HP Service Manager Software

For the Windows® and Unix® Operating Systems

Software version 9.20

Wizards Guide

Document Release Date: June 2010

Software Release Date: June 2010

Legal Notices

Warranty

The only warranties for HP products and services are set forth in the express warranty statements accompanying such products and services. Nothing herein should be construed as constituting an additional warranty. HP shall not be liable for technical or editorial errors or omissions contained herein.

The information contained herein is subject to change without notice.

Restricted Rights Legend

Confidential computer software. Valid license from HP required for possession, use or copying. Consistent with FAR 12.211 and 12.212, Commercial Computer Software, Computer Software Documentation, and Technical Data for Commercial Items are licensed to the U.S. Government under vendor's standard commercial license.

Copyright Notices

Trademark Notices

Java™ and all Java based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the U.S. and other countries.

Microsoft® and Windows® are U.S. registered trademarks of Microsoft Corporation.

Oracle® is a registered US trademark of Oracle Corporation, Redwood City, California.

Unix® is a registered trademark of The Open Group.

Support

Visit the HP Software Support Online web site at:

http://www.hp.com/go/hpsoftwaresupport

This web site provides contact information and details about the products, services, and support that HP Software offers.

HP Software online support provides customer self-solve capabilities. It provides a fast and efficient way to access interactive technical support tools needed to manage your business. As a valued support customer, you can benefit by using the support web site to:

- Search for knowledge documents of interest
- Submit and track support cases and enhancement requests
- Download software patches
- Manage support contracts
- Look up HP support contacts
- Review information about available services
- Enter into discussions with other software customers
- Research and register for software training

Most of the support areas require that you register as an HP Passport user and sign in. Many also require a support contract. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

To find more information about access levels, go to:

http://h20230.www2.hp.com/new_access_levels.jsp

Documentation Updates

The title page of this document contains the following identifying information:

- Software Version number, which indicates the software version.
- Document Release Date, which changes each time the document is updated.
- Software Release Date, which indicates the release date of this version of the software.

To check for recent updates or to verify that you are using the most recent edition of a document, go to:

http://h20230.www2.hp.com/selfsolve/manuals

This site requires that you register for an HP Passport and sign in. To register for an HP Passport ID, go to:

http://h20229.www2.hp.com/passport-registration.html

Or click the **New users - please register** link on the HP Passport login page.

You will also receive updated or new editions if you subscribe to the appropriate product support service. Contact your HP sales representative for details.

Contents

Contents 5

1 Wizards Overview 9

What this document covers 9

Prerequisites 9

The functions and purpose of wizards 9

Wizard flow 10

Wizards versus scripts 11

2 Using the Wizard Tool 13

Wizard order of execution 14

Wizard Info tab field definitions 16

File Selection tab field definitions 17

Usage tab field definitions 18

Actions tab field definitions 22

Messaging tab field definitions 25

Variables tab field definitions 25

Next Wizard tab field definitions 26

Previous Wizard tab field definitions 26

Comments tab field definitions 26

Cancel Expressions tab field definitions 27

3 Calling Wizards 29

Call a wizard from menus 29

Call a wizard from display options 30

Call a wizard from Format Control 31

Call a wizard from process records 31

4 Example: Creating a wizard that creates a new SLA 33

Adding an SLO to the SLA 33

Designing the SLA wizard 34

Planning the SLA wizard panels 36

Building the SLA wizard 38

Step 1: Prompt the user to select SLA type 38

Create a wizard for SLA type 39

Create a form for SLA type 41

Step 2: Prompt for customer information 42

Create a wizard to gather customer information. . . . 43
Create a form to gather customer information. . . . 45
Step 3: Prompt for SLA title and description. . . . 47
Create a wizard for SLA description. . . . 48
Create a form for SLA description. . . . 50
Step 4: Prompt for effective dates. . . . 51
Create a wizard for SLA effective dates. . . . 52
Create a form for SLA effective dates. . . . 54
Step 5: Ask about agreements and other documents. . . . 55
Create a wizard to ask about attachments. . . . 56
Create a form to ask about attachments. . . . 58
Step 6: Prompt to add agreements. . . . 60
Create a wizard to add agreements. . . . 61
Create a form to add agreements. . . . 63
Step 7: Prompt to attach other documents. . . . 64
Create a wizard to attach other documents. . . . 65
Create a form to attach other documents. . . . 67
Step 8: Save new SLA and prompt for SLOs. . . . 68
Create a process to save the new SLA. . . . 69
Create a wizard to check SLO requirements. . . . 70
Create a form to check SLO requirements. . . . 73
Step 9: Call the Availability SLO wizard. . . . 75
Create a wizard to call an Availability SLO. . . . 75
Step 10: Prompt for another Availability SLO. . . . 77
Create a wizard for more Availability SLOs. . . . 78
Create a form for more Availability SLOs. . . . 80
Step 11: Call the Response SLO wizard. . . . 81
Create a wizard to call a Response SLO. . . . 82
Step 12: Prompt for another Response SLO. . . . 83
Create a wizard for more Response SLOs. . . . 84
Create a form for more Response SLOs. . . . 86
Step 13: Display new SLA number and title. . . . 88
Create a wizard for the completed SLA. . . . 89
Create a form for the completed SLA. . . . 91
Step 14: Set menu to call the new SLA wizard. . . . 92
5 Troubleshooting wizards. . . . 95

Button options are not displaying properly. 95
JavaScript syntax errors. 96
Cause. 96
Fix. 96
File variables not passed into wizard flow. 97
Cause. 97
Fix. 97
Format not found. 97
Cause. 97
Fix. 97
Records are not sorted properly. 97
Cause. 97
Fix. 97
Query failed to provide correct results. 98
Query returned a list of records rather than a single record. 98
Cause. 98
Fix. 98
RAD expression syntax errors. 98
Cause. 98
Fix. 99
Wizard stops with "Could not continue" errors. 100
Run a trace. 101
Variables to Avoid Using in Wizards. 103

1 Wizards Overview

Wizards provide a powerful way to gather user input and help your users quickly and efficiently accomplish many different tasks within Service Manager. The Service Manager applications include many out-of-box wizards, and the Service Manager tailoring tools include a Wizard creation tool that enables you to design and build your own custom wizards that can be called from the various Service Manager applications and modules.

What this document covers

This document discusses the purpose and functions of wizards. It describes the Wizard tool that you use to create new wizards, and defines and describes each of the fields the tool includes. A detailed example walks you through all of the steps involved in planning and building a sample wizard that creates a new Service Level Agreement. Troubleshooting information and common errors are also explained.

Prerequisites

The Wizard tool provides a powerful way to configure the Service Manager environment. However, it also requires a solid understanding of the following areas, which are covered in other parts of the Service Manager tailoring documentation:

- Designing and creating forms
- Using the syntax of the Service Manager system language
- Performing traces for debugging purposes
- Creating processes and using the Document Engine

The functions and purpose of wizards

The Wizard creation tool enables implementers and administrators to add wizards within Service Manager that assist users with certain tasks, such as adding contacts to a database, escalating a Service Desk ticket to the applicable Service Manager module, and creating a new configuration item type for Configuration Management.

Wizards permit multiple options during the creation process. Functionality in a wizard record includes:

• Creating records

• Selecting records from a database

• Modifying the current record

• Executing expressions (RAD or JavaScript), processes, and format control

• Asking for user input

• Stacking wizards to run in succession

Service Manager includes a number of embedded, out-of-box wizards that simplify repetitive tasks. These wizards offer a variety of functionality, such as:

• Create a User (the User Quick Add Utility)

• Escalate an Interaction

• Reject a Change

• Add a CI Type

• Add a Contract

You can view the wizards that are shipped with Service Manager by clicking **Tailoring > Wizards**, and then clicking the **Search** button.

In addition to simplifying repetitive user tasks, Wizards are very useful for processes that require the user to supply prerequisite information. For example, during the interaction creation cycle you could present a wizard panel flow for the operator to enter data. Based on how the caller replies to questions regarding the problem, the wizard determines which panels the operator sees next. While the wizard is executing, the operator-entered data may be accumulated in a file variable that is returned to the calling application when the wizard is complete.

The wizard can display a form and execute a Format Control record or process. The instructions in the Format Control record where there is a condition on "initial" that evaluates to true will be executed before the form opens. The instructions where there is a condition on "add" that evaluates to true will be executed after the user clicks OK.

Wizards are also very helpful when multiple complex decisions must be made in order to reach a conclusion. For example, Change Management approval requirement conditions are normally based on the data contents of one field in the change record. An example of such a condition is `risk.assessment in $L.file="1."` However, there may be circumstances where the approval requirement condition is based on the values of several different fields. For instance, there may be three fields that affect the approval requirements: division, area, and department. Hard coding all the possible combinations of these field values into condition statements requires a great deal of work and would be difficult to maintain. You can define these conditions in wizard records that do not display forms but that allow you to call a subroutine or execute standard Service Manager processing statements (similar to Format Control calculation statements). These options allow for the manipulation of record data. At runtime, the wizard acts as a decision tree that results in significant processing reductions compared to that involved in the original method of coding.

Wizard flow

Wizard flow defines the order in which wizard panels execute. Wizard panels can move in a straight line from start to finish, or branch into several different process flows.

The more complex wizard flows use condition statements that must evaluate to true before certain wizard panels can execute. If none of the specified conditions evaluate to true, the wizard flow is considered complete, and control passes back to the calling application.

It is very helpful to develop a map of the entire flow showing the name of the form (if any) displayed by each wizard panel and the conditions controlling the flow from panel to panel. You can follow the map throughout the wizard creation process to avoid simple errors that could prevent your wizard from operating properly.

Wizards versus scripts

Service Manager wizards have much in common with the Service Manager scripts utility, a legacy tool that has been part of the system tailoring utilities for many releases. Wizards go beyond scripting, enabling much greater functionality and flexibility.

A major advantage of wizards over scripts is security. Wizards are more robust and ensure data integrity, which scripts cannot do.

Unlike scripts, wizards allow you to specify the format control and display screen; you are not limited to the format control for a particular form, or to a particular display screen.

In addition, wizards themselves can call Format Control records and processes. Wizards do not allow direct calls to RAD applications.

2 Using the Wizard Tool

Before beginning to create a wizard, you should have a basic plan for the wizard flow, including the function that each wizard panel must perform. This requirement is demonstrated for the example of planning the SLA wizard panels for a new SLA wizard.

When you are ready to create the wizard, you use the Wizard creation tool to create each successive panel of the complete wizard. Generally each panel represents a screen or step of the wizard that the end user will see, although some panels are not visible to the user and instead perform background activities, such as calling a Document Engine process.

Best Practice for variables: It is best to use local variables in a wizard in order to eliminate the chance of a variable collision between threads (or even in the same thread if the wizard calls an external application). However, variables local to the wizard application cannot be used by the display application or in Process records called by the wizard.

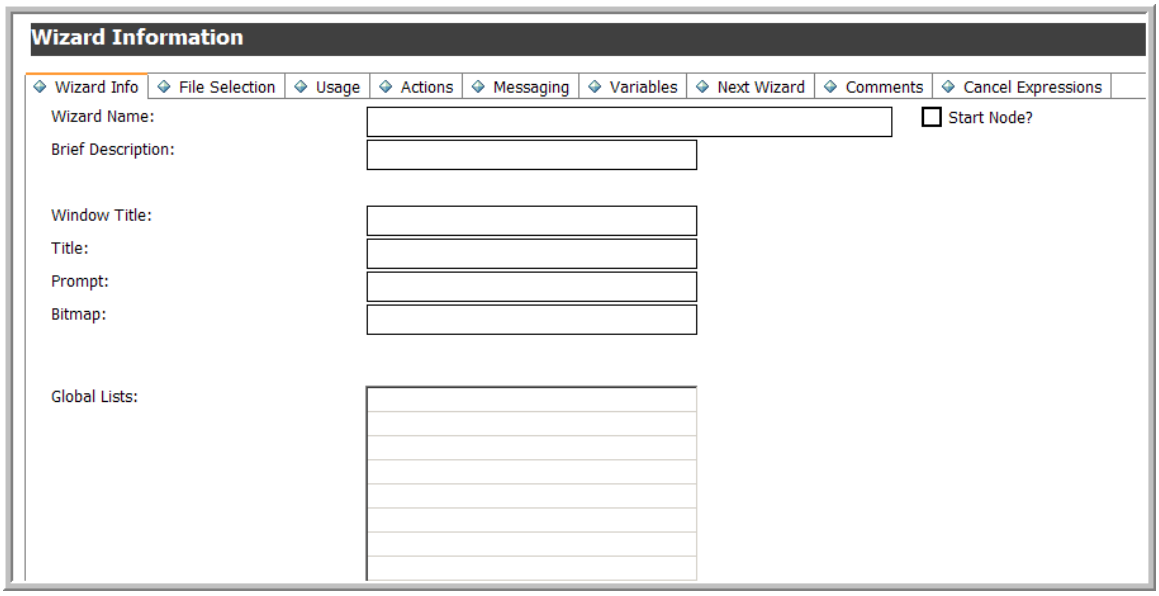
Specifying a local variable as a wizard variable increases the scope of the variable to include the display application and called Processes. The variable will automatically be cleaned up by the server when the entire wizard flow is complete, so there is no danger of namespace collisions with variables from other threads or processes.

To access the Wizard creation tool, click **Tailoring > Wizards** or type `wizards` in the Service Manager command line field.

The Wizard creation tool displays ten tabs:

- **Wizard Info**
- **File Selection**
- **Usage**
- **Actions**
- **Messaging**
- **Variables**
- **Next Wizard**
- **Comments**
- **Cancel Expressions**
- **Previous Wizard** (only visible after you click **Add**)

The wizard creation tool interface is shown below:



The remainder of this chapter discusses the order of execution of the tabs in a wizard and how to use each of the fields in the Wizard tool tabs to create wizard panels.

Wizard order of execution

When a wizard is run, it executes in the order the tabs are displayed in the wizard tool interface (Wizard Info, File Selection, Usage, and so on.)

The diagram labeled Order of Execution of Wizard Tabs illustrates this order. On the Usage tab, the developer determines whether the wizard panel is displayed to the end user. If the panel is displayed, the order of execution then depends on which button the end user clicks, Next, Previous, Cancel, or Finish, as indicated in the diagram.

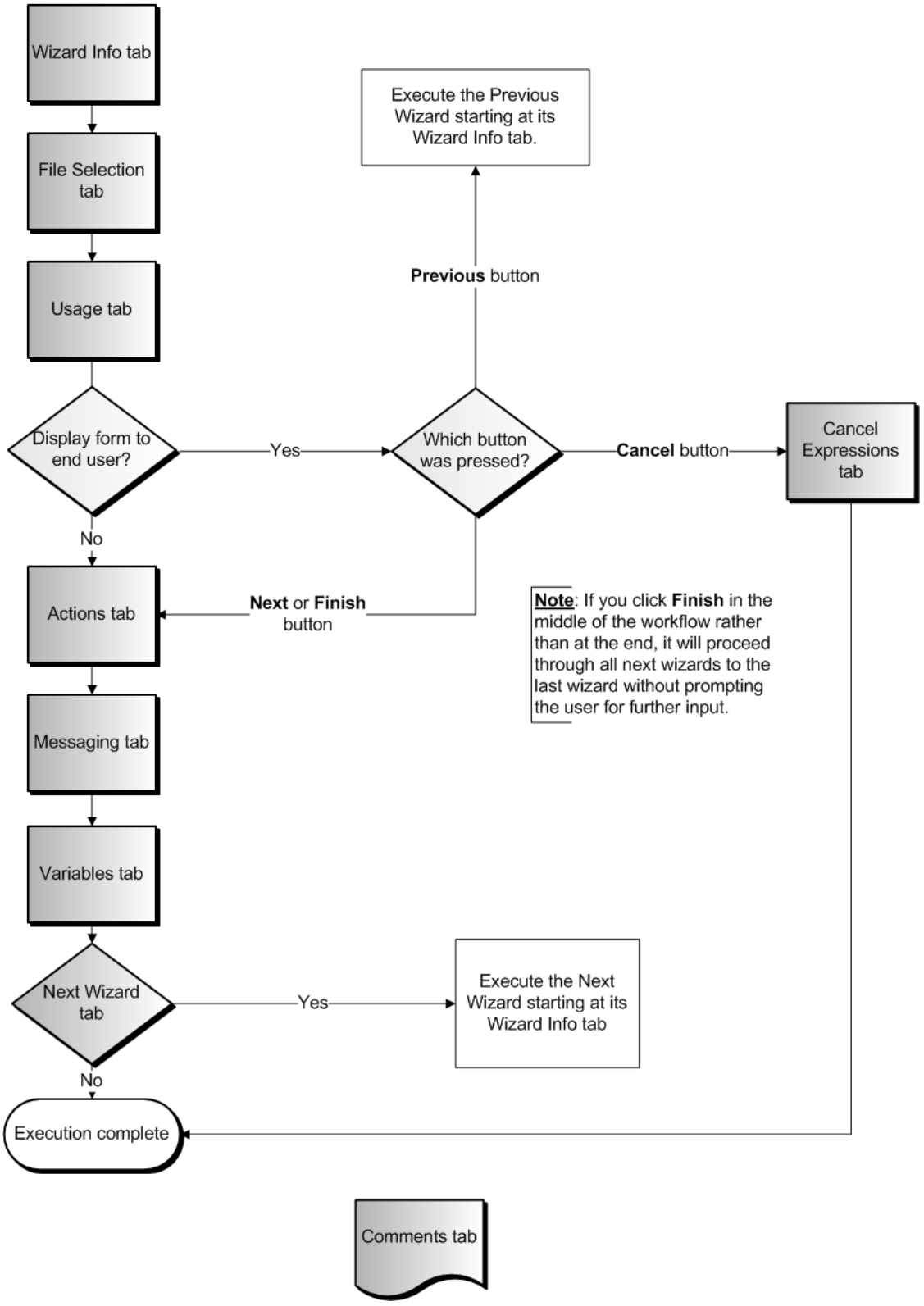
If the end user clicks the Previous button, the previous wizard panel is displayed, or some other panel as set by the developer using the Previous Wizard tab in the wizard tool.

The Finish button is intended to be used at the end of the workflow. If the user clicks it in the middle of the workflow, the wizard will proceed through the rest of the sequential wizard panels without prompting the user for further input.

The Comments tab is used for internal development comments, and has no effect on the wizard processing flow.

Within the File Selection tab, the Initital Expressions and Javascript evaluate *after* the **Select $L.file by** tab. The means that you cannot populate a query variable in Initial Expressions and then use it in "Select $L.file by."

Order of Execution of Wizard Tabs



Wizard Info tab field definitions

Use this tab to enter basic information such as the wizard name, description, window title, and images to be included on the panels displayed to the end user.

Field	Description
Wizard Name *name*	This field forms the unique key for the wizard in the wizard dbdict. Enter a brief descriptive name for the wizard as the unique identifier for this wizard record. As a best practice, include the table (or module) for the wizard, the action the wizard panel performs, and other detail such as whether the wizard is the Start Node. For example, your first wizard panel name for creating a wizard panel for SLA could be something like "SLA Create 1 - Start - Type." This is a required field.
Start Node? *start*	Selecting this checkbox (setting it to true), indicates that this is the first in a series of wizard panels. When a wizard set as the Start Node also requests user input, the Previous button is automatically disabled when the wizard panel is displayed to the user. **Note**: If the Start Node wizard is not displayed to the user, be sure to disable the Previous button on the first wizard panel that is displayed.
Brief Description *brief.description*	Enter a brief, meaningful description of the wizard, including any important identifying information.
Window Title *window.title*	The window title is displayed on the tab that holds the wizard thread when displaying the wizard panel to the user. This field also takes a message number from the scmessage database. For example, scmsg(18,"wizard"). **Note**: If the wizard you are creating will be localized (translated to another language or languages), then using the scmessage database is the preferred method.
Title *title*	The title of the wizard entered in this field appears in bold in the upper left of the wizard panel when displayed. The title can be entered as a message number from the scmessage database such as scmsg(18,"wizard"). Enter a title that makes indicates the purpose of the panel to the user, such as "Create SLA." If more detail is needed, you can enter something like "Create SLA - Enter Description."
Prompt *prompt*	Enter a prompt that instructs the end user if this is a wizard panels that will be displayed to the user. HP Service Manager displays this information in the center of the panel, between the title and the wizard subformat, to give the user more detailed instructions on what kind of information should be entered in the panel. The prompt should be in large type. **Note**: Localizable messages, such as scmsg(18,"wizard"), can be used for the titles and prompt rather than hard-coding the wizard with text. This allows for localized languages in a multi-lingual environment.

Field	Description
Bitmap *bitmap*	The image to display on the left side of the wizard panel. The default image is located in the Service Manager Install Directory at: `Client\plugins\com.hp.ov.sm.client.common_<SM version and build number>\src\resources\icons\obj16\wizard<1,2,3>.gif` If you want to use a different image, add it to the installation path above and enter the image file name in this field.
Global Lists *global.list*	Click Fill to select the global lists to be built or updated so that they are available when the wizard is run.

File Selection tab field definitions

Use this tab to designate system language or JavaScript code to be run before file selection and determine the content of $L.file. The tab includes three subtabs, described below. The **Select $L.file by** subtab includes four main options, listed from top to bottom as they appear on the screen.

Field	Description
Initial Expressions subtab *init.expressions*	Enter expressions using system language syntax that will be evaluated after the $L.file is initialized. Use these expressions to initialize variables.
JavaScript subtab *javascript.init*	Enter the JavaScript code that will run after the $L.file is initialized. Use this section to initialize variables or select values from related tables.
Select $L.file by subtab	You must choose one of the four main options listed below to designate how the wizard will select the $L.file variable.
No $L.file (use typecheck) *file.selection*	This option means that the wizard should not expect a file variable to be passed in from wherever the wizard is called; in fact, if one is passed in, it will be ignored. Instead it creates $L.file by initializing it as a record in the "typecheck" table, which is an internal table that contains fields corresponding to different field types supported by the system. This is often used in simple wizards that require only a small amount of input that does not need to be saved into the database.
$L.file passed in *file.selection*	Selecting this option indicates that the file variable ($L.file) was passed into this wizard as a parameter from the calling application or wizard. If a wizard is called as a next wizard, the file variable is automatically passed on. This option should be selected if you plan to continue to work on the current record stored in $L.file.

Field	Description
Create a record *file.selection*	Selecting this option indicates that the wizard will create a new record. You must enter or select the dbdict name in the record of type field. This initializes $L.file with the structure of the record from the table selected in the "of type" option.
of type (for Create a record) *create.record.filename*	Click Fill to select the dbdict that defines the record to be created.
Select records *file.selection*	Select this option to use a query statement to select one or many records.
of type (for Select records) *select.record.filename*	This field determines the record type for a query. Enter the dbdict name of the table from which the records should be selected, such as location. Click Fill to select a name from a list of all dbdict names.
using query *select.query*	Enter a query, using system language syntax, to search for records that will be used to select records. For example: `"agreement.id="+$G.new.sla` **Note**: The quotes (") are required.
Resolve Variables *resolve.variables*	Select this box (set to true) to ensure that any variables entered in the Select Records field are evaluated before running the query.
No Records Message *select.no.records*	The text to display to the user if the query returns no records. This field also takes a message number from the message database. For example: `scmsg(111,"wizard")`.

Usage tab field definitions

Use this tab to set wizard usage options, such as whether and which subformat will be displayed to the user and which buttons will be available on the panel displayed.

Field	Description
Wizard Usage section	Determines the action that HP Service Manager should take when the wizard is run. You must select one of the five main options presented on the screen.
Select one record from list *wizard.type*	Choose from the **Selection Criteria** listed below.
Selection Criteria	When you choose to **Select one record from list**, selecting one of the following options allows you to select a record from a list. The list is formed based on the options selected.

Field	Description
Use $L.file as $L.selection *use.file.as.selection*	This option uses the $L.file variable to form the list of records where the selection is to be performed.
Query for records *query.for.records*	This option allows you to specify a query to form the list of records. The fields that are needed for the query are noted below.
of type *query.for.records.filename*	Enter the name of the table (dbdict name) that the query should run against. Check that the dbdict for field names is valid for the query.
using query *query.for.records.query*	This option is used in conjunction with the options "Query for Records" and "of type." Enter the query using Service Manager system language syntax.
sort by *query.for.records.sort*	This option allows you to present the list, sorted in order, based on the fields specified in this option. Enter field names that are valid for the selected dbdict and sorted in ascending order.
If no records *query.select.no.records*	This option allows you to specify an action if there are no records on the list. The actions are as follows: **return**: Returns to the previous wizard panel if there is one, or to the place from which the wizard was called (if this is the first panel). **continue**: Shows the empty list and allows you to continue. **skip display**: Skips the wizard panel and goes to the following panel, if applicable.
No records message *query.no.records.msg*	This option allows you to display a message to present if there are no records in the list. An scmessage record can be used to construct the message.
If one record *query.select.one.record*	This option allows you to specify an action if there is just one record in the list. The one record can be displayed in a list, or it can be used as the current file variable without being presented to the user for selection.
Allow "Skip" option? *allow.skip*	If selected (set to true), enables users to skip the panel.
Request user input *wizard.type*	This field indicates that the user will be prompted for information. When this option is selected, the **Sub Format to display** entry becomes required and a prompt with instructions needs to be entered in the **Wizard Info** tab.
Skip Display *wizard.type*	Skip Display should be used when the current wizard manipulates the current record without the need for user input. Typically, a wizard that is not displayed to the user is part of a complex wizard workflow.

Field	Description
Cancel Immediately *wizard.type*	This option is used to cancel a process immediately if a wizard is started by a user type who is not intended to use the wizard. For example: A company creates a wizard to gather information from top-level management customers. Let us say that a user who does not fit the criteria chooses to start the wizard and make his way through the wizard panels. The Cancel Immediately option causes the process to immediately end when it becomes clear that the information does not need to be gathered from that user.
Select multiple records from list *wizard.type*	Selecting this option displays a list and allows the user to select multiple records from the list. The list is formed based on the options selected in the "Selection Criteria" section.
Multiple Selection field *mult.field.name*	This option determines which field in the list becomes available for the user to select multiple records from.
Sub Format to Display *sub.format*	Enter the name of the format for user input to display within the wizard format.

Field	Description
Main Format (Defaults to Medium) *form.Name*	The wizard creation tool offers several format choices that effect the size of the wizard and the placement of buttons. HP Service Manager offers the following wizard format options. If no format is selected, the default is "Medium." **Small (wizard.small)** - This wizard offers the least amount of space for text and options, but also requires the least amount of screen space. This wizard is ideal for clients running in low resolution environments. It allows only a small space for the subformat, so the subformats need to be kept small. Buttons available are **Previous**, **Next**, **Finish**, and **Cancel**, and placement is horizontally-aligned along the bottom. **Medium (wizard)** - This wizard offers sufficient space for most wizard tasks. Buttons available are **Previous**, **Next**, **Finish**, and **Cancel**, and placement is horizontally-aligned along the bottom. **Large (wizard.large)** - This wizard offers the most amount of space for text and options, but also requires the most amount of screen space. This wizard is ideal for clients running with high screen resoltuions. It offers the most amount of space for the subformat to display, allowing for more complex subforms. Buttons available are **Previous**, **Next**, **Finish**, and **Cancel**, and placement is horizontally-aligned along the bottom. **OK Only - Small (wizard.okonly.small**) - Uses the small wizard size and limits the buttons available to navigate through the task. Buttons available are **OK** and **Cancel**, and placement is horizontally-aligned along the bottom. **OK Only - Medium (wizard.okonly)** - Uses the medium wizard size and limits the buttons available to navigate through the task. Buttons available are **OK** and **Cancel**, and placement is horizontally-aligned along the bottom. **Buttons on Top (wizard.buttonsOnTop)** - Uses the medium wizard size and places the buttons along the top of the wizard. Buttons available are **Previous**, **Next**, **Finish**, and **Cancel**, and placement is horizontally-aligned along the top. **Buttons on Side (wizard.buttonsOnSide)** - Uses the medium wizard size and places the buttons along the right edge of the wizard. Buttons available are **Previous**, **Next**, and **Finish**, and placement is vertically-aligned along the right side.
Display Screen (Defaults to wizard.display) *display.screen*	Enter a display screen, if you want to customize the button and menu options available within the wizard. The default display screen is `wizard.display`.

Field	Description
Activate "Finish" option? *allow.finish*	Use this option for the final wizard in the series. Selecting this checkbox makes a Finish button appear on the wizard panel. **Note**: It is possible to have the Finish button available prior to the last wizard in the workflow, if the wizard panels displayed after this wizard only provide optional information. The Finish button cannot be activated before all required information can be collected from the user.
Turn off Next and Previous buttons? *disable.next.previous*	Selecting this checkbox makes the Next and Previous buttons unavailable in the wizard, so the user cannot click "Next" or "Previous" to move forward or backward in the wizard. **Note**: This selection should only be made on the very last wizard panel in the workflow. Also, the Finsih button needs to be activated if the Next and Previous buttons are disabled.
Turn off Next button only? *disable.next*	Selecting this checkbox makes the Next button unavailable in the wizard, so the user cannot click "Next" to move forward in the wizard.
Turn off Previous button only? *disable.previous*	Selecting this checkbox makes the Previous button unavailable in the wizard, so the user cannot click "Previous" to move backward in the wizard. **Note**: This option should be used if the first displayed panel of the wizard is not the Start Node. Selecting Previous in such a case would exit the wizard without going through proper cancel processing.
Use Conditional Previous Exits? *previous.cond*	When you select this option, the Previous Wizard tab appears on the form. In the Previous Wizard tab, you define the different possible exits when the Previous button is clicked.
Never prompt on cancel?	Determines whether the user is prompted when canceling out of the wizard. The Return prompt can send the user back to the panel.

Actions tab field definitions

Use this tab to set wizard actions, including defining which Process or Format Control will be executed and on which file variable or record they will be performed.

Field	Description
Initial Process *initial.process*	This option lets you specify a process definition to be executed prior to the actions to be performed. For information on process definitions, see the *Document Engine* documentation.

Field	Description
Perform Actions On section	You must select one of the three options to specify which record(s) will be affected by the action you are going to perform. The file is either the current file, the whole list selected, or an action to be performed on every single record selected. Actions are defined in the wizard, and define what will happen when a user clicks **Next** on a wizard. **Note**: Depending on what you selected in the **File Selection** tab, you can choose whether the actions performed when a user clicks **Next** will be performed on the current file or a record selection. Record selection ($L.selection) must only be chosen if the $L.selection file variable was filled when you chose Select Records on the **File Selection** tab, and if you chose either "select one" or "select multiple records" from the list on the **Usage** tab.
Current File ($L.file) *perform.action.on*	The actions will be performed on the single record represented by the $L.file file variable. The current file may have been passed in, created, or selected with a query returning only a single record. **Note**: The actions defined in the Actions to Perform section apply to the current file selected in the **File Selection** tab and/or **Usage** tab.
Selection ($L.-selection) *perform.action.on*	The actions defined in the Actions to Perform section apply to the group of records selected in the **File Selection** tab.
Each record in selection ($L.selection) *perform.action.on*	The actions defined in the Actions to Perform block apply to each record selected.
Actions to Perform section	Use the three subtabs in this section to indicate the actions to perform on the selected file.
Expressions subtab *expressions*	Enter any expressions that should be run as part of the wizard after the initial process, before the JavaScript and Format Control (or process). These expressions use Service Manager system language syntax, such as: `name in $L.file="Test"` These expressions are evaluated before the Reset Current File to Selection action is performed. Therefore, actions performed here should use $L.selection rather than $L.file if they are intended to manipulate the user selected record. **Note**: These expressions can use the record(s) selected, such as storing information in a field.
JavaScript subtab *javascript.actions*	The JavaScript expressions will run as part of the wizard and are executed after the initial process and expressions, but before the Format Control or process entered on the next subtab. These JavaScript language expressions can be used to manipulate the current file variable or initialize variables.

Field	Description
Format Control/Process Name subtab	
Format Control *format.control*	The Format Control record to use. You can enter a Format Control record to execute queries, calculations, validations, or subroutines after running through the expressions and JavaScript, and before calling the process.
of type *format.control.type*	The Format Control type to use for this wizard to evaluate the applicable conditions, so that Format Control can be executed on the action (**Add**, **Update**, or **Delete**) that is evaluated in the Format Control record.
on bad validation *bad.validation.action*	When data validation in Format Control fails, choose to return the user to the current wizard panel to enter the correct information, or choose to let the wizard continue to the next panel, ignoring the failed validation. *__Return__ sends the user back to the panel. Should only be used if the user has the opportunity to fix the failed validation. If the wizard does not request information from the user by displaying a subformat to enter information, selecting Return will end up in an infinite loop. *__Continue__ lets the user continue.
Process Name *process.name*	Enter the name of a process to execute on this wizard panel. For more information, refer to the Document Engine documentation.
Reset Current File to Selections? *reset.to.selections*	If selected (set to true), replaces the current file variable ($L.file) with the value selected by the user ($L.selection). **Note**: This field depends on the selections in the Usage tab. If "Select one record fromlist" or "Select multiple records from list" are chosen, those options generate $L.selection.
Restart Panel if *restart.condition*	Under some circumstances it may be necessary to run the wizard panel again. For example, an expected field may not have been filled in correctly, or an error code may have been returned from Format Control or the process. Enter a condition that evaluates to true or false to determine if this panel needs to be rerun.
Display Record(s) when complete? *display.when.complete*	Determines whether to display the record to the user when the wizard completes. If the condition is true, HP Service Manager displays the record to the user when complete. If the condition is false, no record is displayed.
Mode *display.mode*	This field determines how a record will be displayed to the user. *__Browse__: (This is the default.) Browse mode does not have any buttons to store a modified record. *__Add__: The Add button becomes avai lable. *__Update__: The Save button becomes available.

Field	Description
Return Current File to calling application? *reset.current.file*	This option returns the current record to the application that called the wizard for further processing.

Messaging tab field definitions

Use this tab to set any messages to be displayed to the end user when specified conditions occur.

Field	Description
Message *message*	Message to be displayed when a certain condition occurs. This field will also take a message number from the scmessage database. For example: `scmsg(106,"fc").`
Condition *message.cond*	Enter a condition for the message that evaluates to true or false to determine if the message should be issued. For example: `not nullsub($L.finish,false)`
Type *message.type*	Select the format of the message to display to users (pop-up or on- screen). The default is on-screen.
Level *message.level*	Select the level of severity of the message: *__Info:__ Informational only *(default)* *__Action needed__ *__Error message:__ The error message issued to the user when a serious error is encountered.

Variables tab field definitions

Use this tab to document the variables used in the wizard workflow.

Field	Description
Wizard variables *wizard.variables*	Use this tab to define variables for use within the wizard. Variables can be passed to applications or formats being created by the wizard, but must be assigned. For example, a variable such as $L.return.action must be defined here, if it will be used within the wizard. The data type and value of a variable can be different at different times, and can have a primitive or compound data type as its value. Service Manager contains three types of variables: 1) local, 2) global, and 3) thread. Local variables begin with $L and persist only within the currently executing RAD application. The server automatically cleans up local variables when it exits a RAD application.

Next Wizard tab field definitions

Use this tab to specify the next wizard panel (if any) to be called.

Field	Description
Wizard Name *next.wizard*	Name of the next wizard to execute within the workflow (series of wizard panels).
Condition *next.wizard.cond*	Enter a condition that evaluates to true or false. The next wizard in the workflow is determined by the first condition that evaluates to true, regardless of the conditions that follow (even if they evaluate to true as well). If all conditions evaluate to false, the wizard workflow will exit and finish.

Previous Wizard tab field definitions

This tab is displayed when you check the **Use Conditional Previous Exits?** checkbox in the Usage tab.

Field	Description
Wizard Name *prev.wizard*	Specify the name of the wizard to which the user should exit.
Condition *prev.wizard.cond*	Set the condition for when the user should be taken to this previous wizard. The condition must evaluate to true or false.

Comments tab field definitions

Use this tab to enter internal development comments about the wizard. These comments have no effect on the wizard workflow itself.

Field	Description
Comments *comments*	Enter any developer comments concerning the wizard here. As a best practice, use the Comments tab to describe details, such as the purpose of the wizard panel, the settings selected to implement that purpose, variables used, wizard workflow, and exit conditions.

Cancel Expressions tab field definitions

Enter system language expressions or JavaScript code that will execute if the end user clicks the Cancel button. This cleans up variables that were used in the wizard to ensure correct behavior the next time the wizard is executed.

Field	Description
Expressions Executed on Cancel subtab *cancel.expressions*	Enter any expressions that will execute when a user cancels the wizard process. This provides the wizard creator an opportunity to reset values or clean up variables initialized in the wizard.
JavaScript Executed on Cancel subtab *javascript.cancel*	Enter any JavaScript code that should be used to reset values and clean up variables that were initialized in the wizard.

3 Calling Wizards

The RAD application in charge of executing wizards is called **wizard.run**. You can call this application from:

- Menus
- Display Options
- Format Control
- Process Records

The sections following this one explain the steps required in calling wizards from these areas.

Wizards themselves can call:

- Format Control records
- Processes
- Other wizards.

Wizards do not allow direct calls to RAD applications.

Warning: It is possible to send a wizard into an infinite loop. For example, this would happen if you called `wizard.run` from a process to execute a wizard that in turn calls the originating process.

Call a wizard from menus

To set up a wizard to be called from menus:

1. Click **Tailoring > Tailoring Tools > Menus**.
2. Select a menu record.
3. Add an option number; a group, if necessary; and a description with the name that you want to display on the menu.
4. Enter **wizard.run** for the application.
5. In the Parameter Name field, specify the name of the parameters to pass to the application. The name of the parameters to pass to the application should be enclosed with quotes "" and curly brackets {}. The available parameters for the `wizard.run` application are: file, name, text, and prompt.
6. In the Parameter Value field, specify the value of the parameter being passed, enclosed with quotes "" and curly brackets {}. For example, `{"createUser Pre"}`.

Example:

In this example, a wizard is set up to be called from the **SYSTEM ADMINISTRATION** menu through the option **Quick Add Utility**. Once the wizard is set up, the user clicks the following:
System Administration > Ongoing Maintenance > User Quick Add Utility.

When the user clicks User Quick Add Utility on the menu, a wizard opens with a predefined set of steps that assist the user in populating information and completing the task of adding an operator.

To see how this wizard has been set up to be called from menus:

1. Click **Tailoring > Tailoring Tools > Menus**.
2. Open the SYSTEM ADMINISTRATION menu record.
3. The Description field displays **User Quick Add Utility**.
4. The Application field specifies **wizard.run**, the RAD application called from the menu.
5. The Parameter Name field specifies the name of a single parameter to pass to the menu item. In this example, {"name"} represents the name of the wizard. The name is enclosed with quotes "" and curly brackets {}.
6. The Parameter Value field specifies the value of the parameter being passed. In this example, {"createUser Pre"}. The name is enclosed with quotes "" and curly brackets {}.

Call a wizard from display options

To set up a wizard to be called from display options:

1. Click **Tailoring > Tailoring Tools > Display Options**.
2. Select the display option that you want to use to call a wizard.
3. Select the **Rad** tab.
4. In the RAD Application field, enter the RAD Application **wizard.run**.
5. The Names field specifies the parameters to pass to the RAD application. The available parameters are: file, name, text, and prompt. The **name** parameter is required.
6. The Values field specifies the values of the parameters being passed to the RAD application.
7. Click **Save**.
8. Click **OK**.

Example: Set up a wizard to be called from display options

In this example, a wizard is set up to be called from the display application **wizard.display.kmsearch**, so that when an operator selects the "Search Again" option, the wizard "KM choose shared content" is called.

To set up a wizard to be called from display options:

1. Click **Tailoring > Tailoring Tools > Display Options**.
2. Select the **wizard.display.kmsearch**, Text Option definition 7500.
3. Select the **Rad** tab.
4. In the RAD Application field, enter the RAD Application **wizard.run**.
5. The Names field specifies the parameters to pass to the RAD application. The available parameters are: file, name, text, and prompt. The **name** parameter is required.
6. In this example, the following values are specified for each parameter named.

Name of parameter	Value of the parameter
file	$L.file
name	KM choose shared content

7. Click **Save**.
8. Click **OK**.

Call a wizard from Format Control

To call a wizard from Format Control:

1. Click **Tailoring > Format Control**.
2. Select the Format Control where you want to call a wizard.
3. Click **Subroutines**.
   The Format Control Maintenance - Subroutines form opens.
4. In the Application field, enter the application **wizard.run**.
5. In the Names field, enter the name of the parameter(s) to pass data to the subroutine application (name, file, text, and prompt).
6. In the Values field, enter the parameters to pass to the subroutine application. The type of parameters in this field must match the type of parameters in the Names field.
7. Specify the condition for each parameter. When a condition is set to "true" and a user clicks Save, Format Control runs every section or expression where the Update field is set to true before it updates the record.

Call a wizard from process records

To set up a wizard to be called from process records:

1. Click **Tailoring > Document Engine > Processes**.
2. Select a process record.
3. Select the **RAD** tab.
4. In the RAD Application field, enter the RAD Application **wizard.run** to associate the wizard with this record.
5. In the Condition field, enter the condition associated with the RAD Application field. For example, "true".
6. In the Parameter Names field, enter the parameter names to pass to the RAD application.
7. In the Values field, enter the parameter values to pass to the RAD application.
8. Click **Save**.
9. Click **OK**.

Example:

In this example, a wizard is set up to be called from the add.device Process record. The add.device Process record is opened to evaluate the RAD call.

To set up a wizard to be called from process records:

1. Click **Tailoring > Document Engine > Processes**.
2. Open the **add.device** Process record.
3. Select the **RAD** tab.
4. In the RAD Application field, enter the RAD Application **wizard.run** to associate the wizard with this record.
5. The Condition field contains the name of the parameters to be used (file and name).
6. In the Parameter Names field, enter the parameter names to pass to the RAD application.
7. The Values field specifies the values of the parameters being passed to the RAD application. In this example, the following values are specified for each parameter named.

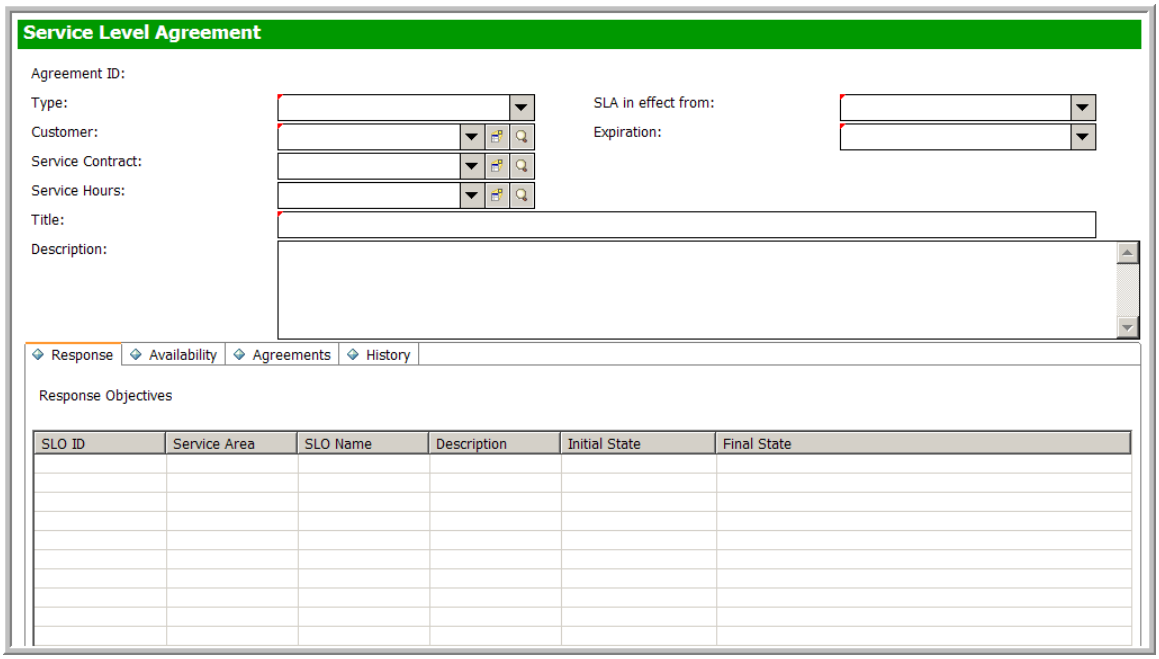
Name of parameter	Value of the parameter
file	$L.file
name	"Add Device"

8. Click **Save**.
9. Click **OK**.

4 Example: Creating a wizard that creates a new SLA

In this example, we will create a wizard that creates a new Service Level Agreement record by prompting the user to enter the necessary data for a new SLA and then saving that data to the database.

Out of box, a Service Manager user creates a new SLA by clicking **Service Level Management > Service Level Agreements > Create New SLA**. The user then fills in the required fields and any relevant optional fields in the Service Level Agreement entry form:

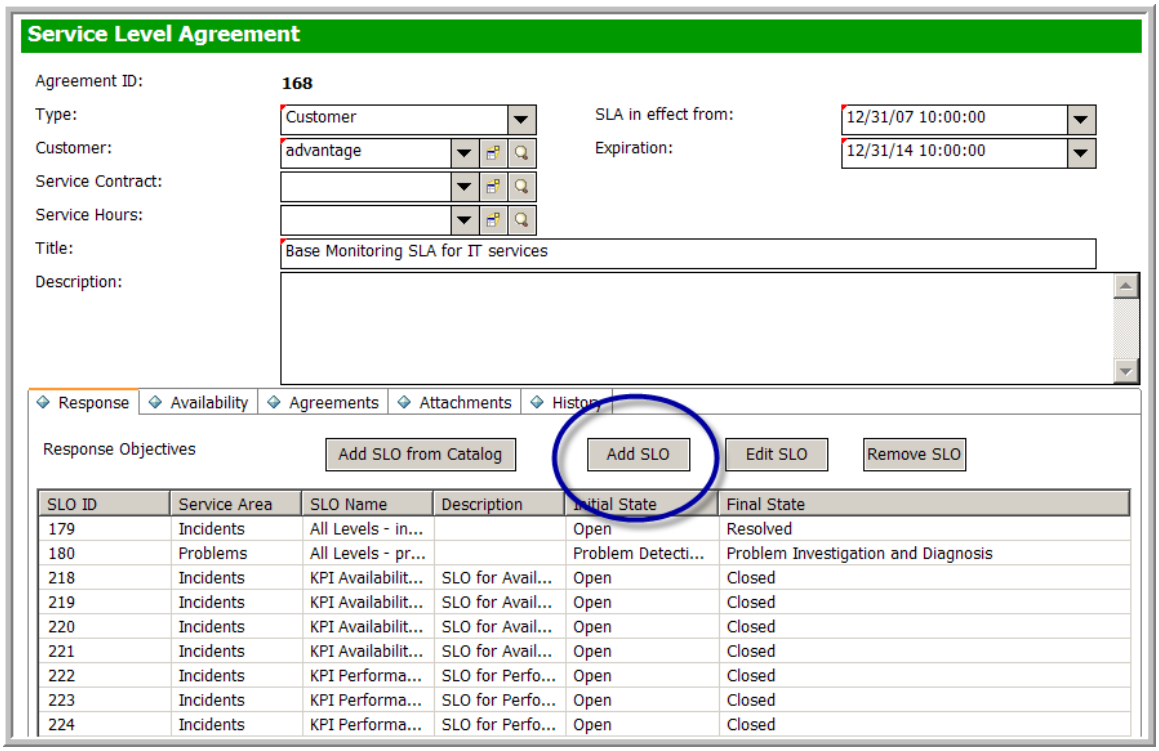


Our intent is to present users with a wizard to guide them through supplying the necessary SLA record information by presenting a series of wizard panels. When all the information has been supplied by the user, the wizard process saves the Service Level Agreement record to the database.

Adding an SLO to the SLA

Service Level Agreements generally have Service Level Objectives associated with them. Service Manager includes two out-of-box wizards for adding a Response SLO (Service Level Objective) and adding an Availability SLO.

After creating an SLA by filling out the Service Level Agreement form, the user can then go to the SLA record and click the Add SLO buttons in the Response and Availability tabs to launch the out-of-box Add Response and Add Availability SLO wizards. This button to launch the out-of-box wizard to add a response SLO is shown in the following screen.



In the example that we are building here, we want our new Create SLA wizard to call the out-of-box Add SLO wizards so that the user can go seamlessly from creating a new SLA to adding any SLOs related to it.

Designing the SLA wizard

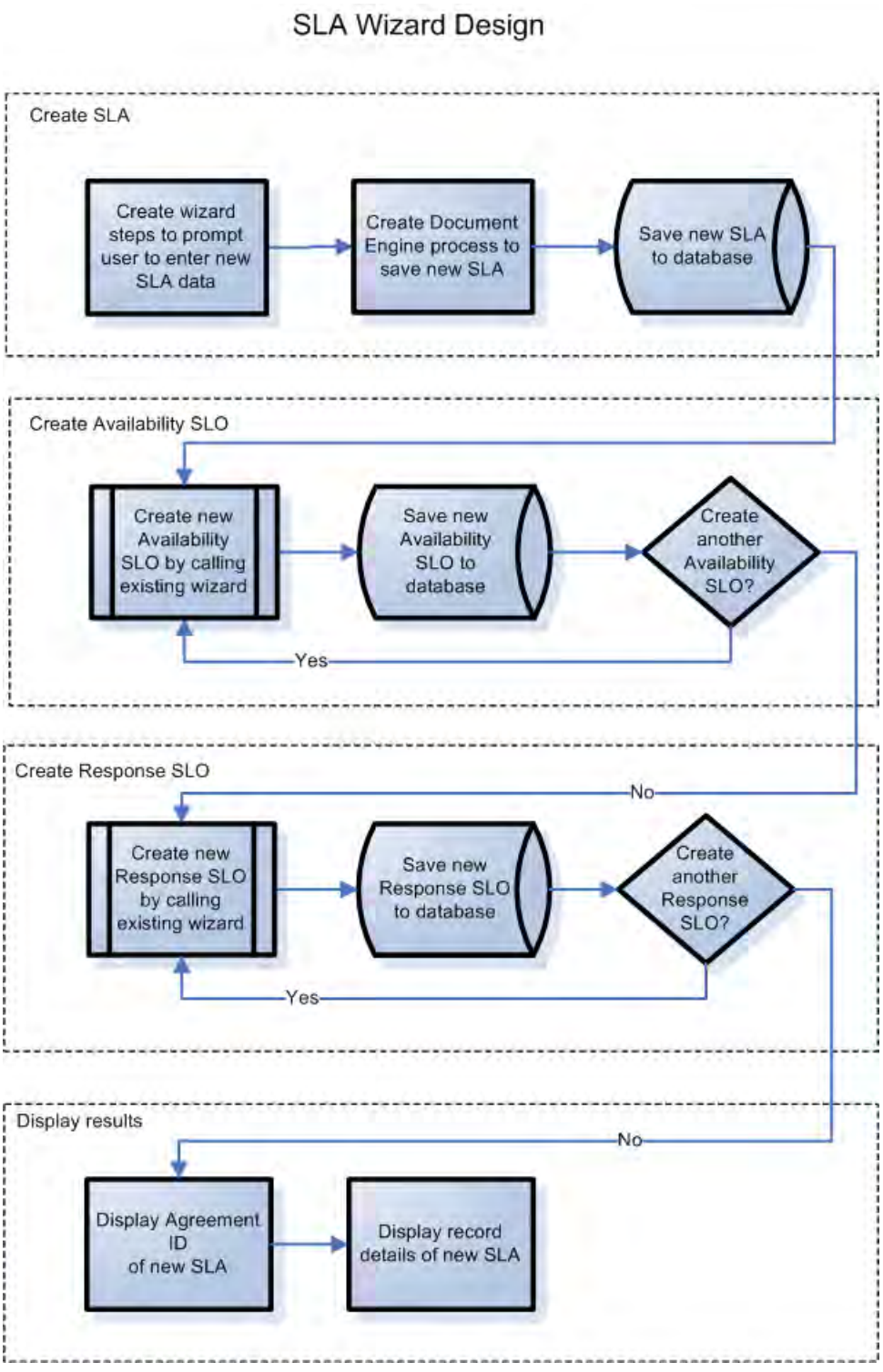
It is helpful to design a new wizard by considering the overall process flow that must occur, including both what the end user needs to do, and any background processes that need to take place.

In this case, we will need to create a series of consecutive wizard steps or panels to display to users to prompt them to enter the basic information needed for the new SLA record. When the required information has been gathered, the new SLA record must be saved by a Document Engine process to the database.

The wizard must then ask users if they want to add any availability or response SLOs. If they do, it will call the out-of-box SLO wizards, first for availability SLOs, and then for response SLOs.

Finally the wizard should display the record number and title of the newly-created SLA, and then provide a button for the user to view the record details.

The overall design of the new Create SLA wizard is shown in the following flowchart.


SLA Wizard Design
Create SLA
Create wizard steps to prompt user to enter new SLA data
Create Document Engine process to save new SLA
Save new SLA to database
Create Availability SLO
Create new Availability SLO by calling existing wizard
Save new Availability SLO to database
Create another Availability SLO?
Yes
Create Response SLO
No
Create new Response SLO by calling existing wizard
Save new Response SLO to database
Create another Response SLO?
Yes
Display results
No
Display Agreement ID of new SLA
Display record details of new SLA

Planning the SLA wizard panels

In addition to the high-level wizard design, it is helpful to plan the function of each of the wizard panels.

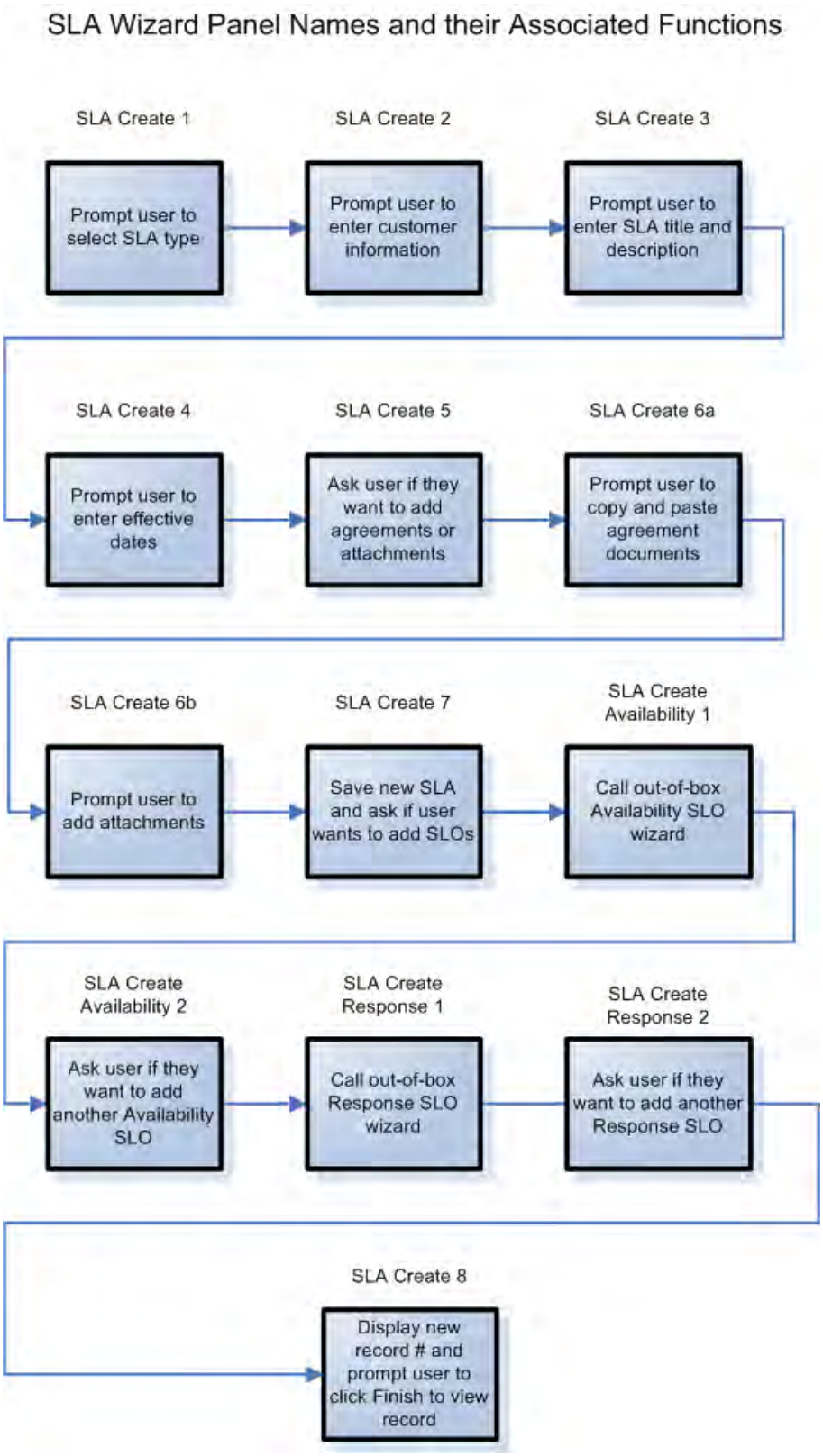
We will create a series of wizard panels that will prompt the user to enter the required fields for an SLA record:

- SLA type (service or customer)
- Customer
- Title and description
- SLA effective dates

In addition, although it is not required, many users will want to copy and paste from the Agreement documents into the SLA record and to add various attachments, so our wizard will present panels to prompt the user for those documents.

We will name each of our wizard panels consecutively (SLA Create 1, SLA Create 2, etc.) so that it is clear that these individual wizard panels connect together to form the Create SLA wizard that will be displayed to the end user.

The flowchart below shows the planned wizard panels and the function that each will perform. Each of the boxes in the flowchart also corresponds to the steps required to build the SLA wizard, described in the Building the SLA wizard section.


SLA Wizard Panel Names and their Associated Functions
SLA Create 1
Prompt user to select SLA type
SLA Create 2
Prompt user to enter customer information
SLA Create 3
Prompt user to enter SLA title and description
SLA Create 4
Prompt user to enter effective dates
SLA Create 5
Ask user if they want to add agreements or attachments
SLA Create 6a
Prompt user to copy and paste agreement documents
SLA Create 6b
Prompt user to add attachments
SLA Create 7
Save new SLA and ask if user wants to add SLOs
SLA Create Availability 1
Call out-of-box Availability SLO wizard
SLA Create Availability 2
Ask user if they want to add another Availability SLO
SLA Create Response 1
Call out-of-box Response SLO wizard
SLA Create Response 2
Ask user if they want to add another Response SLO
SLA Create 8
Display new record # and prompt user to click Finish to view record

Building the SLA wizard

We will use the Wizard tool to design each panel of the Create SLA wizard.

Most of the panels involve prompting the user to provide information, so they also require us to use the Forms Designer to design the form controls for the subformat that is displayed within the wizard panel.

Each of the overview steps listed below corresponds to a box on the flowchart in the previous section. These steps, along with the specific tasks that need to be done using the Wizard Tool and Forms Designer, are listed below.

Step 1: Prompt the user to select SLA type

Step 2: Prompt for customer information

Step 3: Prompt for SLA title and description

Step 4: Prompt for SLA effective dates

Step 5: Ask about attachments and agreements

Step 6: Prompt user for agreements

Step 7: Prompt user to attach other documents

Step 8: Save SLA and ask for SLOs

Step 9: Call Availability SLO wizard

Step 10: Prompt for additional Availability SLOs

Step 11: Call Response SLO wizard

Step 12: Prompt for additional Response SLOs

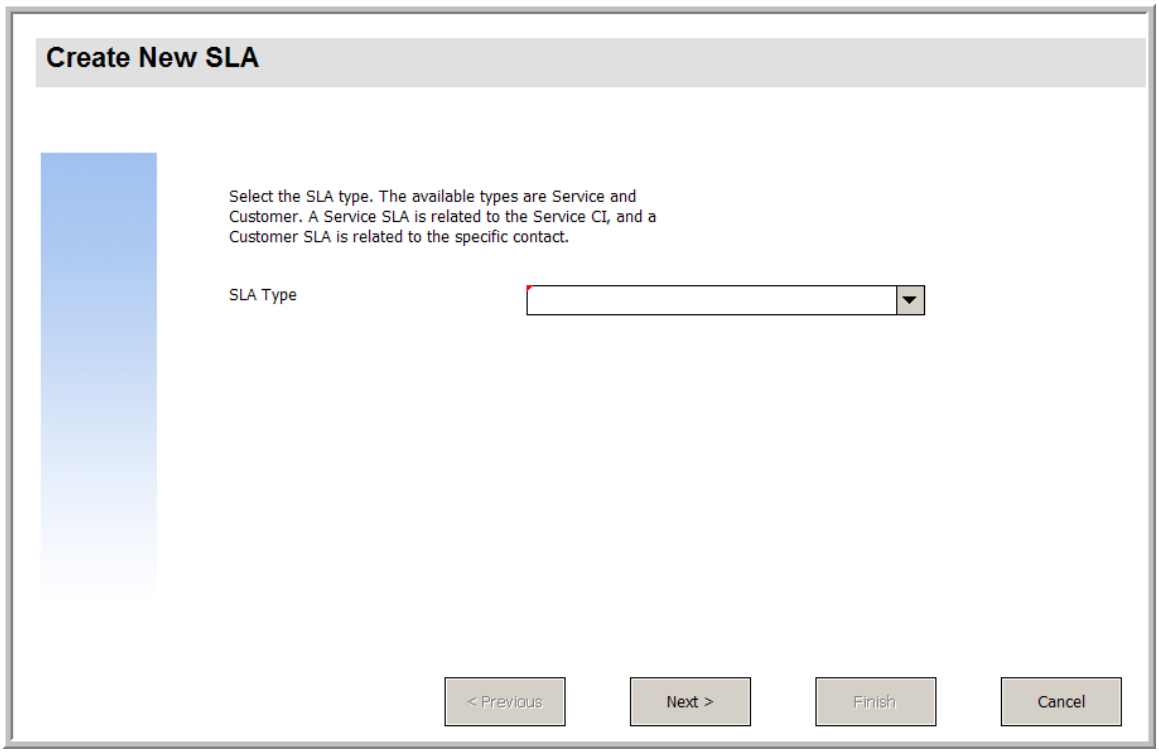
Step 13: Display the new record created

Step 14: Set menu to call the new wizard

Step 1: Prompt the user to select SLA type

The first step is to create the first wizard panel the user will see when executing the Create New SLA wizard from the Service Manager menu navigation area. This panel will prompt the user to select the SLA type, Service or Customer.

We will use the Wizard Tool to define this wizard record, and Forms Designer to design the subformat that is displayed in the wizard panel. This first panel will look like this to the user:



Perform the following two tasks to design the first wizard panel:

Create a wizard for SLA type

Create a form for SLA type

Create a wizard for SLA type

This first task creates the first wizard panel the user sees when executing the Create New SLA wizard from the Service Manager menu. We will use the Wizard Tool to define this first wizard panel.

To create a wizard to prompt for SLA type:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 1**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Select this checkbox to set to **true**, as this wizard is the first in a series of wizard panels.
Brief Description	Type: **This wizard will create an SLA record and guide the user through creating associated SLOs.**
Window Title	Type: **Create New SLA**
Title	Type: **Create New SLA**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following fields:

Field	Value
Create a record	Select this option to indicate that the wizard will create a record. You must enter or select the record type in the record "of type" field. This becomes $L.file.
of type (for Create a record)	Click **Fill** to select the type of record to be created, and select **sla**.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.type**

7. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLACreate 2**

Field	Value
Condition	The condition determines what happens next. In this case, you can enter the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizards, based on user actions or selections. Type: **true**

8. When you are finished, click **Save**.

You have created your first wizard panel. Next, you will use Forms Designer to create the subformat that will display in this wizard panel.

Create a form for SLA type

For this task, you will use Forms Designer to create the form controls that prompt the user to select the SLA type. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 1: Prompt the user to select SLA type.

To create a new form for the SLA type wizard:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.type**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This helps avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Wrap Label	
Caption	Type: **Select the SLA type. The available types are Service and Customer. A Service SLA is related to the Service CI, and a Customer SLA is related to the specific contract.**
Visible	Select.
Label	

Property	Value
Caption	Type: **SLA Type**
Visible	Select.
Combo Box	The Combo Box should be place to the right of the Regular label box and contains a list of choices that are associated with the label.
Display List	Type two entries (each with an initial capital letter): ▪ **Service** ▪ **Customer**
Input	Type: **type**
Mandatory	Select.
Select Only	Select.
Value List	Type two entries (all lowercase letters): ▪ **service** ▪ **customer**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

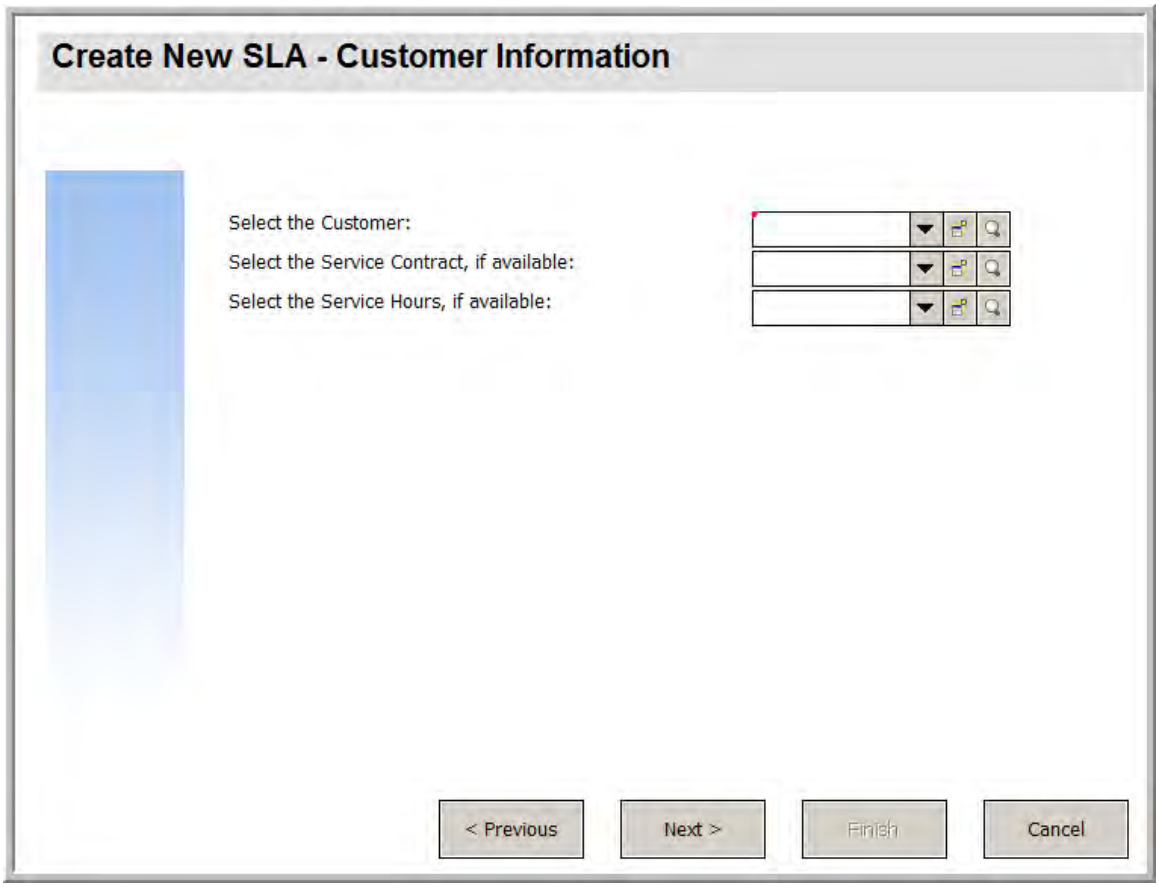
You have completing creating the form for the first panel of the Create New SLA wizard that the user will see.

Step 2: Prompt for customer information

Step 2 is to create the next step of the wizard that the user will see, prompting them to enter customer, service contract, and service hours.

Note: We can use the record we just created, SLA Create 1, as a starting point, and revise the fields to reflect the information below. It is very important to remember to click the **Add** button after you start to make changes so that you do not overwrite the SLA Create 1 record.

The panel we create in this new record will look like this to the user:



Perform the following two tasks to design a wizard panel that gathers customer information:

Create a wizard to gather customer information

Create a form to gather customer information

Create a wizard to gather customer information

For this task, you will create the next wizard panel the users will see, prompting them to enter customer, service contract, and service hours.

To create a wizard to gather customer information:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 2**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than Save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard panel prompts for the customer name, service contract, and service hours.**
Window Title	Type: **Create New SLA - Customer Information**
Title	Type: **Create New SLA - Customer Information**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
$L.file passed in	Select this option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.customer**
Use Conditional Previous Exits?	Select this option to define possible exits when the "Previous" button is clicked.

7. Select the **Actions** tab and the **Format Control/Process Name** subtab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that is being performed. This can either be the current file, the list selected (as a whole) or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLACreate 3**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizards, based on user actions or selections. Type: **true**

9. When you are finished, click **Save**.

You have created the customer information wizard panel. Next, you will create the new wizard form in Forms Designer.

Create a form to gather customer information

For this task, you will use Forms Designer to create controls that prompt the user to supply basic customer information. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 2: Prompt for customer information.

To create a new SLA form to gather customer information:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.customer**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Label	

Property	Value
Caption	Type: **Select the customer:**
Visible	Select.
Comfill	
Combo Button Visible	Select.
Input	customer
Mandatory	Select.
Select Only	Select.
Third Button Visible	Select.
Value List	Type: **$lo.colist**
Visible	Select.
Label	
Caption	Type: **Select the Service Contract, if available:**
Visible	Select.
Comfill	
Combo Button Visible	Select.
Fill Button Visible	Select.
Input	Type: **service.contract**
Select Only	Select.
Third Button Visible	Select.
Value List Condition	Type: **select("contract.id","servicecontract","provider",[customer])**
Visible	Select.

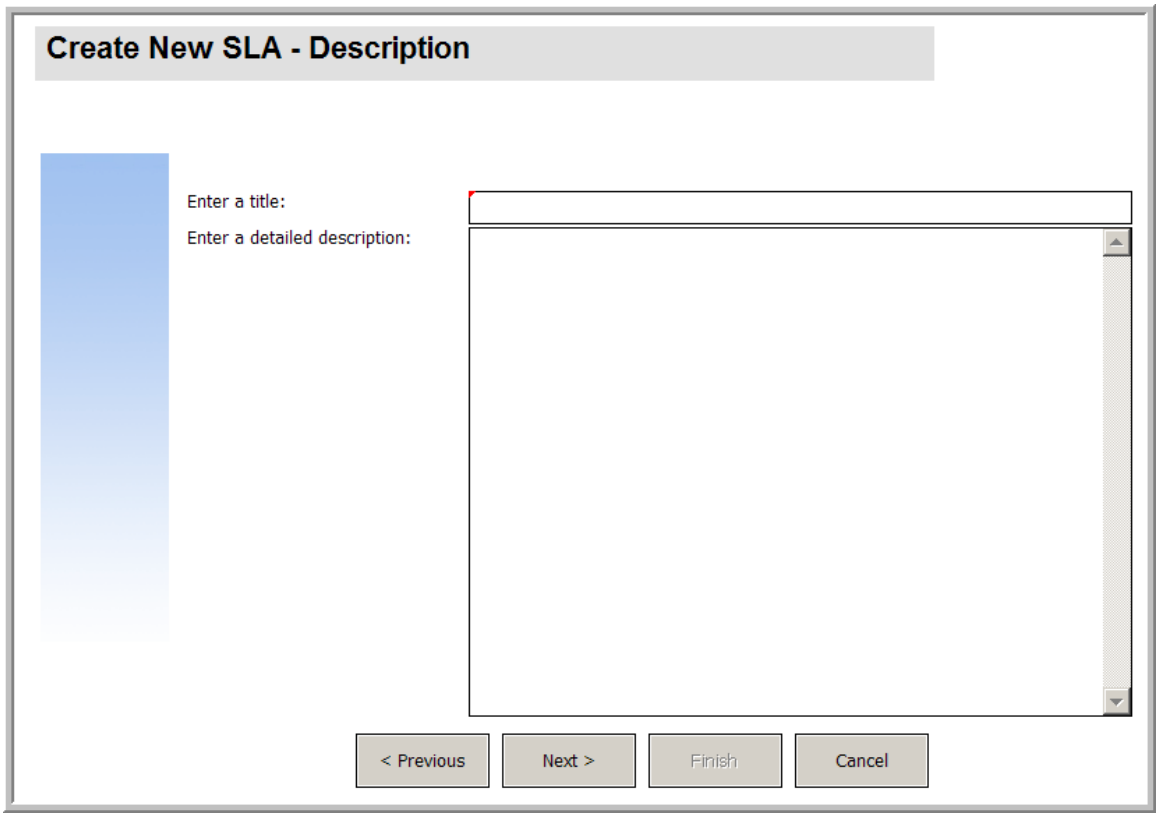
Property	Value
Label	
Caption	Type: **Select the Service Hours, if available:**
Visible	Select.
Comfill	
Combo Button Visible	Select.
Fill Button Visible	Select.
Input	Type: **service.hours**
Select Only	Select.
Third Button Visible	Select.
Value List	Type: **$G.calendars**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have completed creating one of the forms needed for the SLA wizard.

Step 3: Prompt for SLA title and description

Step 3 is to create the next wizard step, prompting the user to enter a title and description for the SLA. The panel will look like this:



Perform the following two tasks to design a wizard panel for the user to enter an SLA title and description:

Create a wizard for SLA description

Create a form for SLA description

Create a wizard for SLA description

For this task, you will create a wizard panel that prompts the user to enter a title and description for the new SLA.

To create a wizard panel for the SLA title and description:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 3**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard panel will prompt the user to enter a title and description for the SLA.**
Window Title	Type: **Create New SLA - Description**
Title	Type: **Create New SLA - Description**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
$L.file passed in	Select this option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.description**

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that is going to be performed. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLACreate 4**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizards, based on user actions or selections. Type: **true**

9. When you are finished, click **Save**.

You have created the new wizard panel. Next, you will create the new wizard form in Forms Designer.

Create a form for SLA description

For this task, you will use Forms Designer to create controls that prompt the user to enter a title and description. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 3: Prompt for SLA title and description.

To create a new form to display the SLA description wizard:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.description**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Label	

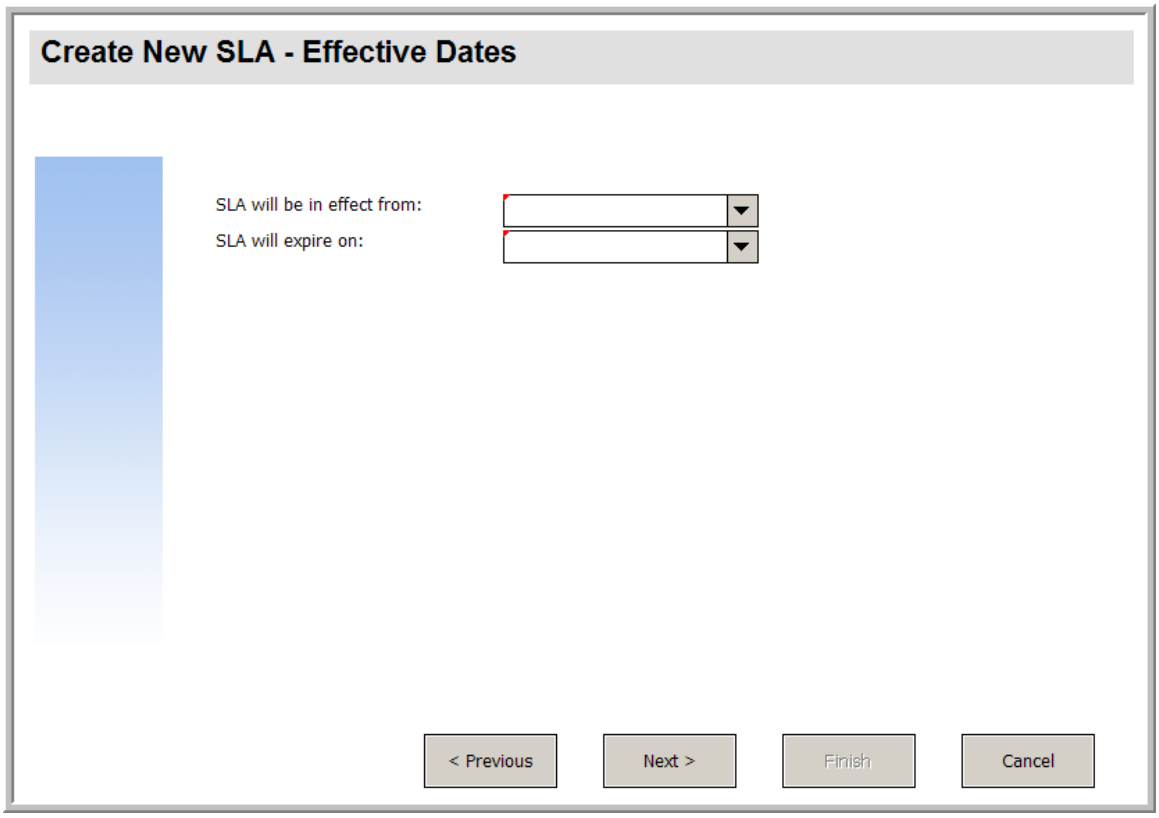
Property	Value
Caption	Type: **Enter a title.**
Visible	Select.
Text	
Input	Type: **title**
Visible	Select.
Label	
Caption	Type: **Enter a detailed description.**
Visible	Select.
Text Area	
Input	Type: **description**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have created one of the forms needed for the SLA wizard.

Step 4: Prompt for effective dates

Step 4 is to create the next wizard step, prompting the user to enter the starting and ending dates the SLA. The panel will look like this:



Perform the following two tasks to design a wizard panel for the user to indicate the effective dates of the SLA:

Create a wizard for SLA effective dates

Create a form for SLA effective dates

Create a wizard for SLA effective dates

For this task, you will create a wizard that prompts the user to enter valid start and end dates.

To create a wizard for valid start and end dates:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 4**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard panel will prompt the user to enter the beginning and ending dates of the SLA.**
Window Title	Type: **Create New SLA - Effective Dates**
Title	Type: **Create New SLA - Effective Dates**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following fields:

Field	Value
$L.file passed in	Select this option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.validfromto**

7. Select the **Actions** tab and the **Format Control/Process Name** subtab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that is going to be performed. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLACreate 5**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizards, based on user actions or selections. Type: **true**

9. When you are finished, click **Save**.

You have created the valid From and To dates wizard panel. Next, you will create the new wizard form in Forms Designer.

Create a form for SLA effective dates

For this task, you will use Forms Designer to create the form controls that prompt the user to indicate the effective dates of the SLA. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 4: Prompt for effective dates.

To create a new form for the SLA effective dates wizard:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.validfromto**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Label	

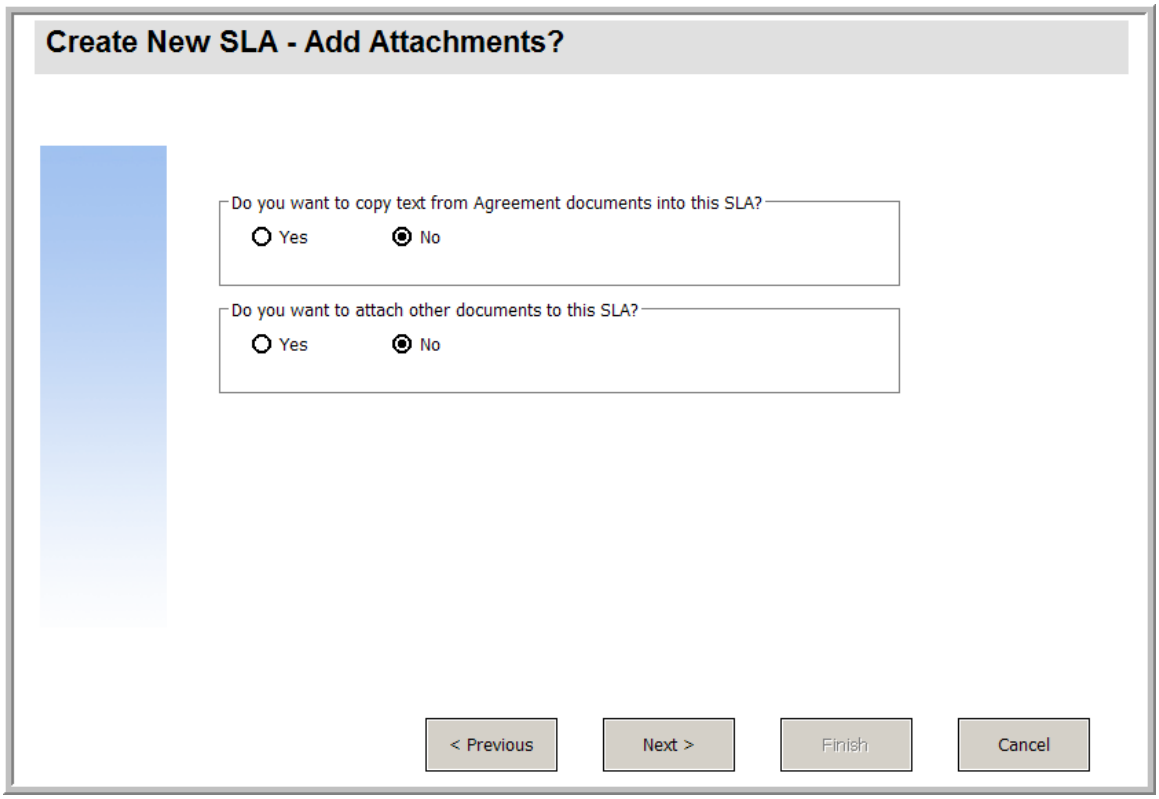
Property	Value
Caption	Type: **SLA will be in effect from:**
Visible	Select.
Comfill	
Fill Button Visible	Select.
Input	start
Mandatory	Select.
Visible	Select.
Label	
Caption	Type: **SLA will expire on:**
Visible	Select.
Comfill	
Fill Button Visible	Select.
Input	Type: **expiration**
Mandatory	Select.
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have finished creating one of the forms needed for the SLA wizard.

Step 5: Ask about agreements and other documents

Step 5 is to create a wizard panel that asks the user whether they want to copy text from an agreement document or attach a document to the SLA record. The panel will look like this:



Perform the following two tasks to design a wizard panel asks users about agreements and other documents related to the SLA:

Create a wizard to ask about attachments

Create a form to ask about attachments

Create a wizard to ask about attachments

For this task, you will create a wizard panel that asks users if they want to copy text or attach documents to this SLA.

To create a wizard to ask users about SLA attachments:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 5**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard will ask the user whether they want to copy text from Agreement documents or attach other documents to the SLA.**
Window Title	Type: **Create New SLA - Add Attachments?**
Title	Type: **Create New SLA - Add Attachments?**

5. Select the **File Selection** tab to complete the following fields:

Field/Subtab	Value
Initial Expressions subtab	Type: **$G.agreement.docs=false** **$G.other.attach=false**
Select $L.file by subtab	Select the **$L.file passed in** option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.ask.attachments**

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 6a**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **$G.agreement.docs=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 6b**
Condition	Type: **$G.other.attach=true**
Wizard Name	This is the name ofthe next wizard in a series of wizards. Type: **SLA Create 7**
Condition	Type: **true**

9. When you are finished, click **Save**.

You have created the wizard panel that asks users about attachments. Next, you will create the new wizard form in Forms Designer.

Create a form to ask about attachments

For this task, you will use Forms Designer to create the form controls that ask users to whether they want top copy text or add attachments related to the SLA. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 5: Ask about agreements and other documents.

To create a new form for the SLA wizard that asks about attachments:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.ask.attachments**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Group	
Caption	Type: **Do you want to copy text from Agreement documents into this SLA?**
Visible	Select.
Radio Button	
Caption	Type: **Yes**
Input	Type: **$G.agreement.docs**
Value	Type: **true**
Visible	Select.
Radio Button	
Caption	Type: **No**
Input	Type: **$G.agreement.docs**
Value	Type: **false**

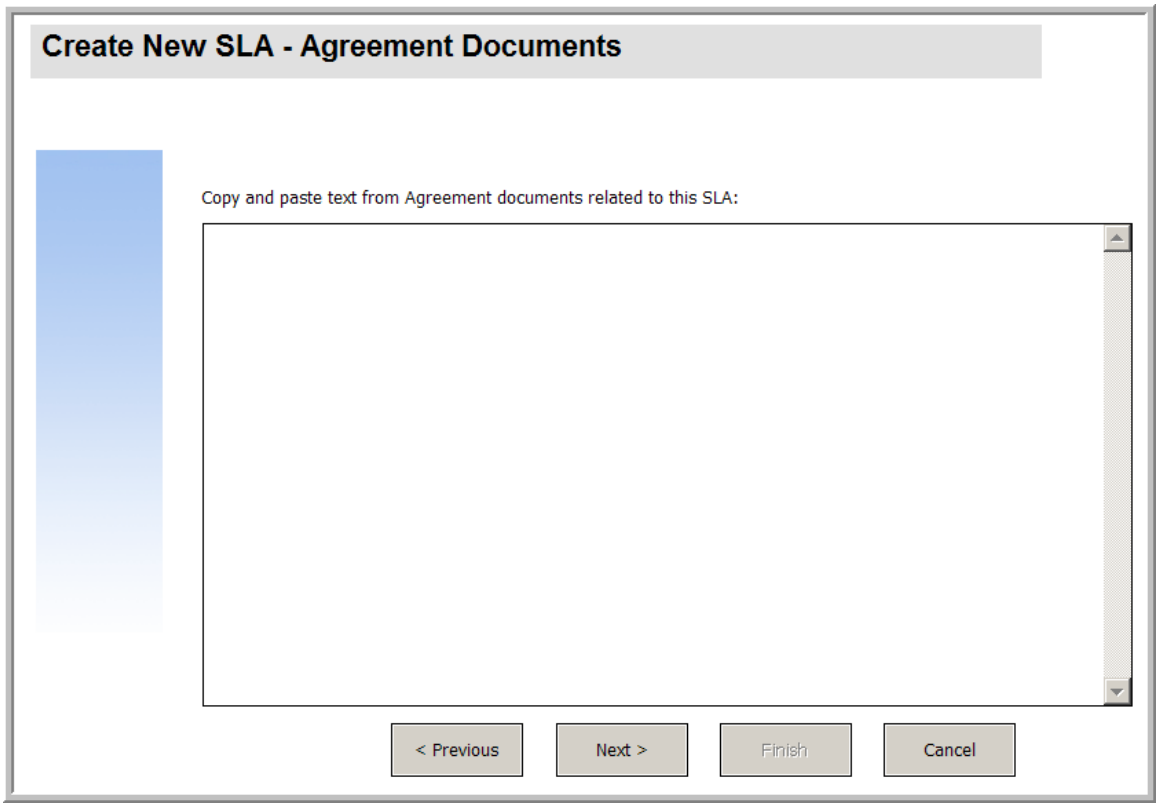
Property	Value
Visible	Select.
Group	
Caption	Type: **Do you want to attach other documents to this SLA?**
Visible	Select.
Radio Button	
Caption	Type: **Yes**
Input	Type: **$G.other.attach**
Value	Type: **true**
Visible	Select.
Radio Button	
Caption	Type: **No**
Input	Type: **$G.other.attach**
Value	Type: **false**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have finished creating one of the forms needed for the SLA wizard.

Step 6: Prompt to add agreements

Step 6 is to create the wizard panel that prompts the user to add agreements related to the SLA if they answered Yes on the previous panel. The user does this by copying and pasting text into the entry field. The panel will look like this:



Perform the following two tasks to design a wizard panel to prompt the user to add agreements:

Create a wizard to add agreements

Create a form to add agreements

Create a wizard to add agreements

For this task, you will create a wizard panel that prompts users to copy and paste text from agreements related to the SLA, if they answered Yes in the previous panel.

To create a wizard panel that prompts users to attach agreement documents:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 6a**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than Save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard will prompt the user to copy and paste text from Agreement documents.**
Window Title	Type: **Create New SLA - Agreement Documents**
Title	Type: **Create New SLA - Agreement Documents**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
$L.file passed in	Select this option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.add.agreements**

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 6b**
Condition	Type: **$G.other.attach=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 7**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizards, based on user actions or selections. Type: **true**

9. When you are finished, click **Save**.

You have created the wizard panel that prompts users to attach agreement documents. Next, you will create the new wizard form in Forms Designer.

Create a form to add agreements

For this task, you will use Forms Designer to create the form controls that prompt the user to copy and paste text from agreement documents related to the SLA. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 6: Prompt to add agreements.

To create a new form to display the wizard that prompts users to add agreement document text:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.add.agreements**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

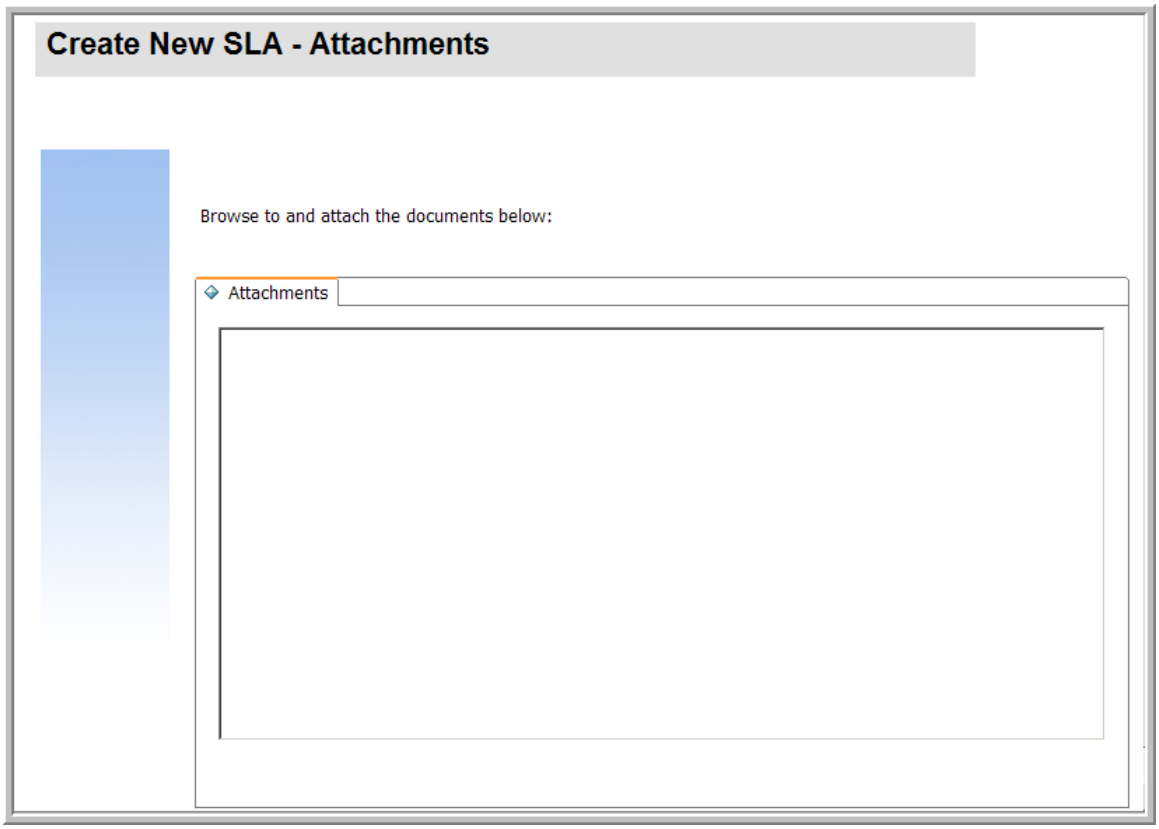
Property	Value
Label	
Caption	Type: **Copy and paste text from Agreement documents related to this SLA:**
Visible	Select.
Text Area	
Input	Type: **agreements**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have finished creating one of the forms needed for the SLA wizard.

Step 7: Prompt to attach other documents

Step 7 is to create the wizard panel that prompts the user to attach documents related to the SLA if they answered Yes on the panel asking if they wanted to do so. The panel will look like this:



Perform the following two tasks to design a wizard panel for the user to attach documents related to the SLA:

Create a wizard to attach other documents

Create a form to attach other documents

Create a wizard to attach other documents

For this task, you will create a wizard panel that prompts users to attach other documents, if they answered Yes to the previous panel.

To create a wizard panel that prompts users to add other documents:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 6b**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than Save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard will prompt the user to add any attachments they have.**
Window Title	Type: **Create New SLA - Attachments**
Title	Type: **Create New SLA - Attachments**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
$L.file passed in	Select this option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.add.otherdocs**

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 7**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you canspecify different wizazrds, based on user actions or selections. Type: **true**

9. When you are finished, click **Save**.

You have created the wizard panel that prompts users to add other documents. Next, you will create the new wizard form in Forms Designer.

Create a form to attach other documents

For this task, you will use Forms Designer to create the form controls that prompt the user to attach documents related to the SLA. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the screen shown in Step 7: Prompt to attach other documents.

To create a new form for the SLA wizard that prompts users to add other documents:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.add.otherdocs**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Click **Design**.

   Note: When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).
6. Create the following for this form.

Property	Value
Label	

Property	Value
Caption	Type: **Browse to and attach the documents below:**
Visible	Select.
Notebook Tab	
Caption	Type: **Attachments**
Visible	Select.
Visible Condition	Type: **[$L.mode]<>"add"**
Attachments	
Visible	Select.
Visible condition	supports("ole")

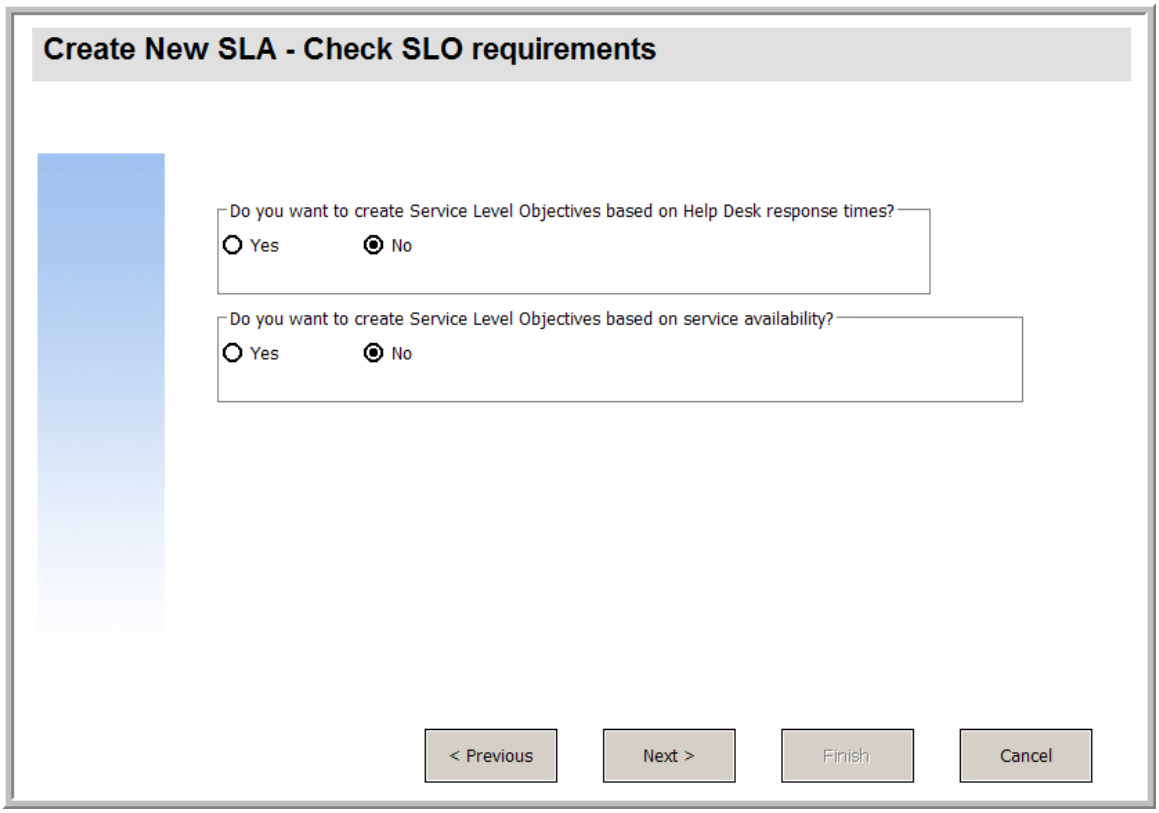
7. Click **OK** twice to save the new form and exit.

You have finished creating one of the forms needed for the SLA wizard.

Step 8: Save new SLA and prompt for SLOs

At this point, we have gathered all of the information we need for the new SLA. The first task we need to do is create a process in Document Engine to save the new SLA to the database.

Then we can continue with our wizard panels. The wizard panel for this step will call the new save process we created, and ask the user if they want to add SLOs to the SLA. The panel the user will see will look like this:



Perform the following three tasks to complete this step:

Create a process to save the new SLA

Create a wizard to check SLO requirements

Create a form to check SLO requirements

Create a process to save the new SLA

For this task, you will create a process to save the new SLA record that you just created.

To create a process to save the new SLA record:

1. Click **Tailoring > Document Engine > Processes**. The Process Definition form opens.
2. In the Process Name field, type **sla.save.from.wizard**.
3. In the RAD tab, type the following expressions in the first two rows of the **Expressions evaluated before RAD call** section:

```
$L.base.vars={$L.object.name, $L.save.mode}
```

$L.action="add"

4. In the **RAD Application** section of the RAD tab, type**se.base.method** as the RAD application to be called, and **true** as the condition to execute it. Then enter the following parameter names and values:

Parameter Name	Parameter Value
file	Type: **$L.file**
prompt	Type: **$L.action**
second.file	Type: **$L.file.save**
record	Type: **$L.fc**
second.record	Type: **$L.object**
text	Type: **$L.exit**
query	Type: **$L.mode**
names	Type: **$L.base.vars**

5. In the **Post RAD Expressions** section of the RAD tab, type: **$G.new.sla=agreement.id in $L.file**.
6. Click **Add**.
7. When you are finished, click **OK**.

You have created the process to save the new SLA record to the database.

Create a wizard to check SLO requirements

For this task, you will create a wizard to call the save process, and then create a subformat that asks users whether they want to add SLOs and check the SLO requirements.

To create a wizard to call the save process and check SLO requirements:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 7**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.
4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard panel will ask the user if they want to create Availability or Response SLOs.**
Window Title	Type: **Create New SLA - Check SLO Requirements**
Title	Type: **Create New SLA - Check SLO Requirements**

5. Select the **File Selection** tab to complete the following fields:

Field/Subtab	Value
Initial Expressions subtab	Type: **cleanup($G.agreement.docs)** **cleanup($G.other.attach)** **$G.create.response.slo=false** **$G.create.avail.slo=false**
Select $L.file by subtab	Select the **$L.file passed in** option to indicate that the $L.file variable should be passed to the wizard from a previous wizard.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.

Field	Value
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.ask.slos**

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.
Format Control/Process Name subtab	Define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:
Process Name	To specify that this wizard can run a process after completing, type: **sla.save.from.wizard**

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Response 1**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **$G.create.response.slo=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Availability 1**
Condition	Type: **$G.create.avail.slo=true**
Wizard Name	Type: **SLA Create 8**

Field	Value
Condition	Type: **true**

9. When you are finished, click **Save**.

You have created the wizard panel that calls the save process and checks SLO requirements. Next, you will create the new wizard form in Forms Designer.

Create a form to check SLO requirements

For this task, you will use Forms Designer to create form controls that ask users if they want to create Service Level Objectives related to the SLA. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the wizard screen shown in Step 8: Save new SLA and prompt for SLOs.

To create a new form for the wizard that calls the save process and checks SLO requirements:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.ask.slos**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This helps avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Group	
Caption	Type: **Do you want to create Service Level Objectives based on Help Desk response times?**
Visible	Select.
Radio Button	

Property	Value
Caption	Type: **Yes**
Input	Type: **$G.create.response.slo**
Value	Type: **true**
Visible	Select.
Radio Button	
Caption	Type: **No**
Input	Type: **$G.create.response.slo**
Value	Type: **false**
Visible	Select.
Group	
Caption	Type: **Do you want to create Service Level Objectives based on service availability?**
Visible	Select.
Radio Button	
Caption	Type: **Yes**
Input	Type: **$G.create.avail.slo**
Value	Type: **true**
Visible	Select.

Property	Value
Radio Button	
Caption	Type: **No**
Input	Type: **$G.create.avail.slo**
Value	Type: **false**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have finished creating one of the forms needed for the SLA wizard.

Step 9: Call the Availability SLO wizard

The next step is to create a wizard panel to call the existing out-of-box Availability Wizard for those users who answered that they did want an availability SLO.

Because this wizard panel only performs the background task of calling another wizard, there is no screen to show the end user. Thus there is no task requiring the use of Forms Designer in this step.

Create a wizard to call an availability SLO

Create a wizard to call an Availability SLO

For this task, you will create a wizard that will call the existing out-of-box Availability SLO wizard.

To create a wizard that calls the existing out-of-box Availability SLO wizard:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create Availability 1**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than Save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard calls the existing wizard to create an Availability SLO.**
Window Title	Type: **Create Availability SLO**
Title	Type: **Create Availability SLO**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
No $L.file (use type-check)	Select this option to initialize a typecheck file, which acts as a holding file.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Skip Display	Select this option as this panel does not include a form that needs to be displayed to the user.
Use Conditional Previous Exits?	Select this option to define possible exits when the "Previous" button is clicked.

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

Field	Value
Format Control/Process Name subtab	Define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:
Process Name	To specify that this wizard can run a process after completing, type: **slo.add.avail.sla**

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Availability 2**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **true**

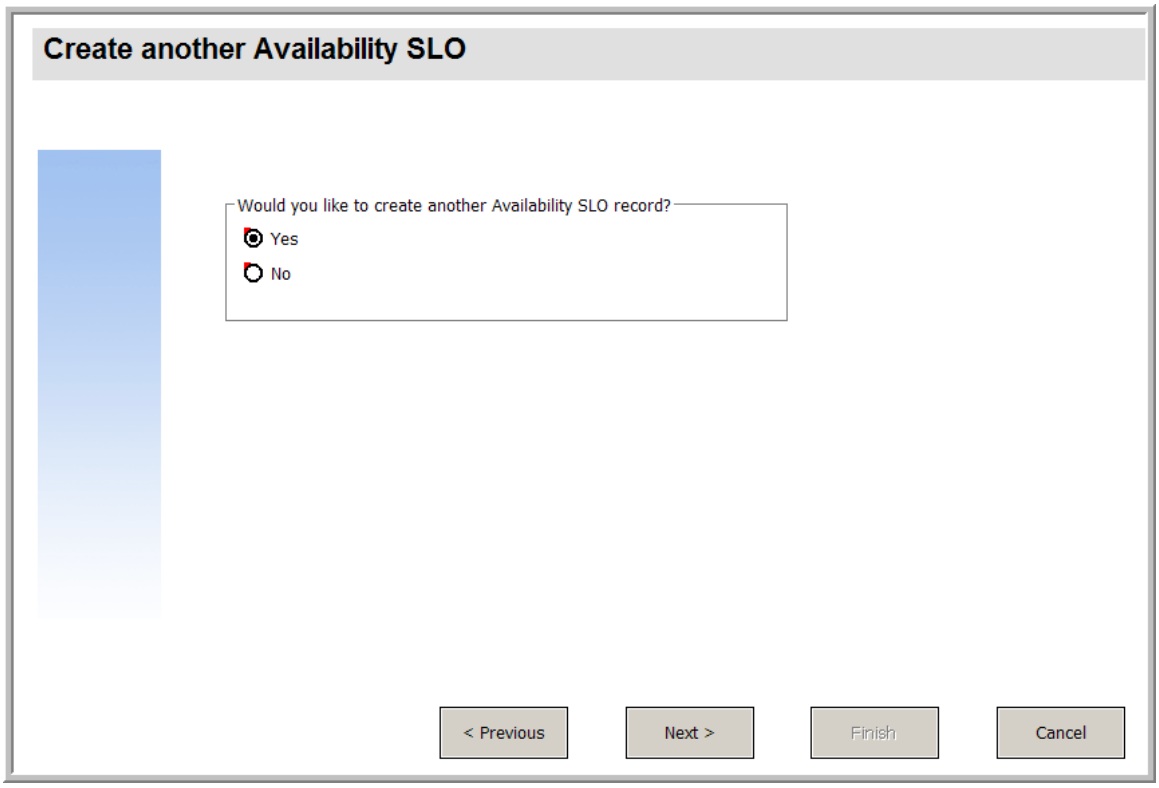
9. When you are finished, click **Save**.

You have created a wizard panel that calls the existing out-of-box Availability SLO. A new form is not needed for this wizard.

Step 10: Prompt for another Availability SLO

The next step is to ask the user if they want to add another availability SLO. If they do, this panel must call the out-of-box Availability SLO wizard again. If they do not, the wizard must call the out-of-box Response SLO if the user indicated they wanted to add a Response SLO, or else go to the final wizard screen.

The wizard panel will look like this:



Perform the following two tasks to design a wizard panel to prompt for another Availability SLO:

Create a wizard for more availability SLOs

Create a form for more availability SLOs

Create a wizard for more Availability SLOs

For this task, you will create a wizard so users can create more Availability SLOs.

To create a wizard so users can create more Availability SLOs:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create Availability 2**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard panel asks the user if they want to create another Availability SLO.**
Window Title	Type: **Create another Availability SLO**
Title	Type: **Create another Availability SLO**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
No $L.file (use typecheck)	Select this option to initialize a typecheck file, which acts as a holding file.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.ask.other.avail**
Use Conditional Previous Exits?	Select this option to define possible exits when the "Previous" button is clicked.

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Availability 1**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **$G.create.avail.slo=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 8**
Condition	**true**

9. When you are finished, click **Save**.

You have created the wizard so users can create more Availability SLOs. Next, you will create the new wizard form in Forms Designer.

Create a form for more Availability SLOs

For this task, you will use Forms Designer to create form controls that ask users if they want to create an additional Service Level Objective record. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the wizard screen shown in Step 10: Prompt for another availability SLO.

To create a new form for the wizard that creates more Availability SLOs:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.ask.other.avail**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Group	

Property	Value
Caption	Type: **Would you like to create another Availability SLO record?**
Visible	Select.
Radio Button	
Caption	Type: **Yes**
Input	Type: **$G.create.avail.slo**
Mandatory	Select.
Value	Type: **true**
Visible	Select.
Radio Button	
Caption	Type: **No**
Input	Type: **$G.create.avail.slo**
Mandatory	Select.
Value	Type: **false**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have finished creating one of the forms needed for the SLA wizard.

Step 11: Call the Response SLO wizard

The next step is to create a wizard panel to call the existing out-of-box Response Wizard for those users who answered that they did want to add a Response SLO.

Because this wizard panel only performs the background task of calling another wizard, there is no screen to show the end user. Thus there is no task requiring the use of Forms Designer in this step.

Create a wizard to call an response SLO

Create a wizard to call a Response SLO

For this task, you will create a wizard that will call the existing out-of-box Response SLO wizard..

To create a wizard to call the existing out-of-box Response SLO wizard:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create Response 1**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than save. If you click Save, you will replace the existing record with the new record you are attempting to add.
4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard will call the out-of-box wizard that creates a Response SLO.**
Window Title	Type: **Create Response SLO**
Title	Type: **Create Response SLO**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
No $L.file (use typecheck)	Select this option to initialize a typecheck file, which acts as a holding file.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Skip Display	Select this option as this panel does not include a form that needs to be displayed to the user.

Field	Value
Use Conditional Previous Exits?	Select this option to define possible exits when the "Previous" button is clicked.

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.
Format Control/Process Name subtab	Define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:
Process Name	To specify that this wizard can run a process after completing, type: **slo.add.resp.sla**

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Response 2**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **true**

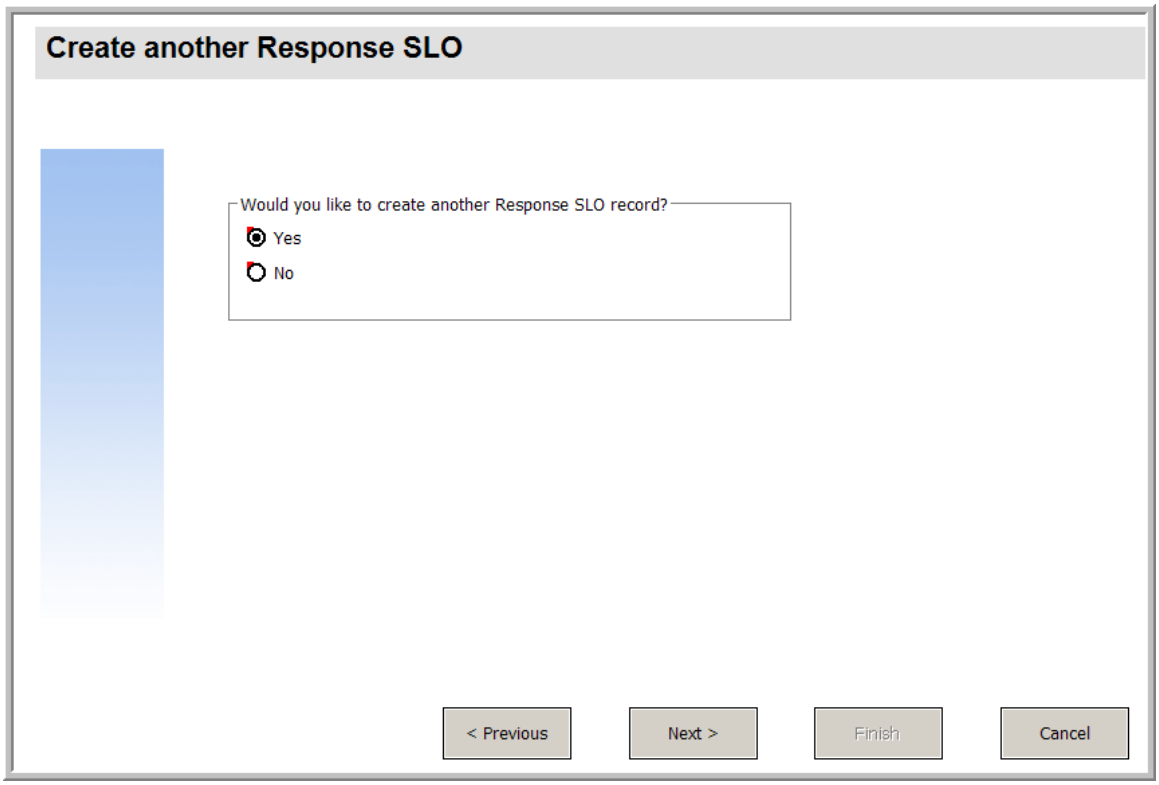
9. When you are finished, click **Save**.

You have created the wizard that will call the existing out-of-box Response SLO wizard. A new form is not needed for this wizard.

Step 12: Prompt for another Response SLO

The next step is to ask the user if they want to add another response SLO. If they do, this panel must call the out-of-box Reponse SLO wizard again. If they do not want to add any more response SLOs, they should be shown the final wizard screen.

The wizard panel will look like this:



Perform the following two tasks to design a wizard panel to prompt for another Response SLO:

Create a wizard for more response SLOs

Create a form for more response SLOs

Create a wizard for more Response SLOs

For this task, you will create a wizard so that users can create more Response SLOs.

To create a wizard so that users can create more Response SLOs:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create Response 2**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than Save. If you click Save, you will replace the existing record with the new record you are attempting to add.

4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard panel will ask the user if they want to add another Response SLO.**
Window Title	Type: **Create another Response SLO**
Title	Type: **Create another Response SLO**

5. Select the **File Selection** tab and the **Select $L.file by** subtab to complete the following field:

Field	Value
No $L.file (use typecheck)	Select this option to initialize a typecheck file, which acts as a holding file.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.ask.other.response**
Use Conditional Previous Exits?	Select this option to define possible exits when the "Previous" button is clicked.

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Response 1**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **$G.create.response.slo=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Availability 1**
Condition	Type: **$G.create.avail.slo=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create 8**
Condition	Type: **true**

9. When you are finished, click **Save**.

You have created the wizard so that users can create more Response SLOs. Next, you will create the new wizard form in Forms Designer.

Create a form for more Response SLOs

For this task, you will use Forms Designer to create form controls that ask users if they want to create an additional Service Level Objective record. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the wizard screen shown in Step 12: Prompt for another response SLO.

To create a new form for a wizard that creates more Response SLOs:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.ask.other.response**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This will help you to avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Group	
Caption	Type: **Would you like to create another Response SLO?**
Visible	Select.
Radio Button	
Caption	Type: **Yes**
Input	Type: **$G.create.response.slo**
Mandatory	Select.
Value	Type: **true**
Visible	Select.
Radio Button	
Caption	Type: **No**
Input	Type: **$G.create.response.slo**
Mandatory	Select.

Property	Value
Value	Type: **false**
Visible	Select.

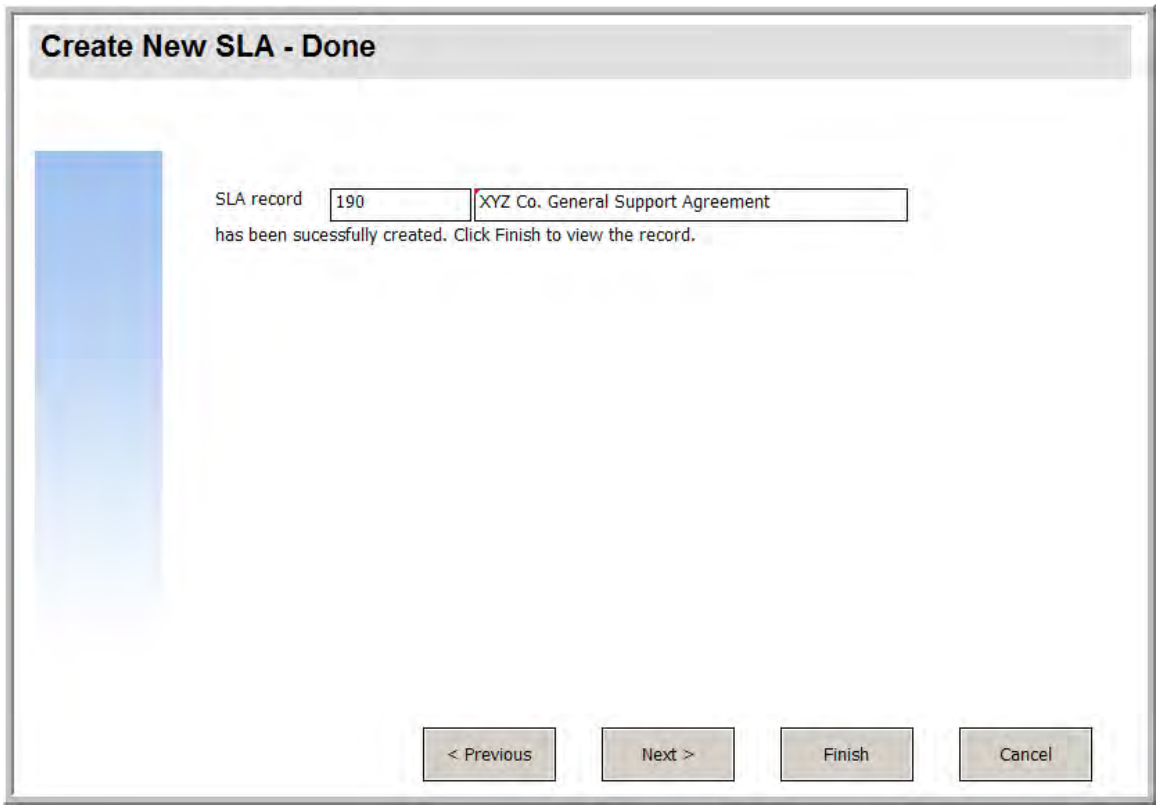
6. Click **OK** twice to save the new form and exit.

You have created one of the forms needed for the SLA wizard.

Step 13: Display new SLA number and title

In this final wizard step, we will display the record number and title of the newly created SLA, and have the record details displayed when the user clicks Finish button.

The panel will look like this:



Complete the following tasks to perform this step:

Create a wizard for the completed SLA

Create a form for the completed SLA

Create a wizard for the completed SLA

For this task, you will create a wizard panel that displays the SLA record number and title of the newly-created SLA record to the user.

To create a wizard for the completed SLA:

1. Click **Tailoring > Wizards**. The Wizard information form opens.
2. Select the **Wizard Info** tab.
3. In the Wizard Name field, type **SLA Create 8**, and then click **Add**.

   Caution: If you use an existing SLA record to create a new SLA wizard panel, click "Add" rather than Save. If you click Save, you will replace the existing record with the new record you are attempting to add.
4. In the **Wizard Info** tab, complete the following fields.

Field	Value
Start Node?	Leave this option unchecked.
Brief Description	Type: **This wizard is the final panel the user sees, telling them the record number and name of the new SLA created.**
Window Title	Type: **Create new SLA - Done**
Title	Type: **Create new SLA - Done**

5. Select the **File Selection** tab to complete the following fields:

Field	Value
Initial Expressions subtab	Type: **cleanup($G.create.response.slo)** **cleanup($G.create.avail.slo)**
Select $L.file by subtab	
Select records "of type"	Select this option and click **Fill** to select **sla** as the record type.

Field	Value
using query	Type: **"agreement.id="+$G.new.sla** **Note**: Be sure to include the double quotes (") at each end of "agreement.id" as these quotes are part of the query.

6. Select the **Usage** tab and complete the following fields to determine the action that HP Service Manager should take when the wizard is run:

Field	Value
Request user input	Select this field to prompt the user for information.
Sub Format to Display	Enter a subformat name of the format to display. Type: **sla.create.finished**
Activate "Finish" option?	Since this is the final wizard in a series of wizard panels, select this option to make a "Finish" button appear on this wizard panel.
Turn off Next button only?	Select this checkbox.

7. Select the **Actions** tab to define the actions when a user clicks Next, Cancel, or Previous within the wizard. Complete the following field:

Field	Value
Perform Actions On	Here you can specify which record(s) will be affected by the action that you are going to perform. This can either be the current file, the list selected (as a whole), or an action to be performed on every single record selected. Select **Current File ($L.file)**.
Expressions subtab	Enter an expression that will run as part of the wizard. Type: **cleanup($G.new.sla)**
Display Record(s) when complete?	Select this option (to set to true) to display the record to the user when the wizard completes.

8. Select the **Next Wizard** tab and complete the following fields:

Field	Value
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Response 1**
Condition	The condition determines what happens next. In this case, you entered the next wizard in sequence and an expression that evaluates to "true." Alternatively, you can specify different wizazrds, based on user actions or selections. Type: **$G.create.response.slo=true**
Wizard Name	This is the name of the next wizard in a series of wizards. Type: **SLA Create Availability 1**
Condition	Type: **$G.create.avail.slo=true**

9. When you are finished, click **Save**.

You have created the last wizard panel that displays the SLA record number and title of the newly-created SLA record. Next, you will create format for this wizard step in Forms Designer.

Create a form for the completed SLA

For this task, you will use Forms Designer to create form controls display the record number and title of the new SLA record created. These form controls, together with the window title, prompt, buttons, and other choices you made in the wizard record, will be displayed to the end user as the wizard screen shown Step 13: Display new SLA number and title.

To create a new form for the completed SLA:

1. Start the Forms Designer.
2. In the Form field, type **sla.create.finished**.

   Best Practice: Because the form name must match the subformat display entry exactly, you can copy the subformat display entry in the Usage tab and paste it into the Form field. This helps avoid spelling errors.
3. Click **New**.
4. Click **No** for Forms wizard.
5. Create the following for this form. When you start to design the new form, make sure the X and Y coordinates are at 0 (zero).

Property	Value
Label	
Caption	Type: **SLA record**
Visible	Select.
Text	
Input	Type: **agreement.id**
Visible	Select.
Text	
Input	Type: **title**
Visible	Select.
Label	
Caption	Type: **has been successfully created. Click Finish to view the record.**
Visible	Select.

6. Click **OK** twice to save the new form and exit.

You have finished creating the final wizard screen that is displayed for the SLA wizard.

Step 14: Set menu to call the new SLA wizard

When we have e completed all of the previous steps, we must change the behavior of the menu so that when the user clicks **Service Level Management > Service Level Agreements > Create New SLA**, the first panel of the new wizard is displayed rather than the Service Level Agreement entry form that is displayed out-of-box.

To set the menu to call the new wizard:

1. Click **Tailoring > Tailoring Tools > Menus**.

2. In the Menu Name field, type: **SLA** and click **Search**.

3. Click in the row for **Option 5 – Service Level Agreements**.

4. In the Application column, replace thread.start by typing: **wizard.run**.

5. In the Parameter Value column, replace sla.add.object by typing: **SLA Create 1**.

6. Click **OK** to save the changes.

Now you can click **Service Level Management > Service Level Agreements > Create New SLA**to confirm that the menu changes were effected and the wizard opens and executes as expected.

5 Troubleshooting wizards

Check the topics in this section to troubleshoot issues that may appear when executing a wizard. Error logs can help with troubleshooting.

The following topics are discussed:

- Button options are not displaying properly
- File variables not passed into wizard flow
- Format not found
- JavaScript syntax errors
- Query failed to provide correct results
- Query returned a list of records rather than a single record
- RAD expression syntax errors
- Records are not sorted properly
- Wizard stops with "Could not continue" errors
- Run a trace

Button options are not displaying properly

If button options are not displaying or unexpectedly displaying, check the **Usage** tab to ensure that the correct checkboxes are enabled.

Also check the following:

- The first displayed panel should **not** have a Previous button.
- The last displayed panel **should** have a Finish button.
- If the Previous button is not displaying as expected, check to see if you selected the "Start Node?" option in the **Wizard Info** tab.

  Tip: Only the first wizard panel should have the "Start Node?" option selected in the **Wizard Info** tab.
- Do not choose the "Next" button as a "Finish" button. If you do, the wizard will stop executing. As a best practice, disable the "Next" button and enable the "Finish" button in the last panel of the wizard work flow.
- Check to see if any of the options on the **Usage** tab were suppressed by mistake.

JavaScript syntax errors

The wizard stops executing, and the user is returned to the calling application with the following error message(s) at the top of the screen:

```
Process panel <panel name> in RAD wizard.run encountered error in line 5
(wizard.run,<panel name>)
```

```
Process panel init.expr in RAD wizard.run encountered error in line 5
(wizard.run,init.expr)
```

```
Cannot evaluate expression (wizard.run,init.expr)
```

`Script <UNKNOWN> line 1: ERROR ReferenceError: test is not defined at char 1`

```
Unrecoverable error in application: se.call.process on panel call.rad.1
```

```
Unrecoverable error in application: wizard.run on panel init.expr
```

Cause

Invalid syntax was entered in a **JavaScript** tab. The RAD application and panel will give you a hint as to which JavaScript tab the syntax error was entered. Refer to the table below for help in determining where to make the fix.

RAD application/panel	Where to fix
`wizard.run,init.expr`	**JavaScript** subtab of the **File Selection** tab.
`wizard.run,select.file.setup`	Query statement in the **Select $L.file by** subtab of the **File Selection** tab.
`wizard.run,run.expressions`	**Expressions** subtab of the **Actions** tab.
`wizard.run,exit.cancel`	**Javascript Executed on Cancel** subtab of the **Cancel Expressions** tab.

Fix

Correct syntax for JavaScript expressions is discussed in the JavaScript Programmer's Guide. When using JavaScript, you must define variables before using them. For example:

```
var <name>=new String()
```

The correct syntax for assigning a value to a field is as follows:

```
Table.field1=value
```

To compare the values of two fields to each other, use the following syntax:

```
Table1.field1==table2.field2
```

File variables not passed into wizard flow

The wizard stops executing, and the user is returned to the calling application with the following error message at the top of the screen:

```
Wizard could not continue. No file variable was passed in.
```

Cause

In the **Select $L.file by** subtab on the **File Selection** tab, the option "$L.file passed in" was selected, but no file variable was passed in.

Fix

Either call `wizard.run` with the file parameter passed in correctly, or choose another option in the **Select $L.file by** tab on the **File Selection** tab. For example, select **Create a record of type**.

Format not found

The wizard stops executing, and the user is returned to the calling application with the following error message at the top of the screen:

```
Format "test" not found (display,show.rio)
```

Cause

An invalid format name was entered in the **Sub Format to Display** field on the **Usage** tab while using the "Request user input" option in the **Wizard Usage** section on the **Usage** tab.

Fix

Enter a valid format name in the **Sub Format to Display** field on the **Usage** tab, so a valid subformat will be displayed. Or, select the **Skip Display** option to bypass attempting to display the form.

Records are not sorted properly

According to the wizard setup, the list of selected records should be sorted by the values in a certain field. When the list of records is displayed, it is sorted by the unique key instead.

Cause

The type of field needed for the query (dbdict name) is invalid.

Fix

Enter a valid sort field in the record type dbdict.

Query failed to provide correct results

When entering a query in the **File Selection** tab, all strings such as field names must be enclosed in quotes. If the strings are not enclosed in quotes, the query will fail to provide the correct results.

For example:

```
"agreement.id="+G.new.sla
```

Or

```
"logical.name=\""+str($group.name.new)+"\""
```

Query returned a list of records rather than a single record

According to the wizard setup, a single record should be returned from the query entered in the **Usage** tab. Instead, a true search was performed.

Cause

Incorrect query syntax was used. For example, `agreement.id in $L.file=168`. Query statements should not include the file variable.

Fix

Use correct query syntax. For example, `"agreement.id="168`.

RAD expression syntax errors

The wizard stops executing, and the user is returned to the calling application with the following error message(s) at the top of the screen:

```
Process panel <panel name> in RAD wizard.run encountered error in line 2
(wizard.run,<panel name>)

Process panel init.expr in RAD wizard.run encountered error in line 2
(wizard.run,init.expr)

Cannot evaluate expression (wizard.run,init.expr)

Bad arg (2) oper = (wizard.run,init.expr)

Bad arg (3) oper index (wizard.run,init.expr)

Unrecoverable error in application: se.call.process on panel call.rad.1

Unrecoverable error in application: wizard.run on panel init.expr
```

Cause

These errors are caused by syntax errors in the wizard panels. The panel name gives you a hint as to where the issue occurred.

Bad arg (x) oper operator <y> indicates improper syntax of a statement or a missing or invalid parameter that is expected to be supplied to the statement.

Bad arg (1 or 2) oper <operator>: arg 1 indicates that the issue is to the left of the <operator>. If it is arg 2, then the issue is to the right of the <operator>. For example, operator can be defined as follows: = , < , > , ~= , ~< , ~> , etc.

Bad arg (1 or 2) oper <in> arg 1 indicates that the issue is to the left of "in" within the statement. If arg is 2, then the issue is to the right of "in" within the statement.

Bad arg (x) oper <function> indicates that the number of the parameter that was passed to the function is invalid.

The following table will help you find where in the wizard panel the error occurred and where to make the fix.

RAD application/panel	Where to fix
`wizard.run,init.expr`	**Initial Expressions** subtab of the **File Selection** tab.
`wizard.run,select.file.setup`	Query statement in the **Select $L.file by** subtab of the **File Selection** tab.
`wizard.run,get.selection.records`	Query statement in **Selection Criteria** on the **Usage** tab.
`wizard.run,run.expressions`	**Expressions** subtab of the **Actions** tab.
`wizard.run,decide.restart`	**Restart Panel if** field on the **Actions** tab.
`wizard.run,get.message`	**Condition** field of the **Messaging** tab.
`wizard.run,setup.wizard.variables`	Wizard variables on the **Variables** tab.
`wizard.run,find.next.wizard`	**Condition** field on the **Next Wizard** tab.
`wizard.run,exit.cancel`	**Expressions Executed on Cancel** subtab of the **Cancel Expressions** tab.

Fix

Correct syntax for RAD expressions is discussed in the System Language topics in the Service Manager Online Help. You can view this help by clicking **Help** > **Help Contents** if you are using a Windows client, or pressing **F1** if you are using a web client.

The correct syntax for assigning a value to a field is as follows:

```
<field> in $L.file=value
```

To compare the values of two fields to each other, use the following syntax:

```
<field1> in $L.file=<field2> in $L.file
```

Any concatenated statement containing a field name and a variable needs to be entered as follows:

`"agreement.id=" +$G.test` where $G.test is numeric or Boolean

```
"agreement.id =\"" + $G.test + "\""
```
where $G.test is a character

```
"agreement.id='" + $G.test + "'"
```
where $G.test is a date/time

Wizard stops with "Could not continue" errors

The wizard stops executing for a variety of reasons, including possible configuration errors. Some of the errors you may encounter are as follows:

Error	Cause/Fix
`Cannot create record of type "test."`	**Cause**: The record type entered does not exist as a record in the dbdict table. **Fix**: Enter a valid record type or dbdict name.
`Cannot select from record of type "test."`	**Cause**: The record type entered does not exist as a record in the dbdict table. **Fix**: Enter a valid record type or dbdict name.
`Cannot initialize record of type "slas."`	**Cause**: On the **Usage** tab, **Query for Records of type** has an invalid record type entered. **Fix**: Enter a valid record type or dbdict name.
`Invalid selection query: agreement=$G.test.`	**Cause**: On the **Usage** tab, an invalid field name was used in the **Query for Records** statement. **Fix**: Determine and use the correct field names in the statement.
`Wizard "test" does not exist. Contact your System Administrator.`	**Cause**: An invalid wizard name was entered on the **Next Wizard** tab. When a nonexistent wizard evaluates to true, the wizard workflow will be caught in an infinite loop and the session thread will be killed. **Fix**: Enter a valid wizard name on the **Next Wizard** tab.

`Display application could not find screen. Query=screen.id="wizard.test.me" and language="ENG"`
`Unrecoverable error in application: se.call.process on panel call.rad.1`
`Display application error exit has been taken. $L.ds.ids=L.ds.desc=`
	Cause: An invalid display screen name was selected on the **Usage** tab.
	Fix: On the **Usage** tab, enter a valid display screen name or leave the field empty for the default option.

Run a trace

If you receive an error message, use the following to trace the error to get more detail on the message you receive. You can also run a trace when the wizard flow does not take the expected path.

To run a trace:

1. Enter the **RTM:3** and **debugdbquery:999** parameters in the Service Manager `sm.ini` file.
2. Start a new client connection.

   For complete details, see the instructions in the *Diagnostics and Tuning* white paper.
3. Within the trace, search for the first call to the `wizard.run` RAD application.
4. Next, follow the flow of the wizard, checking the dbquery statements against the wizard table to see which wizard is called next.
5. Follow the trace to the error message if one was received, or follow the trace to understand which path the application took through the wizard flow.
6. Use print statements in the JavaScript expressions within the wizard panels, to write additional information to the log file. For example, the content of variables that will influence which wizard will be run next or statements as to which wizard is executed at the time.

   Note: Remember to remove these debug statements before putting the wizard into production.

Variables to Avoid Using in Wizards

When creating a wizard, you should not use local variables that are used by the **wizard.run** RAD application. Using those variables could cause errors and unexpected results.

The variables to avoid using when creating a wizard are listed below.

$L.action	$L.action.file	$L.action.file.save
$L.action.save	$L.addition	$L.allow.finish
$L.allow.skip	$L.base.format	$L.bitmap
$L.category	$L.disable.next	$L.disable.next.previous
$L.disable.previous	$L.display.screen	$L.ds.i
$L.ds.junk	$L.ds.nme	$L.ds.var
$L.env	$L.eval	$L.exit
$L.fc	$L.field.line.is	$L.field.name.is
$L.file	$L.file.save	$L.filename
$L.finish	$L.first.parent	$L.format
$L.format.name	$L.gen.ms	$L.i
$L.link	$L.list.field.name	$L.lng
$L.main.title	$L.messages	$L.mode
$L.ms	$L.msg	$L.msg.cond
$L.msg.lvl	$L.msg.type	$L.mult.sub
$L.next.wizard	$L.no.recs.msg	$L.object
$L.object.type	$L.old.format.name	$L.old.subformat
$L.option	$L.override.link	$L.par.lng
$L.parent	$L.parent.file	$L.parent.id
$L.parent.object	$L.pass.names	$L.pass.values
$L.passed.file	$L.phase	$L.prev.wizard
$L.prompt	$L.qbe	$L.qbe.format
$L.same	$L.save.name	$L.selected

$L.selection	$L.selection.filename	$L.selection.save
$L.selection.sort	$L.selection.sql	$L.sql
$L.start	$L.sub.format	$L.title
$L.var	$L.window.title	$L.wizard
$L.wizard.bitmap	$L.wizard.error.msg	$L.wizard.format
$L.wizard.name	$L.wizard.names	$L.wizard.prompt
$L.wizard.sql	$L.wizard.title	$L.wizard.values
$L.x	$exit	$normal
$error		