SDL reference. The reference is shown in an SDL Editor.
- An MSC reference. The reference is shown in an MSC Editor.
- An OM reference.

The reference is shown in an OM Editor.

• A text reference

The reference is shown in a text editor. The text editor is chosen by the preference SDT*<u>TextEditor</u>.

For a complete description of the format of an SDT reference, please see <u>chapter 19</u>, *SDT References*.

Tools Supporting the Service

SET_ORGANIZER

Service Request

SESHOWREF

Parameter	Туре	Description
SDTRef	string	A valid SDT reference.

Service Reply

SESHOWREFREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

SDT Reference Errors

Emitted Notifications

Notification	Description
SELOADNOTIFY	If any editor loaded the required diagram.
SESDLELOADNOTIFY	If the SDL Editor loaded the required dia- gram.
SEOMELOADNOTIFY	If the OM editor loaded the required diagram.
SESTARTNOTIFY	If the editor was started.

Obtain GR Reference

Description

Returns the SDT references for the current selection(s) in the editors.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEOBTAINGRREF

Parameter	Туре	Description
-	-	N/A

Service Reply

SEOBTAINGRREFREPLY

Parameter	Туре	Description
status	integer	Service reply status.
NoOfRef	integer	The number of references found.
ref*	string	A complete SDT reference.

Errors

```
<u>SDT Reference Errors</u>
Illegal use of qualifier
Illegal reference type
```

Editor – Diagram Services

The Buffer Concept

The editor defines the concept buffer, which basically identifies a diagram currently loaded in the editor. Each buffer is identified with a *buffer id* which is unique within one session of the editor as long as the editor is not stopped)

A buffer id is returned when an existing diagram is successfully loaded in an editor or a new diagram is created and implicitly loaded in the editor. Then, most services manipulating diagrams in editors, refer to the buffer containing the diagram via the buffer id.

Load Diagram

Description

Loads a diagram or text document specified by filename in an editor buffer. If the diagram was already loaded, the existing buffer id will be returned.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME SET_TE

Service Request

SELOAD

Parameter	Туре	Description
filename	string	The diagram file to load specified with the full directory path.

Service Reply

SELOADREPLY

Parameter	Туре	Description
status	integer	Service reply status.
bufId	integer	Refers to an allocated buffer in the editor.
type	integer	The type of diagram.
name	string	The name of the loaded diagram.

Errors

```
Can not read file <filename> <error message>
SDLE is busy, syntax error in text (SDLE only)
```

Notification	Description
SELOADNOTIFY	If the diagram was loaded.
SESDLELOADNOTIFY	If an SDL diagram was loaded.
SEOMELOADNOTIFY	If an OM, SC or HMSC diagram was loaded.

Unload Diagram

Description

Unloads a diagram from an editor.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME SET_TE

Service Request

SEUNLOAD

Parameter	Туре	Description
bufId	integer	Refers to a buffer in the editor.
forceUnload	bool	If true, the editor will force a modified diagram to be unloaded. If false, the editor will not unload the diagram if it is modified.

Service Reply

SEUNLOADREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid diagram buffer id
Diagram is changed If force Unload is false
SDLE is busy, syntax error in text (SDLE only)
```

Notification	Description
SEUNLOADNOTIFY	If the diagram was unloaded.

Show Diagram

Description

The service will raise a window in the editor showing the specified buffer.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME SET_TE

Service Request

SESHOW

Parameter	Туре	Description
bufId	integer	Refers to a buffer in the editor.
pageName	string	Name of the page to show. If the string is empty, the last recently used, or if no such page exist, the de- fault page will be shown. Applicable all diagrams except MSC and text documents. For MSC diagrams this parameter may be omitted or left empty. For text document (loaded in the TE), this option is ignored.

Service Reply

SESHOWREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid diagram buffer id
Unable to open page (SDLE/OME only)
SDLE is busy, syntax error in text (SDLE only)
```

Notification	Description
SESHOWNOTIFY	If the diagram was raised.

Save Diagram

Description

The service will save the diagram in the specified file. If not a valid filename or if there is no permission to save the file, an error will be returned. When saved, the editor buffer is marked as not edited.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME SET_TE

Service Request

SESAVE

Parameter	Туре	Description
bufId	integer	Refers to a buffer in the editor.
filename	string	The file where to save the buffer.

Service Reply

SESAVEREPLY

Parameter	Туре	Description
status	integer	Service reply status.
saveok	bool	Returns true if the buffer was successfully saved.

Errors

```
Invalid diagram buffer Id
Diagram is new. Filename is missing.
Cannot save file <filename> <error message>
SDLE is busy, syntax error in text (SDLE only)
```

Notification	Description
SESAVENOTIFY	If the diagram was saved.

Create SDL Diagram

Description

Creates an empty SDL diagram in a new buffer. The diagram gets an unconnected status. Corresponds to SDL Editor command *New*.

Tools Supporting the Service

SET_SDLE

Service Request

SESDLECREATEDIAGRAM

Parameter	Туре	Description
virtuality	integer	Kind of virtuality. Only applicable to typed diagrams. Other diagrams should use the value NOVIRTUAL Possible values are: NOVIRTUAL VIRTUAL REDEFINED FINALIZED These values are defined in sdtsym.h
DiagramType	integer	Diagram type. See sdtsym.h for valid diagram types.
name	string	Diagram name.
qualifier	string	SDL qualifier. Could be empty.
радеТуре	integer	Type of the first page in the dia- gram. See sdtsym.h for valid page types. Note that the pageType must correspond with the diagramType.
pageName	string	Name of the first page in the dia- gram.

Chapter 12 The Telelogic Tau Public Interface

Service Reply

SESDLECREATEDIAGRAMREPLY

Parameter	Туре	Description
status	integer	Service reply status.
bufId	integer	Refers to a buffer in the editor.

Errors

SDLE is busy, syntax error in text

Notification	Description
SESDLECREATENOTIFY	As a result of creating the diagram.
SEPAGENOTIFY	As a result of creating a page.

Create MSC Diagram

Description

Creates an empty MSC diagram in new buffer. Corresponds to MSC Editor command *New*.

Tools Supporting the Service

SET_MSCE

Service Request

SEMSCECREATEDIAGRAM

Parameter	Туре	Description
diagramType	integer	Diagram type. Defined in sdtsym.h. The value MSCDIAGRAM (16) should be used.
name	string	Name of the diagram to create.
qualifier	string	For future use. Should be empty.

Service Reply

SEMSCECREATEDIAGRAMREPLY

Parameter	Туре	Description
status	integer	Service reply status.
bufId	integer	Refers to a buffer in the editor.

Errors

-

Notification	Description
SEMSCENEWNOTIFY	As a result of creating the diagram.

Create OM Diagram

Description

Creates an empty OM diagram in new buffer. Corresponds to OM Editor command *New*.

Tools Supporting the Service

SET_OME

Service Request

SEOMECREATEDIAGRAM

Parameter	Туре	Description
diagramType	integer	Diagram type. Defined in sdtsym.h. The value CLASSDIAGRAM (21) should be used.
name	string	Name of the diagram to create.
радеТуре	integer	Page type Defined in sdtsym.h. The value CLASSPAGE (16) should be used
pageName	string	name of the first page

Service Reply

SEOMECREATEDIAGRAMREPLY

Parameter	Туре	Description
status	integer	Service reply status.
bufId	integer	Refers to a buffer in the editor.

Errors

_

Notification	Description
SEOMENEWNOTIFY	As a result of creating the diagram.

Create Text Diagram

Description

Creates an empty text diagram in new buffer. Corresponds to Text Editor command *New*.

Tools Supporting the Service

SET_TE

Service Request

SETECREATEDIAGRAM

Parameter	Туре	Description
diagramType	integer	Diagram type. See sdtsym.h for valid diagram types.
name	string	Name of the diagram to create.

Service Reply

SETECREATEREPLY

Parameter	Туре	Description
status	integer	Service reply status.
bufId	integer	Refers to a buffer in the editor.

Errors

-

Notification	Description
SETENEWNOTIFY	As a result of creating the diagram.

Chapter 12 The Telelogic Tau Public Interface

Editor – Object Services

Select Object

Description

The *Select Object* service will highlight an object in a diagram. Note that this command does not show the buffer and the selected object in a window. The difference between this service and the <u>Obtain GR Refer</u><u>ence</u> service is that the <u>Select Object</u> service does not require the diagram to be saved on a file.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SESELECTOBJECT

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which an object should be shown.
objectid	integer	Identifier to the object to show.
row	integer	The row where to put the text cursor.
column	integer	The column where to put the text cursor in the row.
keepSelections	bool	Flag indicating if old selections should be kept.

Service Reply

SESELECTOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid diagram buffer id
Invalid object id
SDLE is busy, syntax error in text (SDLE only)
```

Show Object

Description

The *Show Object* service will make sure that the specified object is visible in a window. This means that it will display the buffer in a window and if necessary scroll into the position where the object is. Often used in combination with <u>Select Object</u>.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SESHOWOBJECT

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which an object should be shown.
objectId	integer	Identifier to the object to show.

Service Reply

SESHOWOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Invalid diagram buffer id Invalid object id SDLE is busy, syntax error in text *(SDLE only)*

Insert SDL Object

Description

Adds an object to the SDL diagram.

Tools Supporting the Service

SET_SDLE

Service Request

SESDLEINSERTOBJECT

Parameter	Туре	Description
bufId	integer	Buffer in which to insert the object.
pageName	string	Page to insert the object on.
shiftIsDow n	bool	Emulate behavior of having <shift> pressed when inserting an symbol.</shift>
objectType	integer	Type of object. Valid objects depend on the type of diagram and if syntax check- ing is on. Corresponds to available sym- bols in the editor symbol menu. NOTE: SDL reference symbols are not possible to add using this service . Definitions of symbols are found in sdtsym.h.
objectText	string	Text in object.

Service Reply

SESDLEINSERTOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Invalid diagram buffer id Invalid object id A reference symbol containing text\ cannot be inserted in diagram <diagram> Object type <objectType> is not allowed\ in diagram <diagram> Unable to open page The page is too small to insert the object SDLE is busy, syntax error in text

Insert MSC Object

Description

The *Insert Object* service will create a new object in the diagram identified by the parameter and return an object identification (an integer) to the client. It will not display the new object.

For a full specification of how to specify the object to insert, the specification of Message Sequence Charts Z.120 should be consulted or the document "MSC Trace and Log Format 2.0 Specification" available from Telelogic.

Tools Supporting the Service

SET_MSCE

Service Request

SEINSERTOBJECT

Parameter	Туре	Description
bufId	integer	Buffer in which to insert the object.
afterObject	integer	Only object id 0 is allowed, and adds the object as the last object.
objectDescription	string	Description of the object to in- sert. The description should be in accordance with Z.120 using <i>event oriented</i> PR. Only one object at the time can be insert- ed.

Service Reply

SEINSERTOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.
objectId	integer	Identifier to object.

Errors

Invalid diagram buffer id Errors from parsing objectDescription

Remove Object

Description

The Remove Object service will delete an object in a diagram.

Tools Supporting the Service

SET_MSCE

Service Request

SEREMOVEOBJECT

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which an object should be removed.
objectId	integer	The object to remove.

Service Reply

SEREMOVEOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid diagram buffer id
Invalid object id
Object can not be removed (MSCE only)
```

Get Object Text

Description

The service will extract all texts for a given object.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEGETOBJECTTEXT

Parameter	Туре	Description
bufid	string	A buffer referencing the diagram where the object exists.
objectId	integer	Identifier to the object.

Service Reply

SEGETOBJECTTEXTREPLY

Parameter	Туре	Description
status	integer	Service reply status.
objectType	integer	The type of the object. Definitions of possible object types are found in sdtsym.h.
textnumber	integer	The number of the text strings.
texts	stringlist	All the texts associated to the object.

Errors

Invalid diagram buffer id No page found with this object id No object with this id found

Editor – Object Attribute Services

Introduction

It is possible to extend the data associated to an object managed by the SDL suite with the user's own data. How this is done is described in Figure 165.



Figure 165: Extending the attributes associated to a symbol

The object can have any number of attributes associated to it, not necessarily 2 as illustrated. The number of extended data attributes is arbitrary.

An object is potentially any item that is handled as a source component by the SDL suite, or any of the sub-components of which it consists. Objects are thus:

- SDL diagram
- MSC diagram
- OM, SC, HMSC diagram
- SDL Page
- OM, SC, HMSC page
- SDL symbol
- MSC symbol
- OM, SC, HMSC symbol

The purpose of the extended data attribute is:

• To allow the user to add his own-defined data to diagrams managed by the SDL suite.

• In addition, data extensions can by connected to operations to be performed on that data. Operations are performed through PostMaster messages or Operating System commands.

Extended Data Attribute

An extended data attribute defines a number of fields. Below is a few of them further elaborated.

• ProprietaryKey

Provides a means to tag extensions with the originating company / organization. Should normally be set to 0.

• AttributeKey

Up to the user's preference when deciding how to use this key.

• data_length

Provides information about how many bytes of data the variant_data union consists of. Generally, all elements of variable length need to have their length specified, since the user must be able to store hex 0 as data.

data_interpretation and data

In the SDL suite, data is always interpreted as *raw data*. The SDL suite have no knowledge about the format and meaning of the data. *Raw data* may consist of information produced by an external tool. Its format is not known by the SDL suite. Thus, operation of any sort applied on that data, cannot be performed.

• comment_length

How many bytes the comment consists of.

• comment

Gives the possibility to attach a (preferable) readable comment in plain text. The comment could assist the user in understanding the meaning of data that he has attached to the object.

Display Key

Description

Request to display, i.e. select, all symbols referred by SDTRef, that have at least one associated attribute that matches the keys ProprietaryKey and AttributeKey.

- AttributeKey will be omitted if it has value 0
- If ProprietaryKey is 0 both keys will be omitted.

SDTRef is typically a diagram, but could also be a page or a specific symbol. The symbols that match the criteria will be marked as selected in the drawing area. Other symbols will be de-selected.

- If SDTRef is empty, display all symbols is all diagrams that are read by the tool, and that match the keys
- If SDTRef denotes a diagram, display all symbols in the diagram that match the keys
- If SDTRef denotes a page, display all symbols in the page that match the keys
- If SDTRef denotes a symbol, display the symbol if it matches the keys.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEDISPLAYKEY

Parameter	Туре	Description
ProprietaryKey	integer	See the introductory description.
AttributeKey	integer	See the introductory description.
SDTRef	string	See the introductory description.

Service Reply

SEDISPLAYKEYREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

SDT Reference Errors

List Key

Description

Request to list all objects referred by SDTRef, that have at least one associated attribute that matches the keys ProprietaryKey and AttributeKey.

- AttributeKey will be omitted if it has value 0.
- If ProprietaryKey is 0 both keys will be omitted.

SDTRef is typically a diagram, but could also be a page or a specific symbol. The response is a count and a list of SDTRef to all objects that match the keys.

- If SDTRef is NULL, list all symbols is all diagrams that are read by the tool, and that match the keys.
- If SDTRef denotes a diagram, return SDTRef to all objects in the diagram that match the keys.
- If SDTRef denotes a page, return SDTRef to objects in the page that match the keys.
- If SDTRef denotes an object, return the SDTRef to the object if it matches the keys.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SELISTKEY

Parameter	Туре	Description
ProprietaryKey	integer	See the introductory description.
AttributeKey	integer	See the introductory description.
SDTRef	string	See the introductory description.

Chapter 12 The Telelogic Tau Public Interface

Service Reply

SELISTKEYREPLY

Parameter	Туре	Description
status	integer	Service reply status.
NoOfKeys	integer	Number of matching keys.
SDTRef*	string	A list of SDT references.

Errors

SDT Reference Errors

Create Attribute

Description

Request add a new attribute to the symbol that matches the SDT reference SDTRef.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SECREATEATTRIBUTE

Parameter	Туре	Description
ProprietaryKey	integer	A key that allows the external user / tool to distinguish his ex- tensions from other users' / tools'.
AttributeKey	integer	A key that allows the external user / tool to classify his own ex- tension.
SDTRef	string	A valid SDT reference identify- ing the object.
datainterpret	integer	Defines how data is to be inter- preted. Should be set to 0.
comment	string	An explanatory (readable) com- ment.
MenuChoice	string	The name of the pop-up menu choice that will be appended to the tool's pop-up menu, upon se- lection of the symbol.
dataLen	integer	Length of data.
data	ByteString	The user's data.

Service Reply

SECREATEATTRIBUTEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

<u>SDT Reference Errors</u> Error in match

Update Attribute

Description

Request to update the attribute that matches the search criteria (SDTRef, ProprietaryKey, AttributeKey). If an attribute matches the search keys, then the entire attribute's contents will be updated (replaced) with new contents.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEUPDATEATTRIBUTE

Parameter	Туре	Description
ProprietaryKey	integer	A key that allows the exter- nal user / tool to distinguish his extensions from other users' / tools'.
AttributeKey	integer	A key that allows the exter- nal user / tool to classify his own extension.
SDTRef	string	A valid SDT reference identifying the object.
datainterpret	integer	Defines how data is to be interpreted. The following values are recognized: Should be set to 0.
comment	string	An explanatory (readable) comment.
MenuChoice	string	The name of the pop-up menu choice that will be appended to the tool's pop- up menu, upon selection of the symbol.
dataLen	integer	Length of data.

Parameter	Туре	Description
data	ByteString	The user's data.

Service Reply

SEUPDATEATTRIBUTEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

<u>SDT Reference Errors</u> Error in match

Read Attribute

Description

Request to read the attribute(s) that match(es) the keys and the diagram identified by SDTRef. The SDTRef reference should contain a page attribute.

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEREADATTRIBUTE

Parameter	Туре	Description
ProprietaryKey	integer	See the introductory description.
AttributeKey	integer	See the introductory description.
SDTRef	string	See the introductory description.

Service Reply

SEREADATTRIBUTEREPLY

Parameter	Туре	Description
status	integer	Service reply status.
DataInterpret	integer	Type of data, always 0.
comment	string	Comment text.
MenuChoice	string	The defined menu choice.
dataLen	integer	Length of binary data.
data	ByteString	Data associated to the extended attribute.

Errors

<u>SDT Reference Errors</u> Error in match

Delete Attribute

Description

Request to delete the attribute that matches the search criteria (SDTRef, ProprietaryKey, AttributeKey).

Tools Supporting the Service

SET_SDLE SET_MSCE SET_OME

Service Request

SEDELETEATTRIBUTE

Parameter	Туре	Description
ProprietaryKey	integer	See the introductory description.
AttributeKey	integer	See the introductory description.
SDTRef	string	See the introductory description.

Service Reply

SEDELETEATTRIBUTEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

<u>SDT Reference Errors</u> More than one object match the search No object match the search

Information Server Services

Load Definition File

Description

Loads a file containing external signal definitions into the Information Server. The contents of such a file are then made available via the <u>Signal</u> <u>Dictionary</u> functionality found in the SDL Editor.

The Information Server could read any ASCII file, but for efficient usage, the format of the files should contain one signal definition per line.

Tools Supporting the Service

SET_INFOSERVER

Service Request

SELOADDEFINITIONMAP

Parameter	Туре	Description
fileName	string	Name of the file to load.
tag	integer	Is used to either add or remove a file. 0 = Add definition 1 = Remove definition.

Service Reply

SELOADDEFINITIONMAPREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

File already added File was not added before

SDL Editor Services

GRPR

Description

Generate PR for a given binary file. Optionally, GR references and CIF comments could be generated.

If the file is already being edited the PR generated will be from the internal version in the SDL Editor and not from the data in the file.

If the diagram is not being edited it will be unloaded after the PR generation.

Tools Supporting the Service

SET_SDLE

Service Request

SEGRPRP

Parameter	Туре	Description
fileName	string	The file where the SDL/GR is stored.
fileName	string	The file where the PR is written
append	bool	If <i>true</i> , the PR will be appended to the file.
generateGRRef	bool	If <i>true</i> , the generated PR will also contain graphical references, that are used in the Analyzer to backtrace errors.
generateCIF	bool	If <i>true</i> , the PR will also contain CIF comments.

Service Reply

SEGRPRREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid file name
SDLE is busy, syntax error in text (SDLE only)
```
Tidy Up

Description

The *Tidy Up* service will perform a tidy up on a specified diagram. The functionality is identical to the <u>*Tidy Up*</u> command available in the SDL Editor.

Tools Supporting the Service

SET SDLE

Service Request

SETIDYUP

Parameter	Туре	Description
bufid	integer	Buffer referencing the diagram to tidy up.

Service Reply

SETIDYUPREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Invalid diagram buffer id

SC Editor Services

Get Diagram Info

Description

Get information about the symbols and lines of a SC diagram. The format of the returned *infoList* is:

Tools Supporting the Service

SET_OME

Service Request

SEGETDIAGRAMINFO

Parameter	Туре	Description
bufId	integer	Refers to a buffer in the editor.

Service Reply

SEGETDIAGRAMINFOREPLY

Parameter	Туре	Description
status	integer	Service reply status.
infoList	stringList	The symbols and lines of the diagram.

Errors

```
Invalid diagram buffer id
Invalid diagram type
```

MSC Editor Services

MSC GRPR

Description

Generate PR for a given MSC binary file. Optionally, GR references can be generated.

If the file is already being edited, the generated PR will be from the internal version in the MSC Editor and not from the data in the file.

If the diagram is not being edited it will be unloaded after the PR generation.

Tools Supporting the Service

SET_MSCE

Service Request

SEMSCGRPR

Parameter	Туре	Description
fileName	string	The file where the MSC/GR is stored.
fileName	string	The file where the PR is written.
mscDocName	string	The name of the MSC document. If emp- ty the PR will be appended to PR file.
generateGRRef	bool	If <i>true</i> , the generated PR will also contain graphical references, that are used in the Analyzer to backtrace errors.

Service Reply

SEMSCGRPRREPLY

Parameter	Туре	Description
errors	integer	Number of errors in the GR file.
warnings	integer	Number of warnings in the GR file.
errorLog	string	Description of errors and warnings, with GR references.

HMSC Editor Services

HMSC GRPR

Description

Generate PR for a given HMSC binary file. Optionally, GR references can be generated.

If the file is already being edited, the generated PR will be from the internal version in the HMSC Editor and not from the data in the file.

If the diagram is not being edited it will be unloaded after the PR generation.

Tools Supporting the Service

SET_OME

Service Request

SEHMSCGRPR

Parameter	Туре	Description
fileName	string	The file where the HMSC/GR is stored.
fileName	string	The file where the PR is written.
mscDocName	string	The name of the HMSC document. If empty the PR will be appended to PR file.
generateGRRef	bool	If <i>true</i> , the generated PR will also contain graphical references, that are used in the Analyzer to backtrace errors.

Service Reply

SEHMSCGRPRREPLY

Parameter	Туре	Description
errors	integer	Number of errors in the GR file.
warnings	integer	Number of warnings in the GR file.
errorLog	string	Description of errors and warnings, with GR references.

CIF Services

The CIF services enables the user to create diagrams where the graphical objects and texts could be positioned in a controlled manner.

These services enables the user to build converters to SDL suite diagram formats.

The diagram types which are supported are:

- Block interaction diagrams,
- Process interaction diagrams

where the services are supported by the SDL Editor, and

• OM diagrams,

where the services are supported by the OM Editor.

Create SDL Diagram

Description

Creates an empty SDL diagram in a new buffer. Corresponds to the SDL Editor command *New*. The defaults applied to the created page are defined for the <u>Create SDL Page</u> service.

The texts for the package reference and the headings will be shown on every page in the diagram. The package reference symbol and the heading symbols will be automatically created, but their size can be changed by using the <u>Insert SDL Object</u> service for these symbols, (see <u>"Object Specific Parameters" on page 638</u>).

The kernel heading will be parsed to extract the necessary attributes like the virtuality, diagram type and diagram name. If this parsing fails the service will return an error.

The splitting of the SDL heading into a kernel heading and an additional heading is done according to the rules in section 2.2.5 'Partitioning of diagrams' in Z.100 with some necessary exceptions. Normally the kernel heading contains the heading up to, and including the diagram name,

```
<kernel heading> ::= [<virtuality>] <diagram kind>
{<diagram name> | <diagram identifier>}
```

The following exceptions to this rule apply:

Chapter 12 The Telelogic Tau Public Interface

For a type based system diagram the kernel heading contains the complete definition except for the ending <end> clause.

<kernel heading> ::= <typebased system heading>

For a process diagram the <number of process instance> might be included in the kernel heading.

For a procedure diagram the kernel heading is

```
<kernel heading> ::= <procedure preamble> procedure
{<procedure name> | <procedure identifier>}
```

Tools Supporting the Service

SET SDLE

Service Request

SESDLECIFCREATEDIAGRAM

Parameter	Туре	Description
packageReferenceText	string	The text that will appear in the package reference sym- bol
kernelHeadingText	string	The text that will appear in the kernel heading symbol
AdditionalHeadingText	string	The text that will appear in the additional heading sym- bol
радеТуре	integer	Type of the first page in the diagram. See sdtsym.h for valid page types. Note that the pageType must corre- spond with the diagram- Type.
pageName	string	Name of the first page in the diagram.
pageWidth	integer	The width of the page.
pageHeight	integer	The height of the page.
gridWidth	integer	The grid width.

Parameter	Туре	Description
gridHeight	integer	The grid height.
autonumbered	bool	If true the page is an auto- numbered page.
showMeFirst	bool	If true this page is set as the ShowMeFirst page.
scale	integer	The scale used when shown, 100 means a 1:1 scale.

Service Reply

SESDLECIFCREATEDIAGRAMREPLY

Parameter	Туре	Description
status	integer	Service reply status.
bufid	integer	Refers to a buffer in the editor.

Errors

The pagename is empty Illegal diagram kernel heading

Emitted Notifications

Notification	Description
SESDLENEWNOTIFY	As a result of creating the diagram.
SEPAGENOTIFY	As a result of creating a page.

Chapter **12** The Telelogic Tau Public Interface

Create SDL Page

Description

The *Create Page* service will create a new page in an existing diagram. The new page will be inserted as the last page in the diagram. The created page will have a frame symbol with margins to the page border as defined by the actual preference file. The kernel heading, additional heading and page number symbols exists and are placed automatically relative to the frame symbol. The package reference symbol exists, automatically placed, if the page type is a system page or a package page. The frame symbol can be changed to another position by using the <u>Insert SDL Object</u> service.

If a page already exists with the given page name, the service will be ignored and the return status is OK. This can be used for convenience to always send Create Page for all pages including the one already sent in the <u>Create SDL Diagram</u>.

The grid values should be multiples of 5 mm. If the values are zero or less than zero this means that the preferences values are used for the grid values.

A page can be set to be autonumbered. When this is used a page name must be given as well. For consecutive autonumbered pages these names must be "1", "2", "3" and so on. It will be checked that the given page name is the same as the generated page name. If not an error will be returned.

If the *ShowMeFirst* attribute is set for a new page it will override an already existing ShowMeFirst page.

Tools Supporting the Service

SET SDLE

Service Request

SESDLECIFCREATEPAGE

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which the page will be created.

Parameter	Туре	Description
радеТуре	integer	Type of the page in the diagram. See sdtsym.h for valid page types. Note that the pageType must correspond with the diagramType.
pageName	string	Name of the new page in the diagram.
pageWidth	integer	The width of the page.
pageHeight	integer	The height of the page.
gridWidth	integer	The grid width.
gridHeight	integer	The grid height.
autonumbered	bool	If true the page is an autonumbered page.
showMeFirst	bool	If true this page is set as the ShowMe- First page.
scale	integer	The scale used when shown, 100 means a 1:1 scale.

Service Reply

SESDLECIFCREATEPAGEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid diagram buffer id
The pagename is empty
Command canceled in add page
For autonumbered pages, the given pagename "2" must
be the same as the generated page name "1"
```

Emitted Notifications

Notification	Description
SEPAGENOTIFY	As a result of creating a page.

Chapter 12 The Telelogic Tau Public Interface

Insert SDL Object

Description

The *Insert Object* service will insert an object in a specified position at a specific page. The return value is the object id for the inserted object. Lines connected to the inserted symbol should refer to the *objectId* returned by this service.

The following checks are applied on parameters:

- If a line is inserted the symbols connecting the line must already have been inserted.
- For lines it is an error if there are less than two breakpoints.
- The object type must be a known symbol or line.

The following is a non-exhaustive list of restrictions, applied inside SDLE when manipulating with symbols, that are **not** checked by this service:

- Ratio for process symbols, should be 1:2 (height vs. width).
- Positions outside the page area.
- Text syntax for reference symbols.
- Illegal texts on flowlines.
- Sizes less than minimum allowed.
- That lines are connected to symbols with the correct syntax.
- That line endpoints fall on enclosing rectangle of the symbol it is connected to.
- Symbols placed on pages where they should not appear, e.g. a task on a system page.
- Overlapping symbols

Tools Supporting the Service

SET_SDLE

Service Request

SESDLECIFINSERTOBJECT

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which an object should be shown.
pagename	string	An existing page within the diagram.
objectType	integer	Identify the type of object to insert. Val- id values for all symbols are given in sdtsym.h. For lines the values are listed below.
updateFlag	bool	If this flag is true a routine will be called for this page to fix some attributes for the objects. This fixing must be done be- fore the page is saved. If this flag is false this fix will not be done.
object specific parameters	varying	One or more parameters defining the in- serted object.

Service Reply

SESDLECIFINSERTOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.
objectId	integer	Identifier to the inserted object.

Errors

Invalid diagram buffer id Invalid pagename Invalid fromSymbol id for a line: Invalid toSymbol id for a line: Invalid fromSymbol. The fromSymbol is a line: Invalid toSymbol. The toSymbol is a line: Number of points in line < 2 : Invalid objectId for ConnectionPoint line: Invalid object type: Chapter 12 The Telelogic Tau Public Interface

Object Specific Parameters

For each object type the needed parameters are listed. All coordinates are given in 1/10 mm units in the coordinate system with the upper left corner of the page as origo and the y-coordinate having the positive direction going downwards.

A given coordinate will be adjusted to fall on a grid. For symbols this grid is given by a grid net of 5 x 5 mm. For line breakpoints and text positions the grid net is 2.5×2.5 mm.

The *Pagenumber* symbol (objectType number 53) has no extra specific parameters. When inserted there will not be a new symbol created as the pagenumber symbol is always created when the page is created. The text for the pagenumber symbol is generated automatically. The only use of inserting this symbol is to force an update of the page by having the *updateFlag* set to *true*.

Process Interaction diagram symbols

Start(1), ProcedureStart(2), MacroInlet(3), Stop(4), ProcedureReturn(5), MacroOutlet(6), State(7), InputRight(8), InputLeft(9), PriorityInputRight(10), PriorityInputLeft(11), ContinuousSignal(12), EnablingCondition(13), Task(14), OutputRight(15), OutputLeft(16), PriorityOutputRight(17), PriorityOutputLeft(18), ProcedureCall(19), InConnector(20), CreateRequest(21), Decision(22), TransitionOption(23), MacroCall(24), OutConnector(25), Save(26)

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
width	integer	The width of the symbol.
height	integer	The height of the symbol.
text	string	The text in the symbol. The text is auto- matically aligned according to the built- in routines in SDLE

• Block Interaction diagram symbols

```
Procedure(27), Service(41), Process(42), Block(43),
BlockSubstructure(45), ChannelSubstructure(46),
SystemType(47), BlockType(48), ServiceType(49),
ProcessType(50), Operator(56).
```

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
width	integer	The width of the symbol.
height	integer	The height of the symbol.
dashed	bool	If true the symbol will be dashed, this is valid for block, process and service symbols.
text	string	The text in the symbol.
XtextPosition	integer	The x coordinate for the left top position of the text.
YtextPosition	integer	The y coordinate for the left top position of the text.

There is no check that the text will fall inside the symbol boundary.

• TextSymbol(28)

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
width	integer	The width of the symbol.
height	integer	The height of the symbol.
text	sString	The text in the symbol.

An automatically sized *TextSymbol* shall have the *width* and *height* set to zero. If values are given this means that the symbol will be clipped using the given sizing values.

• **Comment**(29), **CommentLeft**(30), **TextExtension**(31) and **TextExtensionLeft**(32)

Parameter	Туре	Description
left	integer	The x coordinate for the left side (Com- ment, TextExtension) or the right side (CommentLeft, TextExtensionLeft)
top	integer	The y coordinate for the top side.
text	string	The text in the symbol.

For a *CommentLeft* or *TextExtensionLeft* the position gives the right top corner of the symbol. The size is automatically adjusted.

• FrameSymbol(44)

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
width	integer	The width of the symbol.
height	integer	The height of the symbol.

A default *FrameSymbol* will always exist on a created page. This service will replace the existing Frame with a new frame symbol at the given position with the new size. The existing *heading symbols*, *package reference symbol* and the *pagenumber symbol* will be automatically adjusted.

• AdditionalHeading(52), PackageReference(55)

Parameter	Туре	Description
width	integer	The width of the symbol.
height	integer	The height of the symbol.

For automatically sized symbols the values for the *width* and the *height* should be zero. If values different from zero are given this means that the symbol will be clipped.

Note that it might be necessary to insert this symbol for every page. This is so because these symbols might have a unique size for every page. If the symbols are not clipped there is no need to insert the symbol as they will be automatically generated as an unclipped symbol.

The size of the kernel heading cannot be changed.

• FlowLine(101)

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts.
toID	integer	The objectId for the symbol where the line ends.
textExists	bool	Is <i>true</i> if there is text on the flow line.
text	string	Optional text in the message. Only exists if textExists is true.
XtextPosition	integer	Optional text position in the message. Only exists if textEx- ists is true. x coordinate for the left top corner of the text.
YtextPosition	integer	Optional text position in the message. Only exists if textEx- ists is true. y coordinate for the left top corner of the text.
dashed	bool	If <i>true</i> the line will be dashed.
numPoints	integer	The number of breakpoints for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exist having the given *fromID* and *toID*. The text and text position shall only exist if the *textExists* flag is *true*. *Flowlines* after *Decision*, *TransitionOption* and *MacroCall* should always have text. There will be *numPoints x* and *y*-values at the end of the message. Line coordinates are adjusted to the nearest grid point but it will not be checked whether line endpoints are placed on the symbol boundaries, as they are expected to be.

• **SignalRoute**(102)

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts. Zero means ENV.
toID	integer	The objectId for the symbol where the line ends. Zero means ENV.
name	string	The name of the signal route.
signalList	string	The text in the signal list.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.
XsignalListPosition	integer	The x coordinate for the left top position of the signal list.
YsignalListPosition	integer	The y coordinate for the left top position of the signal list.
numPoints	integer	The number of breakpoints for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exists having the given *fromID* and *toID*. There will be *numPoints x* and *y*-values at the end of the message. Text position and line coordinates are adjusted to the nearest grid point but it will not be checked whether line endpoints are placed on the symbol boundary (the rectangle enclosing the symbol), as they are expected to be. Extra line segments needed for e.g. a signal route entering a service will be automatically generated.

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts. Zero means ENV.
toID	integer	The objectId for the symbol where the line ends. Zero means ENV.
name	string	The name of the signal route.
signalList1	string	The text in the signal list in the direction from fro- mID to toID.
signalList2	string	The text in the signal list in the reversed direc- tion.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.
XsignalList1Position	integer	The x coordinate for the left top position of the first signal list.

• **DoubleSignalRoute**(103)

Parameter	Туре	Description
YsignalList1Position	integer	The y coordinate for the left top position of the first signal list.
XsignalList2Position	integer	The x coordinate for the left top position of the second signal list.
YsignalList2Position	integer	The y coordinate for the left top position of the second signal list.
numPoints	integer	The number of break- points for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

• **Channel**(104)

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts. Zero means ENV.
toID	integer	The objectId for the symbol where the line ends. Zero means ENV.
name	string	The name of the channel.
signalList	string	The text in the signal list.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.

Parameter	Туре	Description
XsignalListPosition	integer	The x coordinate for the left top position of the signal list.
YsignalListPosition	integer	The y coordinate for the left top position of the signal list.
XarrowPosition	integer	The x coordinate for the arrow.
YarrowPosition	integer	The y coordinate for the arrow.
noDelay	bool	If true the channel is a non delaying channel.
numPoints	integer	The number of break- points for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exists having the given *fromID* and *toID*. There will be *numPoints x* and *y*-values at the end of the message. Text position and line coordinates are adjusted to the nearest grid point but it will not be checked whether line endpoints are placed on the symbol boundary (the rectangle enclosing the symbol), as they are expected to be. The arrow position are automatically adjusted to be placed at the line as close as possible to the given arrow point. The angle of the arrow is automatically calculated. For a non delaying channel the arrow position is ignored and it will be automatically placed at the last breakpoint.

• **DoubleChannel**(104)

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts. Zero means ENV.
toID	integer	The objectId for the symbol where the line ends. Zero means ENV.
name	string	The name of the channel.
signalList1	string	The text in the signal list in the direction from fromID to toID.
signalList2	sString	The text in the signal list in the reversed direction.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.
XsignalList1Position	integer	The x coordinate for the left top position of the first sig- nal list.
YsignalList1Position	integer	The y coordinate for the left top position of the first sig- nal list.
XsignalList2Position	integer	The x coordinate for the left top position of the second signal list.
YsignalList2Position	integer	The y coordinate for the left top position of the second signal list.
Xarrow1Position	integer	The x coordinate for the first arrow.

Parameter	Туре	Description
YarrowlPosition	integer	The y coordinate for the first arrow.
Xarrow2Position	integer	The x coordinate for the second arrow.
Yarrow2Position	integer	The y coordinate for the second arrow.
noDelay	bool	If true the channel is a non delaying channel.
numPoints	integer	The number of breakpoints for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

• CreateLine(106), ChannelSubstructureLine(107)

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts. This object is a channel for the ChannelSubstructureLine.
toID	integer	The objectId for the symbol where the line ends.
numPoints	integer	The number of breakpoints for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exists having the given *fromID* and *toID*. There will be *numPoints x* and *y*-values at the end of the message. Line coordinates are adjusted to the nearest grid point but it will not be checked whether line endpoints are placed on the symbol boundary, as they are expected to be. For a *ChannelSubstructureLine* the starting is adjusted to be placed on the channel.

• **GateIn**(109), **GateOut**(110)

Parameter	Туре	Description
X	integer	The x coordinate for the gate position on the FrameSymbol boundary.
У	integer	The y coordinate for the gate position on the FrameSymbol boundary.
dashed	bool	If true the gate is dashed.
name	string	The name of the gate.
signalList	string	The text in the signal list.
constraint	string	The text in the constraint symbol connected to the gate.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.
XsignalListPosition	integer	The x coordinate for the left top position of the sig- nal list.
YsignalListPosition	integer	The y coordinate for the left top position of the sig- nal list.

The length of the gate is automatically calculated, so also the arrow position. The point will be adjusted such that it is guaranteed to stay on the Frame boundary. If the frame symbol is inserted after a gate, the gate might be positioned outside the frame. The correct positions will be fixed in the update that must be performed before saving the page.

• **DoubleGate**(111)

Parameter	Туре	Description
X	integer	The x coordinate for the gate position on the FrameSymbol boundary.
У	integer	The y coordinate for the gate position on the FrameSymbol boundary.
dashed	bool	If true the gate is dashed.
name	string	The name of the gate.
signalList1	string	The text in the signal list in the direction into the diagram.
signalList2	string	The text in the signal list in the direction out of the diagram.
constraint	string	The text in the constraint symbol connected to the gate.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.
XsignalList1Position	integer	The x coordinate for the left top position of the first signal list.
YsignalList1Position	integer	The y coordinate for the left top position of the first signal list.

Parameter	Туре	Description
XsignalList2Position	integer	The x coordinate for the left top position of the second signal list.
YsignalList2Position	integer	The y coordinate for the left top position of the second signal list.

• **ConnectionPoint**(112)

Parameter	Туре	Description
objectID	integer	The objectId for the symbol where the ConnectionPoint exists. Zero means ENV.
Х	integer	The x coordinate for the ConnectionPoint.
У	integer	The y coordinate for the Connec- tionPoint.
name	string	The name of the ConnectionPoint.
XnamePosition	integer	The x coordinate for the left top position of the name.
YnamePosition	integer	The y coordinate for the left top position of the name.

A *ConnectionPoint* is used for gate names on the Frame symbol (ENV), channel names in block diagrams for graphical connect statements and for gate names inside block/process/service instances of block/process/service types. The *objectID* must be a valid symbol. The coordinate is adjusted such that the point will fall on the symbol boundary. If the frame symbol is inserted after a connection point residing on the frame, it might be positioned outside the frame. The correct positions will be fixed in the update that must be performed before saving the page.

Create OM Diagram

Description

Creates an empty OM diagram in a new buffer. Corresponds to the OM Editor command *New*. The defaults applied to the created page are defined for the <u>Create OM Page</u> service.

Tools Supporting the Service

SET OME

Service Request

SEOMECIFCREATEDIAGRAM

	1	1
Parameter	Туре	Description
name	string	Diagram name.
pageName	string	Name of the first page in the dia- gram.
pageWidth	integer	The width of the page.
pageHeight	integer	The height of the page.
gridWidth	integer	The grid width.
gridHeight	integer	The grid height.
autonumbered	bool	If true the page is an autonumbered page.
showMeFirst	bool	If true this page is set as the Show- MeFirst page.
scale	integer	The scale used when shown, 100 means a 1:1 scale.

Service Reply

SEOMECIFCREATEDIAGRAMREPLY

Parameter	Туре	Description	
status	integer	Service reply status.	
bufid	integer	Refers to a buffer in the editor	

Errors

The pagename is empty

Emitted Notifications

Notification	Description
SEOMENEWNOTIFY	As a result of creating the diagram.
SEPAGENOTIFY	As a result of creating a page.

Create OM Page

Description

The *Create Page* service will create a new page in an existing diagram. The new page will be inserted as the last page in the diagram. The heading and page number symbols exists and are placed automatically relative to the frame symbol. The frame symbol can not be changed to another position.

If a page already exists with the given page name, the service will be ignored and the return status is OK. This can be used for convenience to always send Create Page for all pages including the one already sent in the <u>Create OM Diagram</u> service.

The grid values should be multiples of 5 mm. If the values are zero or less than zero this means that the preferences values are used for the grid values.

A page can be set to be autonumbered. When this is used a page name must be given as well. For consecutive autonumbered pages these names must be "1", "2", "3" and so on. It is considered an error if the given page name not is the same as the generated page name when this flag is set.

If the *ShowMeFirst* attribute is set for a new page it will override an already existing ShowMeFirst page.

Tools Supporting the Service

SET OME

Service Request

SEOMECIFCREATEPAGE

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which the page will be created.
pageName	string	Name of the new page in the dia- gram.
pageWidth	integer	The width of the page.
pageHeight	integer	The height of the page.

Chapter 12 The Telelogic Tau Public Interface

Parameter	Туре	Description
gridWidth	integer	The grid width.
gridHeight	integer	The grid height.
autonumbered	bool	If true the page is an autonumbered page.
showMeFirst	bool	If true this page is set as the Show- MeFirst page.
scale	integer	The scale used when shown, 100 means a 1:1 scale.

Service Reply

SEOMECIFCREATEPAGEREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

```
Invalid diagram buffer id
The pagename is empty
For an autonumbered page the given name must be the
same as the generated pagename: "<gen.pagename>"
```

Emitted Notifications

Notification	Description
SEPAGENOTIFY	As a result of creating a page.

Insert OM Object

Description

The *Insert Object* service will insert an object in a specified position at a specific page. The return value is the object id for the inserted object. Lines connected to the inserted symbol should refer to the objectId returned by this service.

The following parameter checks are performed:

- If a line is inserted the symbols connecting the line must already have been inserted.
- For lines it is an error if there are less than two breakpoints.
- The object type must be a known symbol or line.

The following is a non-exhaustive list of restrictions, applied inside OME when manipulating with symbols, that are **not** checked by this service:

- Positions outside the page area.
- Text syntax for class and instance symbols. However, when an object is inserted with the *updateFlag* set there will be a syntactic check and all eventual syntax errors will be marked in the normal way.
- That lines are connected to symbols with the correct syntax.
- That line endpoints fall on or inside the enclosing rectangle of the symbol it is connected to.
- Overlapping symbols

Tools Supporting the Service

SET_OME

Service Request

SEOMECIFINSERTOBJECT

Parameter	Туре	Description
bufId	integer	Buffer referencing the diagram in which an object should be shown.
pagename	string	An existing page within the diagram.
objectType	integer	Identify the type of object to insert. Val- id values for all symbols are given in sdtsym.h. For lines the values are listed below.
updateFlag	bool	If this flag is true a routine will be called for this page to fix some attributes for the objects. This fixing must be done be- fore the page is saved. If this flag is false this fix will not be done.
object specific parameters	varying	One or more parameters defining the in- serted object.

Service Reply

SEOMECIFINSERTOBJECTREPLY

Parameter	Туре	Description
status	integer	Service reply status.
objectId	integer	Identifier to the inserted object

Errors

Invalid diagram buffer id Invalid pagename Invalid fromSymbol id for a line: Invalid toSymbol id for a line: Invalid fromSymbol. The fromSymbol is a line: Invalid toSymbol. The toSymbol is a line: Number of points in line < 2 : Invalid objectId for ConnectionPoint line: Invalid object type:

Object Specific Parameters

For each object type the needed parameters are listed below. All coordinates are given in 1/10 mm units in the coordinate system with the upper left corner of the page as origo and the y-coordinate having the positive direction going downwards.

A given coordinate will be adjusted to fall on a grid. For symbols this grid is given by a grid net of 5 x 5 mm. For line breakpoints and text positions the grid net is 2.5×2.5 mm.

The *Pagenumber* symbol (objectType number 53) have no extra specific parameters. When inserted there will not be a new symbol created as the pagenumber symbol is always created when the page is created. The text for the pagenumber symbol is generated automatically. The only use of inserting this symbol is to force an update of the page by having the *updateFlag* set to true

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
nameText	string	The name text part of the symbol.
attributeText	string	The attribute text part of the symbol.
operationText	string	The operation text part of the symbol.
stereotypeText	string	The stereotype text part of the symbol.
propertiesText	string	The properties text part of the symbol.
collapsed	bool	If this is true the symbol is collapsed.

• Class(57)

• Instance(58)

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
nameText	string	The name text part of the symbol.
attributeText	string	The attribute text part of the symbol.
stereotypeText	string	The stereotype text part of the symbol.
propertiesText	string	The properties text part of the symbol.
collapsed	bool	If this is true the symbol is collapsed.

• TextSymbol(28)

Parameter	Туре	Description
left	integer	The x coordinate for the left side.
top	integer	The y coordinate for the top side.
text	string	The text in the symbol.
collapsed	bool	If this is true the symbol is collapsed

• AssociationLine(113), AggregationLine(114)

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts.
toID	integer	The objectId for the symbol where the line ends.
haveName	bool	Flag set if the line has forward name attributes

Parameter	Туре	Description
name	string	Only if <i>haveName</i> set: The for- ward name of the line.
nameArrow	bool	Only if <i>haveName</i> set: Arrow on the forward name.
XPosition	integer	Only if <i>haveName</i> set: The x co- ordinate for the left top position of the name.
YPosition	integer	Only if <i>haveName</i> set: The y co- ordinate for the left top position of the name.
haveRevName	bool	Flag set if the line has reversed name attributes.
revName	string	Only if <i>haveRevName</i> set: The reversed name of the line.
revNameArrow	bool	Only if <i>haveRevName</i> set: Arrow on the reversed name.
XPosition	integer	Only if <i>haveRevName</i> set: The x coordinate for the left top position of the reversed name.
YPosition	integer	Only if <i>haveRevName</i> set: The y coordinate for the left top position of the reversed name.
haveRoleName	bool	Flag set if the line has role name attributes.
roleName	string	Only if <i>haveRoleName</i> set: The forward role name of the line.
XPosition	integer	Only if <i>haveRoleName</i> set: The x coordinate for the left top position of the forward role name.
YPosition	integer	Only if <i>haveRoleName</i> set: The y coordinate for the left top position of the forward role name.

Parameter	Туре	Description
haveRevRoleName	bool	Flag set if the line has reversed role name attributes.
revRoleName	string	Only if <i>haveRevRoleName</i> set: The reversed role name of the line.
XPosition	integer	Only if <i>haveRoleName</i> set: The x coordinate for the left top position of the reversed role name.
YPosition	integer	Only if <i>haveRoleName</i> set: The y coordinate for the left top position of the reversed role name.
haveMultText	bool	Flag set if the line has multiplic- ity text attributes.
multText	string	Only if <i>haveMultText</i> set: The forward multiplicity text of the line.
XPosition	integer	Only if <i>haveMultText</i> set: The x coordinate for the left top position of the multiplicity text.
YPosition	integer	Only if <i>haveMultText</i> set: The y coordinate for the left top position of the multiplicity text.
haveRevMultText	bool	Flag set if the line has reversed multiplicity text attributes.
revMultText	string	Only if <i>haveRevMultText</i> set: The reversed multiplicity text of the line.
XPosition	integer	Only if <i>haveRevMultText</i> set: The x coordinate for the left top position of the multiplicity text.
YPosition	integer	Only if <i>haveRevMultText</i> set: The y coordinate for the left top position of the multiplicity text.

Parameter	Туре	Description
haveQualText	bool	Flag set if the line has qualifier text attributes.
qualText	string	Only if <i>haveQualText</i> set: The forward qualifier text of the line.
haveRevQualText	bool	Flag set if the line has reversed qualifier text attributes.
revQualText	string	Only if <i>haveRevQualText</i> set: The reversed qualifier text of the line.
haveConstText	bool	Flag set if the line has constraint text attributes.
constraintText	string	Only if <i>haveConstText</i> set: The constraint text of the line.
XPosition	integer	Only if <i>haveConstText</i> set: The x coordinate for the left top position of the constraint text.
YPosition	integer	Only if <i>haveConstText</i> set: The y coordinate for the left top position of the constraint text.
haveOrdered	bool	Flag set if the line has the Or- dered attribute.
XPosition	integer	Only if <i>haveOrdered</i> set: The x coordinate for the left top position of the Ordered text.
YPosition	integer	Only if <i>haveOrdered</i> set: The y coordinate for the left top position of the Ordered text.
haveSorted	bool	Flag set if the line has the Sorted attribute.
XPosition	integer	Only if <i>haveSorted</i> set: The x coordinate for the left top position of the Sorted text.

Parameter	Туре	Description
YPosition	integer	Only if <i>haveSorted</i> set: The y coordinate for the left top position of the Sorted text.
haveRevOrdered	bool	Flag set if the line has the re- versed Ordered attribute.
XPosition	integer	Only if <i>haveRevOrdered</i> set: The x coordinate for the left top position of the reversed Ordered text.
YPosition	integer	Only if <i>haveRevOrdered</i> set: The y coordinate for the left top position of the reversed Ordered text.
haveRevSorted	bool	Flag set if the line has the re- versed Sorted attribute.
XPosition	integer	Only if <i>haveRevSorted</i> set: The x coordinate for the left top position of the reversed Sorted text.
YPosition	integer	Only if <i>haveRev</i> Sorted set: The y coordinate for the left top position of the reversed Sorted text.
derived	bool	Flag set if the line shall be marked as derived.
numPoints	integer	The number of breakpoints for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exists having the given *fromID* and *toID*. There will be *numPoints x* and *y*-values at the end of the message. Text position and line coordinates are adjusted to the nearest grid point but it will not be checked whether line endpoints are placed on
the symbol boundary (the rectangle enclosing the symbol), as they are expected to be. Extra line segments needed for e.g. a signal route entering a service will be automatically generated.

•	GeneralizationLine (115)
---	---------------------------------

Parameter	Туре	Description
fromID	integer	The objectId for the symbol where the line starts.
toID	integer	The objectId for the symbol where the line ends.
haveName	bool	Flag set if the line has a name attribute.
name	QuotedString	Only if <i>haveName</i> set: The name of the line.
XnamePosition	integer	Only if <i>haveName</i> set: The x coordinate for the left top position of the name.
YnamePosition	integer	Only if <i>haveName</i> set: The y coordinate for the left top position of the name.
numPoints	integer	The number of breakpoints for the line. There must be at least 2.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exists having the given *fromID* and *toID*. There will be *numPoints x* and *y*-values at the end of the message. Text position and line coordinates are adjusted to the nearest grid point.

• Class LinkLine(123)

Parameter	Туре	Description
fromID	integer	The objectId for the line where the link line starts.
toID	integer	The objectId for the symbol where the line ends.
numPoints	integer	The number of breakpoints for the line. Always 2 since no breakpoints are al- lowed on this line.
x	integer	x coordinate for a point.
У	integer	y coordinate for a point.

It is checked that symbols exists having the given *fromID* and <u>toID</u>. Observe that the *fromID* must denote an association or aggregation line. Note that *numPoints* always must be 2, and consequently there will be two x and y coordinates.

Text Editor Services

Show Position

Description

The *Show Position* service makes sure that a certain character specified by line and column is visible in a text editor window:

- If no window is displaying the document identified by bufferId, the buffer is displayed in a window according to the user's window reuse policy.
- A window is brought to the front.
- The text in the window is re-centered so that the indicated character is displayed.

Note that the actual display position after re-centering is undefined if the line and/or column values do not make sense.

Tools Supporting the Service

SET_TE

Service Request

SETESHOWPOSITION

Parameter	Туре	Description
bufid	integer	Identifies the buffer
line	integer	Indicates the line that should be shown
column	integer	Indicates which character on the line which should be shown

Service Reply

SETESHOWPOSITIONREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Buffer id does not exist

Select Text

Description

The *Select Text* service selects a range of text in a window displaying a given text buffer.

- If no window is displaying the document identified by bufferId, the buffer is displayed in a window according to the user's window reuse policy.
- A window is brought to the front.
- The text in the window is re-centered so that the indicated character is displayed.

Note that the actual selection is undefined if one or more of the line and/or column values do not make sense.

Tools Supporting the Service

SET_TE

Service Request

SETESELECTTEXT

Parameter	Туре	Description
bufid	integer	Identifies the buffer
line1	integer	Indicates where the selection should start
column1	integer	Indicates the character where the selection should start.
line2	integer	Indicates the line where the selection should end.
column2	integer	Indicates the character where the selection should end.

Service Reply

SETESELECTTEXTREPLY

Parameter	Туре	Description
status	integer	Service reply status.

Errors

Buffer id does not exist

Notifications

The *notifications* inform the environment when something significantly in Telelogic Tau happens. Notifications are only available as PostMaster messages.

To receive a notification one must subscribe on that particular notification (or on all notifications). This is accomplished either statically, by an entry in an additional configuration file, or dynamically by the Telelogic Tau services <u>Add Tool</u> and <u>Add Tool Subscription</u>.

Overview of Available Notifications

MessageDescriptionSESTARTNOTIFYWhen a tool is started.SESTOPNOTIFYWhen a tool is stopped.

Tool notifications

Diagram notifications

Message	Description
SELOADNOTIFY	When a diagram is loaded.
SEUNLOADNOTIFY	When a diagram is unloaded.
SEDIRTYNOTIFY	When a diagram becomes modified.
SESAVENOTIFY	When a diagram is saved.
SESDLENEWNOTIFY	When an SDL diagram is created.
SEMSCENEWNOTIFY	When an MSC diagram is created.
SEOMENEWNOTIFY	When an OM diagram is created.
SETENEWNOTIFY	When a text diagram is created.

Start Notify

Description

This message is broadcast when a new tool is started. That is, when it connects to the PostMaster. It is sent automatically by the <u>SPInit</u> function.

Tools Issuing the Notification

The started tool

Notification

SESTARTNOTIFY

Parameter	Туре	Description
toolType	integer	The kind of tool that was started.
argv0	string	The filename (with a complete path) of the started tool as obtained by reading argv[0].

Stop Notify

Description

This message is broadcast when the tool disconnects from the PostMaster and terminates.

Tools Issuing the Notification

The tool which stops

Notification

SESTOPNOTIFY

Parameter	Туре	Description
toolType	integer	The kind of tool that stopped.

Load Notify

Description

Broadcast when the diagram is loaded in an editor.

Tools Issuing the Notification

SET_SDLE SET_MSCE SET_OME SET_TE

Notification

SELOADNOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the loaded diagram.
fileName	string	Name of the loaded file.
diagramType	integer	Diagram type of the loaded diagram.
diagramName	string	Diagram name of the loaded diagram.

Unload Notify

Description

Broadcast when the diagram is unloaded in the editor. No assumptions could be made whether or not the diagram was saved. The buffer id of the unloaded diagram is then not longer valid and cannot be used anymore.

Tools Issuing the Notification

SET_SDLE SET_MSCE SET_OME SET_TE

Notification

SEUNLOADNOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the unloaded diagram.

Dirty Notify

Description

Broadcast when a diagram becomes dirty, i.e. when the user has modified the diagram.

Tools Issuing the Notification

SET_SDLE SET_MSCE SET_OME SET_TE

Notification

SEDIRTYNOTIFY

Parameter	Туре	Description	
bufId	integer	Buffer id of the diagram that became dirty.	

Save Notify

Description

The message is broadcast when a diagram is saved.

Tools Issuing the Notification

SET_SDLE SET_MSCE SET_OME SET_TE

Notification

SESAVENOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the saved diagram.
fileName	string	The filename in which the diagram was saved.

SDL New Notify

Description

Broadcast when a diagram is created in the SDL Editor. In this case a *Load Notify* is not broadcast.

Tools Issuing the Notification

SET_SDLE

Notification

SESDLENEWNOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the new diagram.
virtuality	string	Kind of virtuality
diagramType	integer	Type of diagram.
diagramName	string	Name of the diagram.
qualifier	string	A qualifier for the new diagram. The qualifier is not a full qualifier, since it does not include the dia- gram path to the root diagram.

MSC New Notify

Description

Broadcast when a diagram is created in the MSC Editor. In this case a *Load Notify* is not broadcast.

Tools Issuing the Notification

SET_MSCE

Notification

SEMSCENEWNOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the new diagram.
diagramName	string	Name of the new diagram.

OM New Notify

Description

Broadcast when a diagram is created in the OM Editor. In this case a *Load Notify* is not broadcast.

Tools Issuing the Notification

SET_OME

Notification

SEOMENEWNOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the new diagram.
diagramName	string	Name of the new diagram.

TE New Notify

Description

Broadcast when a diagram is created in the Text Editor. In this case a *Load Notify* is not broadcast.

Tools Issuing the Notification

SET_TE

Notification

SETENEWNOTIFY

Parameter	Туре	Description
bufId	integer	Buffer id of the new diagram.
diagramName	string	Name of the new diagram.

Auxiliary Messages

Communicating with Simulators

Description

SESDLSIGNAL is sent by an SDL simulator when it sends an SDL signal to its SDL environment. Other simulators (or any applications connected to the PostMaster) can then receive the message and send back messages of the same type.

The PIds consist of a UNIX PId **(on UNIX)**, or an identifier "pid" known by the PostMaster **(in Windows)**, in both cases an integer, and an SDL process instance number (a hex number), which normally is used only in comparisons. The PIds may be used to uniquely identify a single receiver if several receivers exist, but can be ignored if only two tools are communicating

Tools Issuing and Recognizing the Message

SET_SDLENV

Message

SESDLSIGNAL

Parameter	Туре	Description
signalName	integer	SDL name of the sent signal
globalReceiverPid	integer	See above.
localReceiverPid	string	See above.
globalSenderPid	integer	See above.
localsenderPid	string	See above.
signalParam	string	Sea above.

Chapter **12** The Telelogic Tau Public Interface

PostMaster Operation Failed

Description

SEOPFAILED is sent by the PostMaster when an internal problem inhibited a message to be handled properly. Three major categories of errors exists:

- Communication error, i.e. the underlying network (e.g. UNIX socket) reported an error
- An explicit address did not exist
- An operation requested from the PostMaster failed.

Tools Issuing and Recognizing the Message

SET POST

Message

SEOPFAILED

Parameter	Туре	Description
message	integer	The message causing the error.
severity	integer	Kind of error.
pid	integer	The PId of the subscriber (if a spe- cific subscriber was involved).
errorCode	integer	The error itself.
messageParam	string	The remaining information is the data (or the first part of the data) of the message causing the error.