HP OpenView Service Desk 4.5

Documentation Addendum
for
Service Pack 8

Second Edition

Manufacturing Part Number: N/A

June 2004

Legal Notices

Hewlett-Packard makes no warranty of any kind with regard to this manual, including, but not limited to, the implied warranties of merchantability and fitness for a particular purpose. Hewlett-Packard shall not be held liable for errors contained herein or direct, indirect, special, incidental or consequential damages in connection with the furnishing, performance, or use of this material.

Restricted Rights Legend. Use, duplication or disclosure by the U.S. Government is subject to restrictions as set forth in subparagraph (c)(1)(ii) of the Rights in Technical Data and Computer Software clause in DFARS 252.227-7013.

Hewlett-Packard Company
3000 Hanover Street
Palo Alto, CA 94304 U.S.A.

Rights for non-DOD U.S. Government Departments and Agencies are as set forth in FAR 52.227-19 (c)(1,2).

The nomenclature of each version of this software (and manuals therefore) has been devised for commercially convenient reasons, and is not intended to denote the degree of originality of any version of the software with respect to any other version. The extent of protection afforded by, and duration of copyright is to be determined entirely independently of this nomenclature.

Trademark Notices.

Adobe® Acrobat® Reader is a trademarks of Adobe Systems Incorporated.

Helvetica* is a trademark of Linotype-Hell AG and/or its subsidiaries in the U.S. and other countries.

Java™ is a U.S. trademark of Sun Microsystems, Inc.

Oracle is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX is a registered trademark of the Open Group.

Windows (NT) is a U.S. registered trademark of Microsoft Corporation.

Contents

Preface

This document explains both the new features of Service Pack 7 and Service Pack 8. Smaller enhancements can be found in the Service Pack \doc\readme.htm file. For information about other Service Desk 4.5 functionality, refer to the manuals listed in “Related Publication” on page 9.

The main new features of Service Pack 7 are:

- Enhancements to UI rules capabilities
- A new Service Impact View that aids in estimating the potential impact of failures
- Semi-automatic merging of conflicting concurrent changes to forms

The main new feature of Service Pack 8 is:

- Importing Operational Level Services into Service Desk

Revision History

When an edition of a manual is issued with a software release, it has been reviewed and tested and is therefore considered correct at the date of publication. However, errors in the software or documentation that were unknown at the time of release, or important new developments, may necessitate the release of a service pack that includes revised documentation. Revised documentation may also be published on the Internet, See the section ”We Welcome Your Comments” in this preface for the URLs.

A revised edition will display change bars in the left-hand margin to indicate revised text. These change bars will only mark the text that has been edited or inserted since the previous edition or revised edition.

When a revised edition of this document is published, the latest revised edition nullifies all previous editions.

Table 1.1 Revision History

Edition and Revision Number	Issue Date	Product Release
First Edition	April 2004	Service Desk 4.5 Service Pack 8
Second Edition	June 2004	Service Desk 4.5 Service Pack 9

NOTE This manual describes the new functionalities introduced with Service Desk 4.5 Service Pack 8. The difference between the first and second edition of this manual is that the appendix B is rewritten so the extra feature is easier to understand. In the first edition the only difference with the Service Pack 7 manual is the added section “Importing Operational Level Services into Service Desk”.

Related Publications

This section helps you find information that is related to the information in this guide. It gives an overview of the Service Desk documentation and lists other publications you may need to refer to when using this guide.

The Service Desk Documentation

Service Desk provides a selection of books and online help to assist you in using Service Desk and improve your understanding of the underlying concepts. This section illustrates what information is available and where you can find it.

NOTE This section lists the publications provided with Service Desk 4.5. Updates of publications and additional publications may be provided in later service packs. For an overview of the documentation provided in service packs, please refer to the Readme file of the latest service pack. The service packs and the latest versions of publications are available on the Internet. See the section ”We Welcome Your Comments! " in this preface for the URLs.

- The `Readme.htm` file on the Service Desk CD-ROM contains information that will help you get started with Service Desk. It also contains any last-minute information that became available after the other documentation went to manufacturing.
- The *HP OpenView Service Desk: Release Notes* give a description of the features that Service Desk provides. In addition, they give information that helps you:
  - Compare the current software’s features with those available in previous versions of the software;
  - Solve known problems.

  The Release Notes are available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Release_Notes.pdf`.
- The *HP OpenView Service Desk: User’s Guide* introduces you to the key concepts behind Service Desk. It gives an overview of what you can do with Service Desk and explains typical tasks of different types of Service Desk users. Scenario descriptions are provided as examples of how the described features could be implemented.

The User's Guide is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `User's_Guide.pdf`.

- The *HP OpenView Service Desk: Supported Platforms List* contains information that helps you determine software requirements. It lists the software versions supported by Hewlett-Packard for Service Desk 4.5.
  The Supported Platforms List is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Supported_Platforms_List.pdf`. It is also online available at http://openview.hp.com/sso/ecare/getsupportdoc?docid=OV-EN007678
- The *HP OpenView Service Desk: Installation Guide* covers all aspects of installing Service Desk.
  The Installation Guide is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Installation_Guide.pdf`.
- The *HP OpenView Service Desk: Administrator' s Guide* provides information that helps application administrators to set up and maintain the Service Desk application server for client usability.
  The Administrator's Guide is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Administrator's_Guide.pdf`.
- The *HP OpenView Service Desk: Data Exchange Administrator's Guide* explains how you can use data from other applications in Service Desk. It explains the underlying concepts of the data exchange process and gives step-by-step instructions on exporting data from external applications and importing it into Service Desk. The data exchange process includes importing single service events and batches of data.
  The Data Exchange Administrator's Guide is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Data_Exchange.pdf`.
- The *HP OpenView Operations Integration Administrator's Guide* explains the integration between Service Desk and HP OpenView Operations for Windows and UNIX®. This guide covers the installation and configuration of the integration and explains how to perform the various tasks available with the integration.
  The HP OpenView Operations Integration Administrator's Guide is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is OVO`_Integration_AG.pdf`.
- The *HP OpenView Service Desk: Web API Programmer's Guide* contains information that will help you create customized

integrations with Service Desk using the Service Desk Web API. This API is particularly suited for developing Web applications.
The Web API Programmer's Guide is available as a PDF file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Web_API_pg.pdf`.

- The *HP OpenView Service Desk: Migration Guide* provides a detailed overview of the migration from ITSM to Service Desk, to include an analysis of the differences in the two applications. Detailed instructions in this guide lead through the installation, configuration and other tasks required for a successful migration. The Migration Guide is available as a PDF file on the HP OpenView Service Desk 4.5 for Windows CD-ROM. The file name is `Migration_Guide.pdf`.
- The *HP OpenView Service Desk: Data Dictionary* contains helpful information about the structure of the application.
  The Data Dictionary is available as an HTML file on the HP OpenView Service Desk 4.5 CD-ROM. The file name is `Data_Dictionary.htm`.
- The *HP OpenView Service Desk 4.5 Computer Based Training* (CBT) CD-ROM is intended to assist you in learning about the functionality of HP OpenView Service Desk 4.5 from both a user and a system administrator perspective. The CD-ROM contains demonstration videos and accompanying texts that explain and show how to perform a wide variety of tasks within the application. The CBT also explains the basic concepts of the Service Desk application.
  The *HP OpenView Service Desk 4.5 Computer Based Training* (CBT) CD-ROM will be shipped automatically with the regular Service Desk software. The CBT will be available for shipment shortly after the release of the Service Desk software.
- The online help is an extensive information system providing:
  - Procedural information to help you perform tasks, whether you are a novice or an experienced user;
  - Background and overview information to help you improve your understanding of the underlying concepts and structure of Service Desk;
  - Information about error messages that may appear when working with Service Desk, together with information on solving these errors;
  - Help on help to learn more about the online help.

  The online help is automatically installed as part of the Service Desk application and can be invoked from within Service Desk. See the following section entitled "Using the Online Help" for more information.

Reading PDF Files

You can view and print the PDF files with Adobe® Acrobat® Reader. This software is included on the HP OpenView Service Desk 4.5 CD-ROM. For installation instructions, see the `readme.htm` file on the CD-ROM.

The latest version of Adobe Acrobat Reader is also freely available from Adobe's Internet site at http://www.adobe.com.

Using the Online Help

You can invoke help from within Service Desk in the following ways:

- To get help for the window or dialog box you are working in, do one of the following:
  - Press **F1**.
  - Click the help toolbar button  .
  - Choose `Help` from the `Help` menu.
  - Click the help command button 

 in a dialog box.
- To search for help on a specific subject using the table of contents or the index of the help system: choose `Help Contents & Index` from the `Help` menu.

When you are in the help viewer, you can find help on how to use the help system itself by clicking the `Help` toolbar button:

Service Desk also provides tooltips and "`What's This?`" help for screen items like buttons, boxes, and menus.

A tooltip is a short description of a screen item. To view a tooltip, rest the mouse pointer on the screen item. The tooltip will appear at the position of the mouse pointer.

"`What's This?`" help is a brief explanation of how to use a screen item. "`What's This?`" help generally gives more information than tooltips. To view "`What's This?`" help:

1 First activate the "What's This?" mouse pointer in one of the following ways:

   - Press Shift+F1.
   - Click the "What's This?" toolbar button  .
   - Choose "What's This?" from the Help menu.
   - In dialog boxes, click the question mark button  in the title bar.

   The mouse pointer changes to a "What's This?" mouse pointer  .

2 Then click the screen item for which you want information. The "What's This?" help information appears in a pop-up window.

To close the pop-up window, click anywhere on the screen or press any key on your keyboard.

Typographic Conventions

The table below illustrates the typographic conventions used in this guide.

Font	What the Font Represents	Example
Italic	References to book titles	See also the *HP OpenView Service Desk: Installation Guide*.
	Emphasized text	*Do not delete* the System user.
Bold	First-time use of a term that is explained in the glossary	The service call is the basis for incident registration.
`Courier`	Menu names	You can adjust the data view with the commands in the `View` menu.
	Menu commands	Choose `Save` from the menu.
	Button names	Click `Add` to open the Add Service call dialog box.
	File names	To start the installation, double-click `setup.htm`.
	Computer-generated output, such as command lines and program listings	If the system displays the text `C:\>dir a: The device is not ready` then check if the disk is placed in the disk drive.
`Courier bold`	User input: text that you must enter in a box or after a command line	If the service call must be solved within 30 minutes, enter `30`.
`Courier italic`	Replaceable text: text that you must replace by the text that is appropriate for your situation	Go to the folder *`X`*`:\\Setup`, where *`X`* is your CD-ROM drive.

Helvetica bold	Keyboard keys	Press Ctrl+F1.
	A plus sign (+) means you must press the first key (Ctrl in the example), hold it, and then press the second key (F1 in the example).	

We Welcome Your Comments!

Your comments and suggestions help us understand your needs, and better meet them. We are interested in what you think of this manual and invite you to alert us to problems or suggest improvements. You can submit your comments through the Internet, using the HP OpenView Documentation Comments Web site at the following URL:

http://ovweb.external.hp.com/lpe/comm_serv

If you encounter errors that impair your ability to use the product, please contact the HP Response Center or your support representative.

The latest versions of OpenView product manuals, including Service Desk manuals, are available on the HP OpenView Manuals Web site at the following URL:

http://ovweb.external.hp.com/lpe/doc_serv

Software patches and documentation updates that occur after a product release, will be available on the HP OpenView Patches Web site at the following URL:

http://support.openview.hp.com/cpe/patches

1 Rules

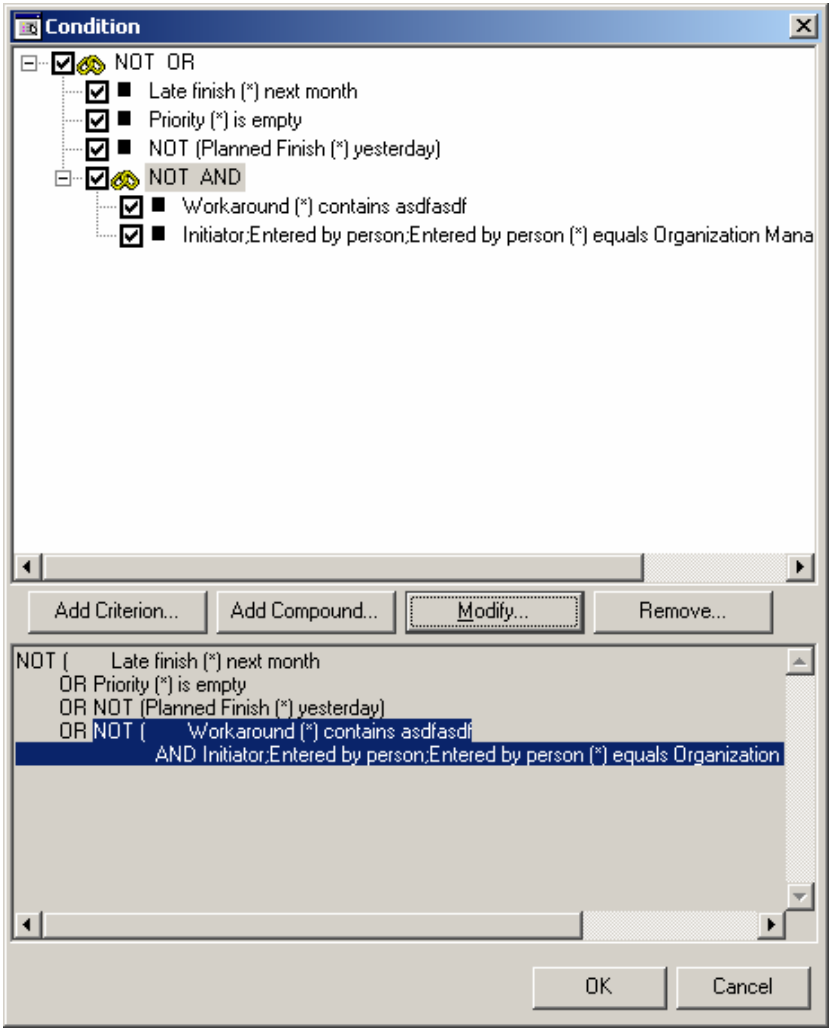
A rule consists of actions that are executed when certain specified conditions are met. They can be divided into User Interface (UI) rules and Database (DB) rules. For more information on these rules, see chapter 4 of the Administrator's Guide.

This chapter describes some of the improvements to the rule functionality.

Conditions

A condition consists of criteria that determine when an action is executed. Conditions can consist of simple criteria, complex compounds of criteria, or other compounds.

Figure 1.1 Creating a Condition

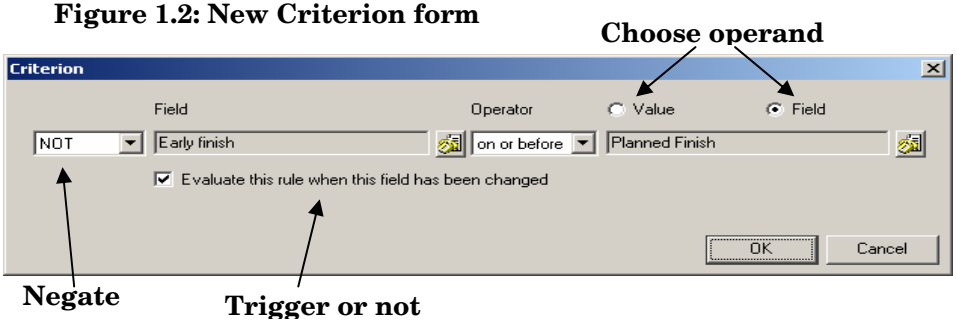


Criteria

The following new functionality is available for criteria:

- Negated criteria
- Field comparison
- New operator types and field updates do not automatically trigger an evaluation of the condition for the rule

Figure 1.2: New Criterion form



Field to Field Comparison

It is now possible to compare a field with another field. The options are:

- When the `Value` radio button is selected, a value is entered in the text box. The field is then compared to this value. (Standard functionality.)
- When `Field` is selected, you can choose a field for comparison.

Removed Operators

You can negate all criteria, so the following operators are now redundant:

- `Not Equal`
- `Not Between`
- `Not Contains`
- `Not Empty`

Since the above criteria no longer exist they are now automatically converted to their positive counterparts and then assigned a NOT value.

Is Current Operator

This operator is available for criteria with a `Person` or `Account` field. See the section “Variables” for more information.

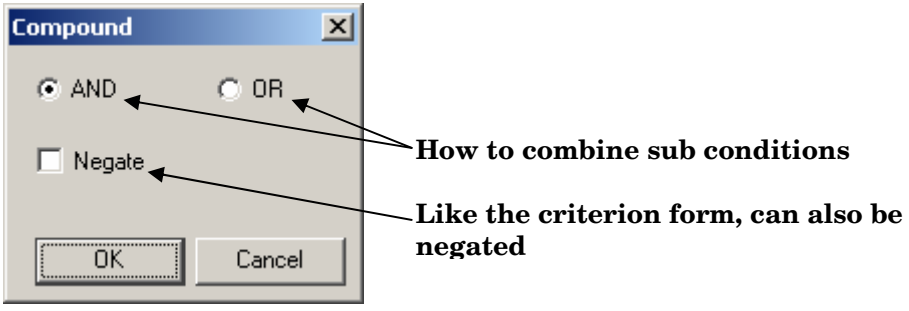
Compounds

A compound is the combination of two or more sub conditions (criteria or other compounds).

There are two ways to combine sub conditions:

- `AND` means the condition is true when all sub conditions are true
- `OR` means the condition is true when at least one sub condition is true

Figure 1.3: Creating a compound condition



New UI Rule Triggers

Actions associated with rules are executed whenever the corresponding conditions are met. The condition for a rule is not checked continuously because that causes an unnecessary drain on resources. Service Desk only checks a condition when a trigger is activated.

Before this release, rule-conditions were evaluated whenever any of the fields used in the condition changed. You can now suppress that evaluation on a field-by-field basis, by using the extra check mark in the criterion form.

The following sections describe the new trigger options:

- **Opening an Item** - This rule is checked whenever a form is opened. This is useful when used in combination with a User Notification Messages (see “User Notification Messages”). For example, on opening an overdue Service Call form a large, red, blinking banner is displayed.
- **Opening an Existing Item** - This rule is the same as the Opening an Item trigger, except that it does not execute when creating a new item.
- **After Opening a New Item** - This rule is the same as the Opening an Item trigger, except that it does not execute when opening an existing item.
- **Before Saving an Item** - This trigger checks the rule before the data is saved to the database. This is useful for checking for any data errors or starting an external process to send data to some other system.

Actions

This section describes the new Actions available with this Service Pack.

Data Update With External System

This action allows you to send the fields of an associated object to an external HTTP server and receive new values for these or other fields. See the following two figures.

Figure 1.4 A Data Update Action

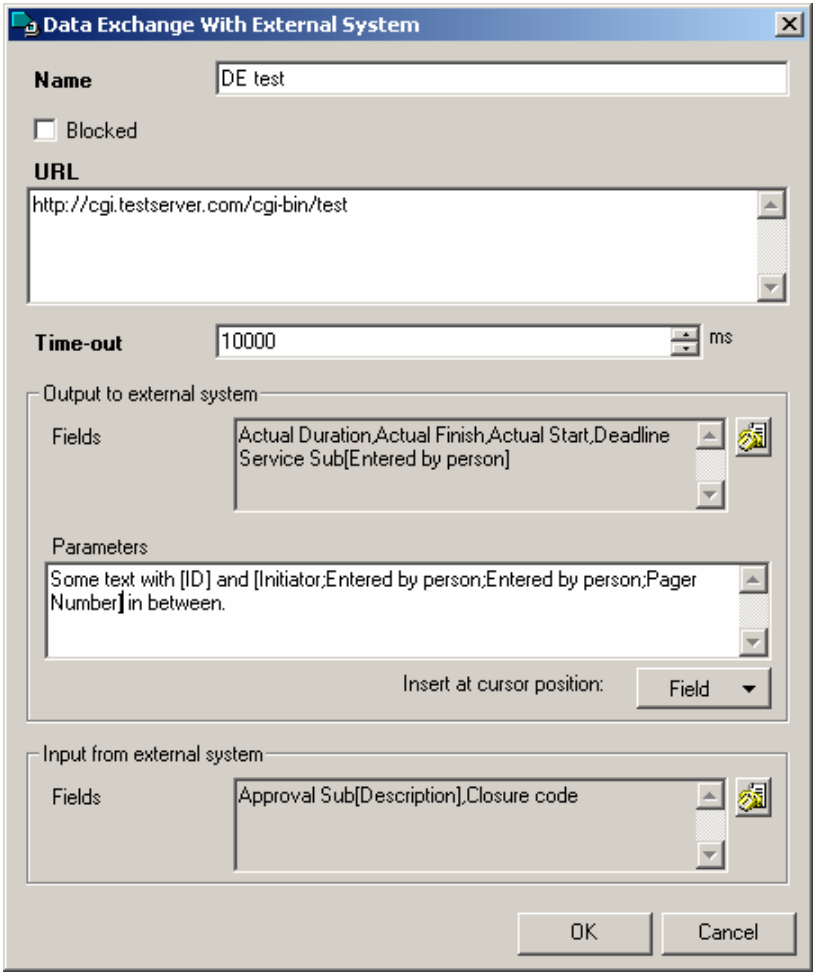
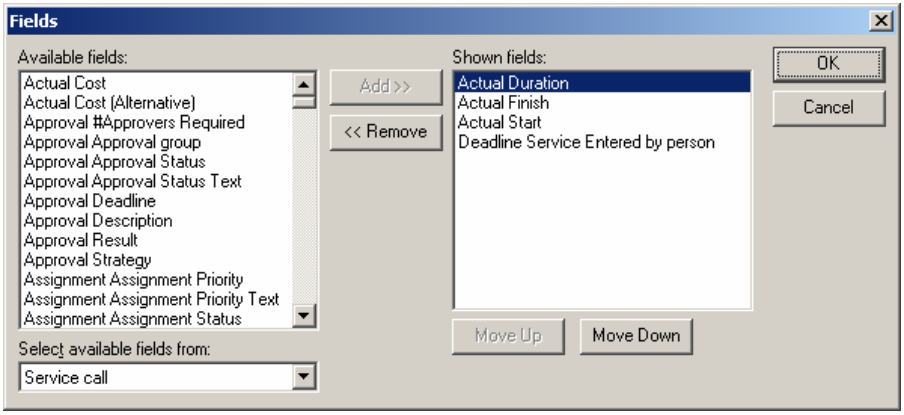


Figure 1.5 Selecting Output Fields for the Data Update Action



When this action is executed, the system performs an `HTTP POST` request to the specified URL[1]. All the output fields are encoded (as in the `POST` body) so that they can be read using standard web server programming techniques.

On the web server any action can be taken and new values returned to the input fields. These fields are set and in turn can execute other UI Rules.

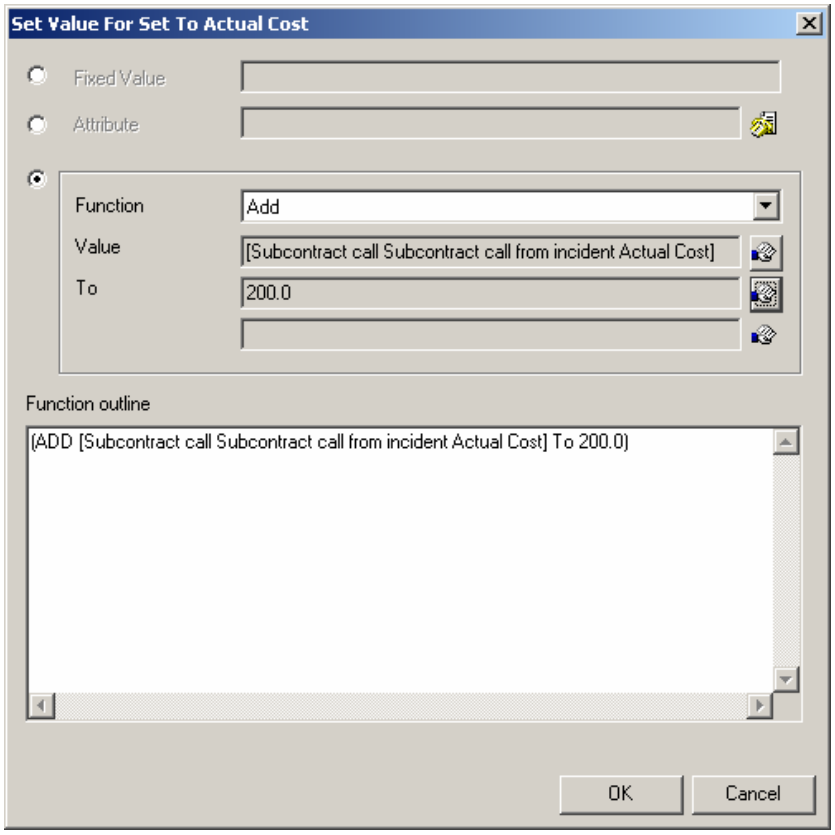
For the protocol used to communicate between Service Desk and the web server (and the encoding and decoding of parameters), See Appendix B Data Update Protocol .

Update Data Enhancements

In this release it is possible to calculate the value that is assigned to a field in an Update Data action using nested functions.

[1] See RFC2616, ftp://ftp.isi.edu/in-notes/rfc2616.txt

Figure 1.6 Function Editor



You can use the Function Editor to enter functions, see the figure above. For example:

```
(SUBTRACT DATE FROM [Planned Finish] With

    (CURRENT DATE))
```

Functions can be grouped by result data type, as described in the following sections.

Numeric functions

`Count characters from` ***`StringFunction`***

Subtract from *NumericFunction* with *NumericFunction*
Multiply by *NumericFunction* by *NumericFunction*
Divide by *NumericFunction* with *NumericFunction*
Add *NumericFunction* to *NumericFunction*

String functions

Take part of *StringFunction* from position *ConstantNumber* number of chars *ConstantNumber*
Trim right-hand characters of *StringFunction* number of chars *ConstantNumber*
Trim left-hand characters of *StringFunction* number of chars *ConstantNumber*
Remove trailing spaces of *StringFunction*
Remove heading spaces of *StringFunction*
Substitute for *StringFunction*
Convert case to upper for *StringFunction*
Convert case to lower for *StringFunction*
Concatenate *StringFunction* with *StringFunction*
Current person

Where:

- Current Person is the full name of the currently user.
- StringFunction can be replaced by any string function.
- ConstantNumber is a number (with a integer value greater than 0)

Date functions

Current date
Add to current date *TimeSpan*
Subtract from Current date *TimeSpan*
Subtract timespan from *DateFunction* timespan *TimeSpan*
Add timespan to *DateFunction* timespan *TimeSpan*

- Where:*DateFunction* is any date function
- *TimeSpan* is the Subtract Date function (see the section about Special Functions below) or a constant amount of time expressed in months, days, hours, minutes and seconds.
- Make empty clears the field. See the section about Special Functions below.

Special functions

`Make empty`: Used as the value of the `Update Data` action (field is cleared). Otherwise, it is used as the parameter for a nested function and it inhibits the assignment. This function can be used with any field type.

`Subtract Date from DateFunction with DateFunction:` This function results in a `timespan`, the difference between the two dates.

User Notification Messages

In this release, UI Rules (**not** DB rules) can show configurable messages to the user as an action. These messages can have different degrees of severity:

Informational messages

These are simple informational messages. The user can only confirm these messages and the actions will continue.

Warnings

With Warning messages the user can choose to continue or cancel. When the user chooses to cancel, further actions and the trigger (such as saving an item) are also canceled.

Error messages

The user has no options when this message is displayed, the action is canceled.

2 The Impacted Services View

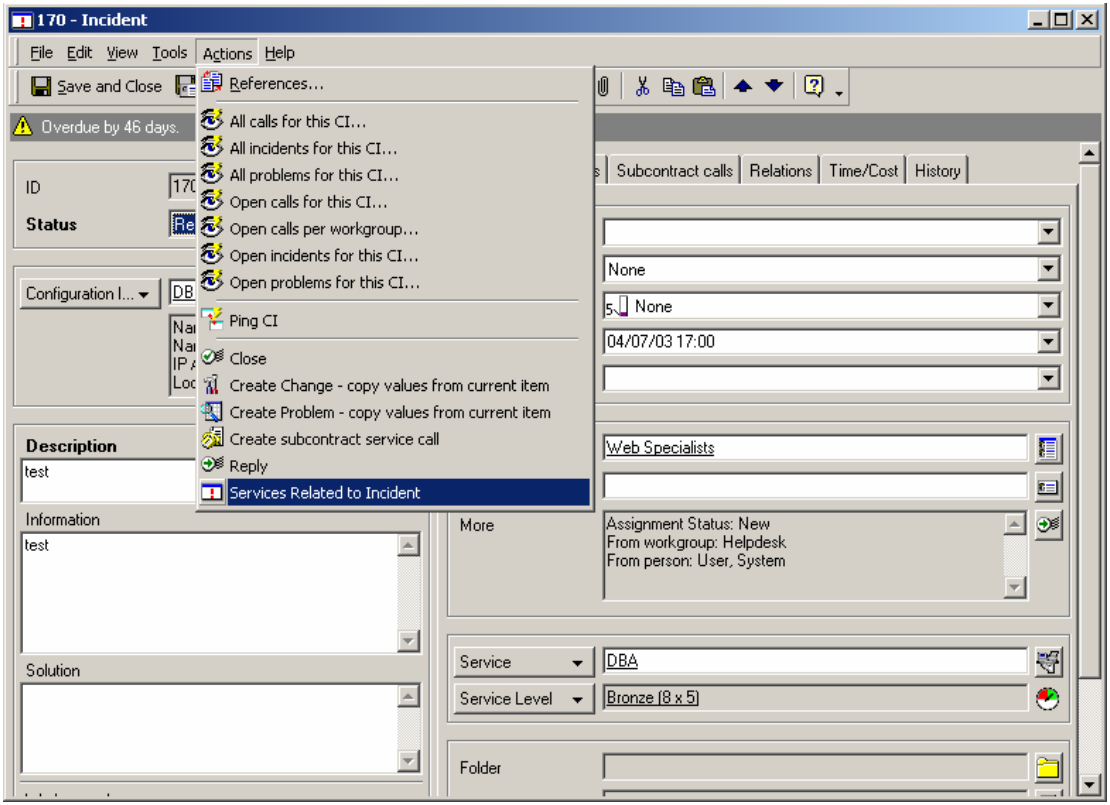
The Impacted Services View shows all Services that might be impacted by an Incident or Service Call. The primary purpose is to allow you to contact customers who may be affected by the Incident/Service Call as quickly as possible, without having to open several Service Desk Forms to find the necessary information. The list of Services that are potentially impacted are sorted by the appropriate **Deadline** and **Service Level**.

Starting the View

The Impacted Services View is started from the `Incident` form or the `Service Call` form through the `Actions` menu.

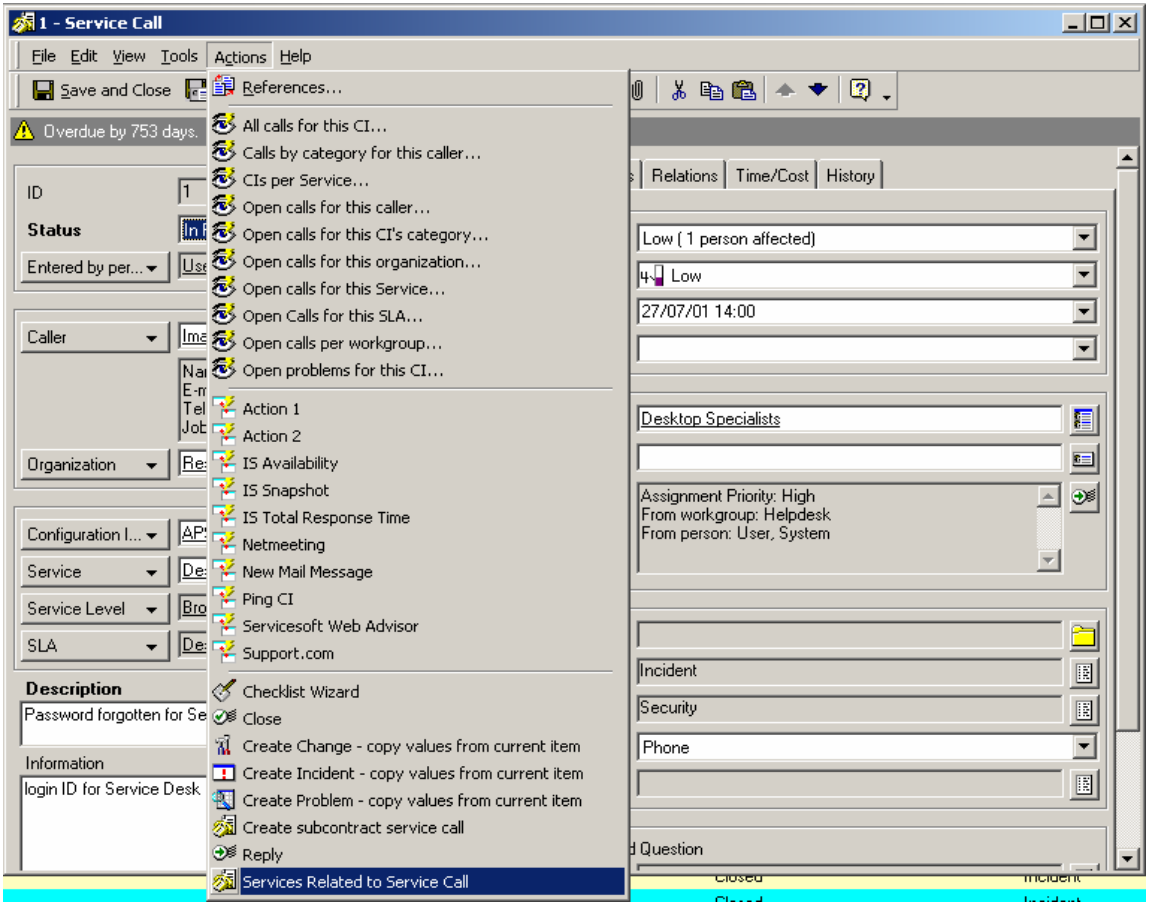
To start the view from the `Incident` form, select the `Actions` menu and then select the `Services Related to Incident` option, see the following figure:

Figure 2.1 Starting the Impacted Services View



To start the view from the `Service Call` form, select the `Actions` menu and then select the **`Services Related to Service Call`** option; see the following figure:

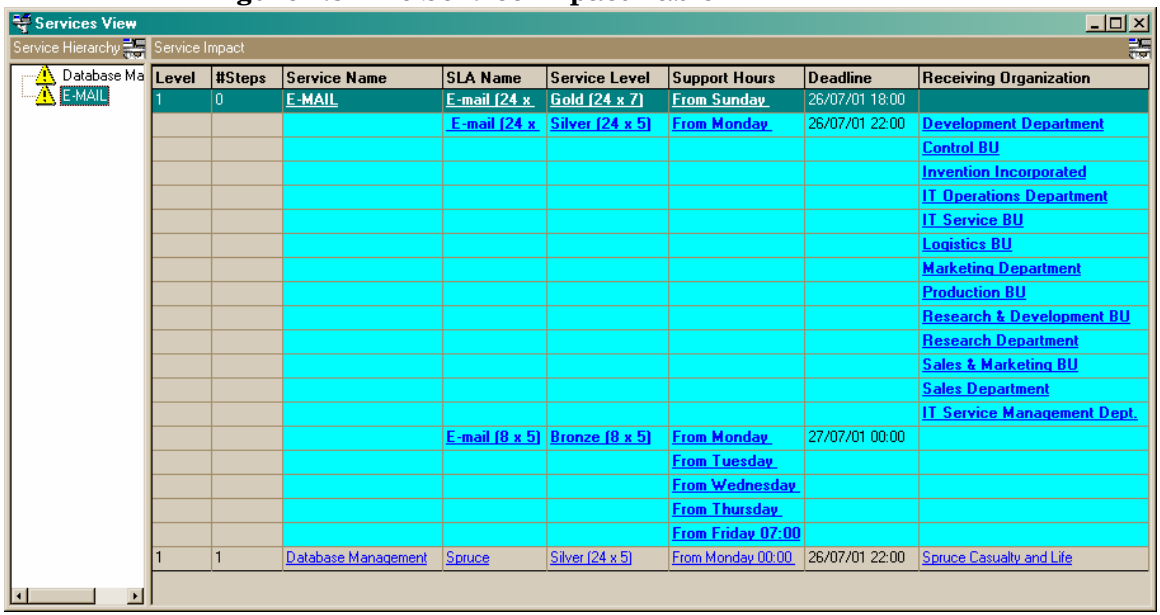
Figure 2.2 Impacted Services View in the Service Call Form



Using the Impacted Services View

The view consists of two parts: the `Service Hierarchy` tree and the `Service Impact` table. See the following figure:

Figure 2.3 The Service Impact Table



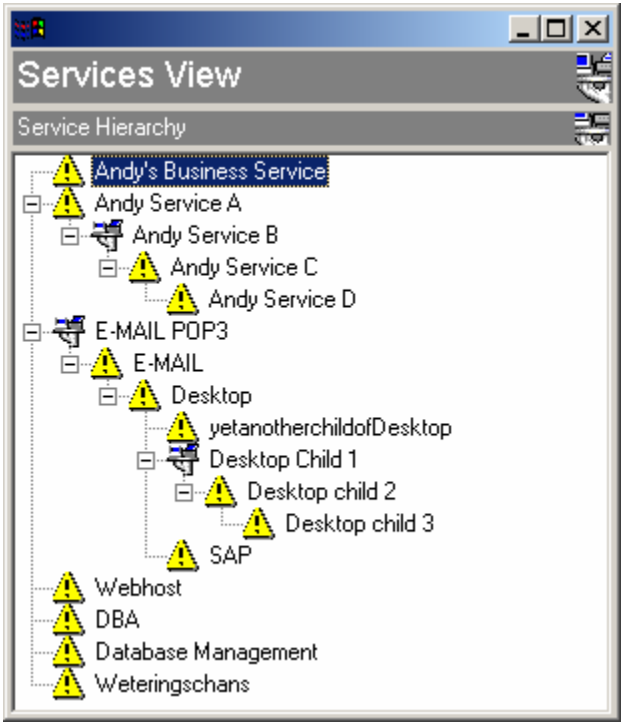
Service Hierarchy

The left side of the view contains the Service Hierarchy tree. This tree contains all potentially impacted services and services that may not be impacted but are part of the hierarchy of an impacted Service (for example, a parent service of an impacted service are shown even if it is not impacted).

Table 2.1Service Hierarchy Tree Icons

Icon	Explanation
	Impacted Services
	Non-impacted services

Figure 2.4 The Service Hierarchy Tree



Locating Impacted Service Information

By selecting an Impacted Service in the tree, the related Service information is automatically highlighted in the `Service Impact` table. This allows you to quickly locate information on a specific Service. If the `Service Impact` table has many entries you can use the scroll bar on the right of the Service Impact table to see the highlighted Service.

NOTE No information appears in the `Service Impact` table if you select a Service in the tree that is not impacted.

Opening the Standard Service Form

To open the `Service Desk Services` form, double-click on any Service in the Service tree. The form contains all the information available for that service.

Service Impact Table

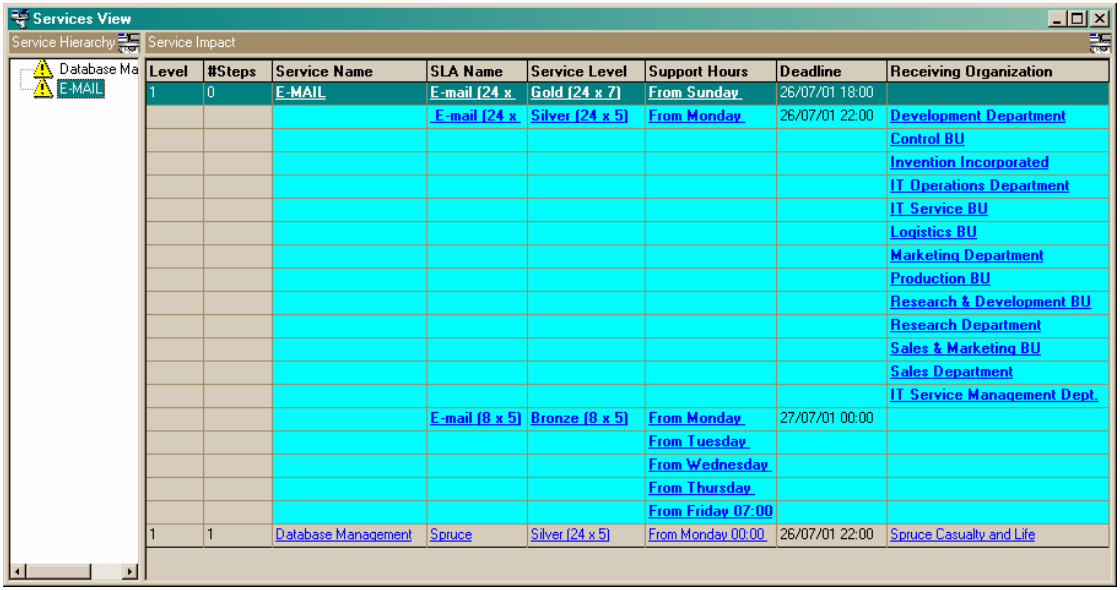
The right side of the view contains the `Service Impact` table, see the following Figure. The table below contains a list of all potentially impacted Services to the CI and/or Service listed in the Incident or Service Call form. In addition to the impacted Services, the table also contains the following information related to each Service:

Table 2.2 Services Related to the CI and/or Service

Level	Position in the Services Hierarchy.
`#Steps`	Distance in the CI / Services graph to the closest definitely impacted CI / Service.
`Service Level`	The `Name1` value of the Service Level in the SLA associated with this the Service.
`Support Hours`	The amount of time that support is provided, as stated in the Service Level in the SLA.
`Deadline`	The date and time at which the Incident needs to have been resolved in order not to exceed SLA conditions.
`Receiving Organization`	The `Name1` value of the Organization that is listed in the SLA as a receiving party of the Service under the SLA.

NOTE If more than one SLA or Receiving Organization is associated with a Service, the information is displayed on multiple rows in the table (see the following figure).

Figure 2.5 The Service Impact Table



Level	#Steps	Service Name	SLA Name	Service Level	Support Hours	Deadline	Receiving Organization
1	0	E-MAIL	E-mail (24 x	Gold (24 x 7)	From Sunday	26/07/01 18:00	
			E-mail (24 x	Silver (24 x 5)	From Monday	26/07/01 22:00	Development Department
							Control BU
							Invention Incorporated
							IT Operations Department
							IT Service BU
							Logistics BU
							Marketing Department
							Production BU
							Research & Development BU
							Research Department
							Sales & Marketing BU
							Sales Department
							IT Service Management Dept.
			E-mail (8 x 5)	Bronze (8 x 5)	From Monday	27/07/01 00:00	
					From Tuesday		
					From Wednesday		
					From Thursday		
					From Friday 07:00		
1	1	Database Management	Spruce	Silver (24 x 5)	From Monday 00:00	26/07/01 22:00	Spruce Casualty and Life

Table Order

The information in the `Service Impact` table is sorted by the closest SLA Deadline, followed by the Service Level associated with the SLA. Each Service Level has a Service Level Weight value (set by the Systems Administrator) that determines the relative importance of the Service Level. For example, a Gold (24 x 7) Service Level may have a Service Level Weight that is higher that a Bronze (8 x 5) Service Level.

Example:

Two Services are listed in the table with the same Deadline (both date and time) but one Service has a Gold (24 x 7) Service Level and the other has a Bronze (8 x 5) Service Level. The Service with the Gold (24 x 7) Service Level should appear in the table first if the associated Service Level Weight is set higher than the Bronze (8 x 5) Service Level Weight.

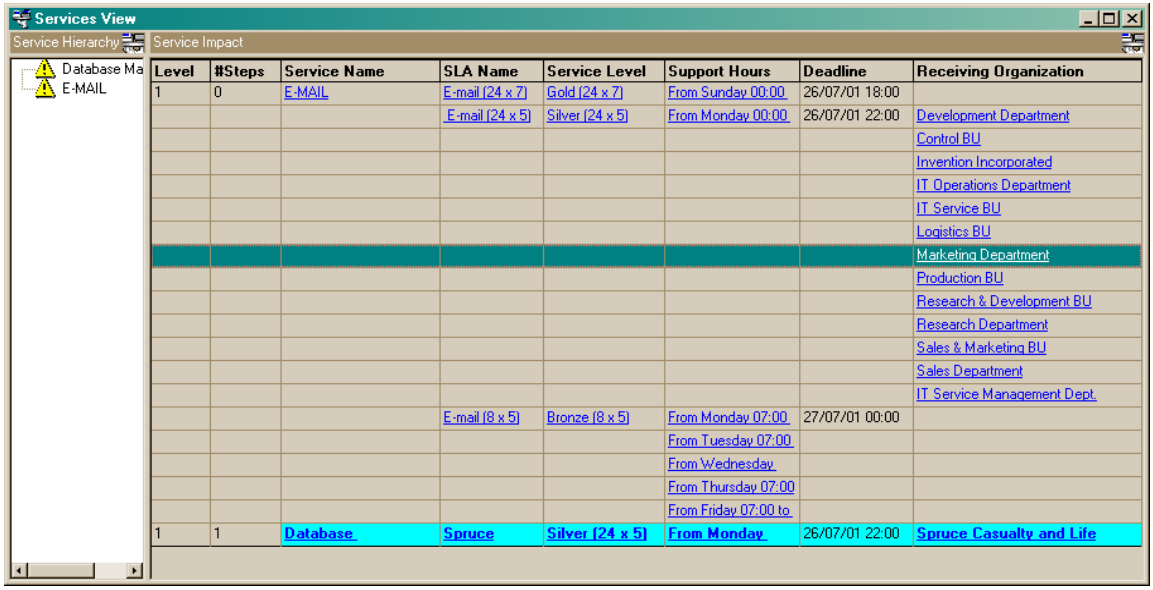
To summarize, the table has been designed so that the information at the top of the table has the greatest impact on the customer. This results in a quicker response to the customer.

Hyperlinks

- The following columns in the table are hyper links to the standard Service Desk forms:Service Name
- SLA Name
- Service Level
- Support Hours
- Receiving Organization

Double-clicking on any one of the above column headings in the `Service Impact` table starts the associated form containing the related information. For example, double-clicking on the `Service Name` column starts the `Services` form containing relevant information to the Service clicked-on.

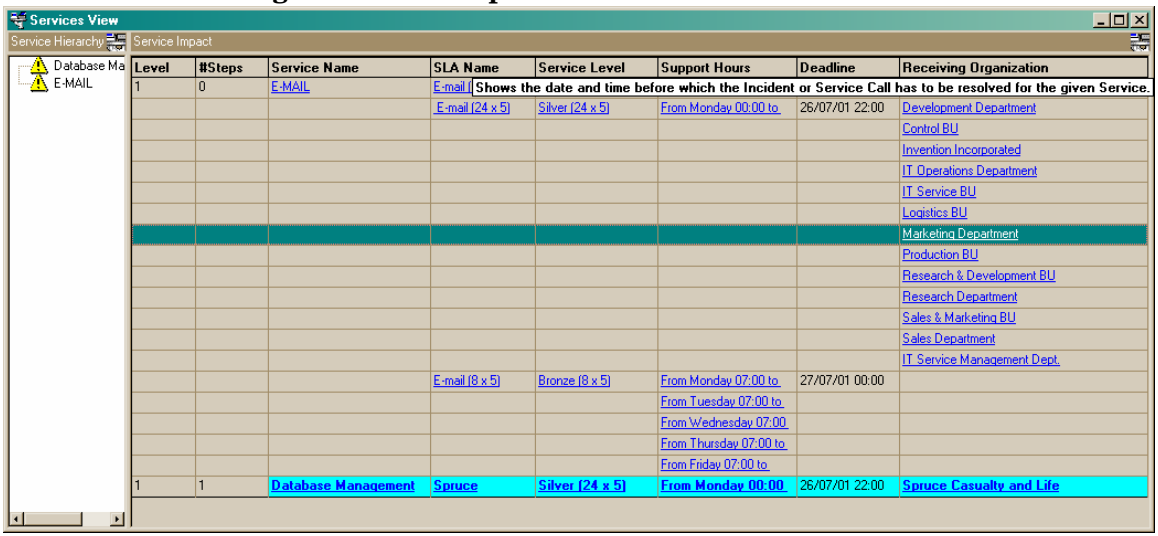
Figure 2.6 Hyperlinks



Tool-tips

To find a description of a column in the table, move the mouse over a column header (for example, `Service Name`) and a tool-tip opens underneath the header; the tip containing a short description of the column. See the following figure:

Figure 2.7 Tool-tip



3 Service Call Enhancements

This chapter describes new and improved search methods added to the `Service Call` form.

Assisted selection of CI by Organization

On the `Service Call` form a new item called `CIs owned by Organization` appears on the `Configuration Item` pop-up menu. To display a list of all CI's owned by the entered Organization, select the `CIs owned by Organization` item from the pop-up menu.

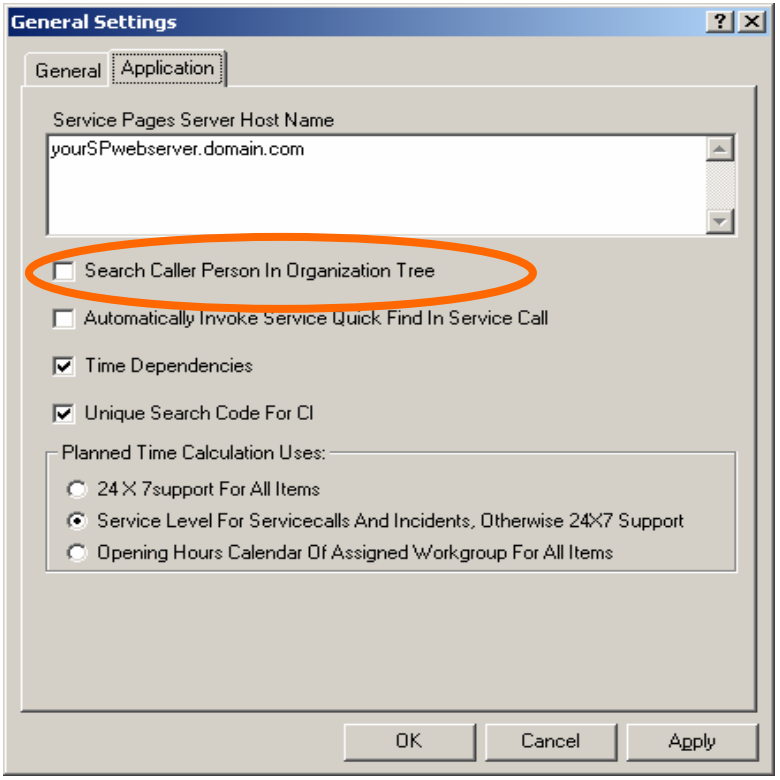
Assisted selection of service by CI

To pick a service that is using a particular Configuration Item (this then is a *Business Service*), select `Business Services using Configuration Item` from the `Service` pop-up menu in the `Service Call` form.

Hierarchical Search for Caller in Organization

After entering an Organization in a Service Call, a search pop-up window allows the user to select the Caller (persons **directly** belonging to that Organization). In this release, this can also include persons belonging to sub organizations. To enable this function use the following administrator setting (circled):

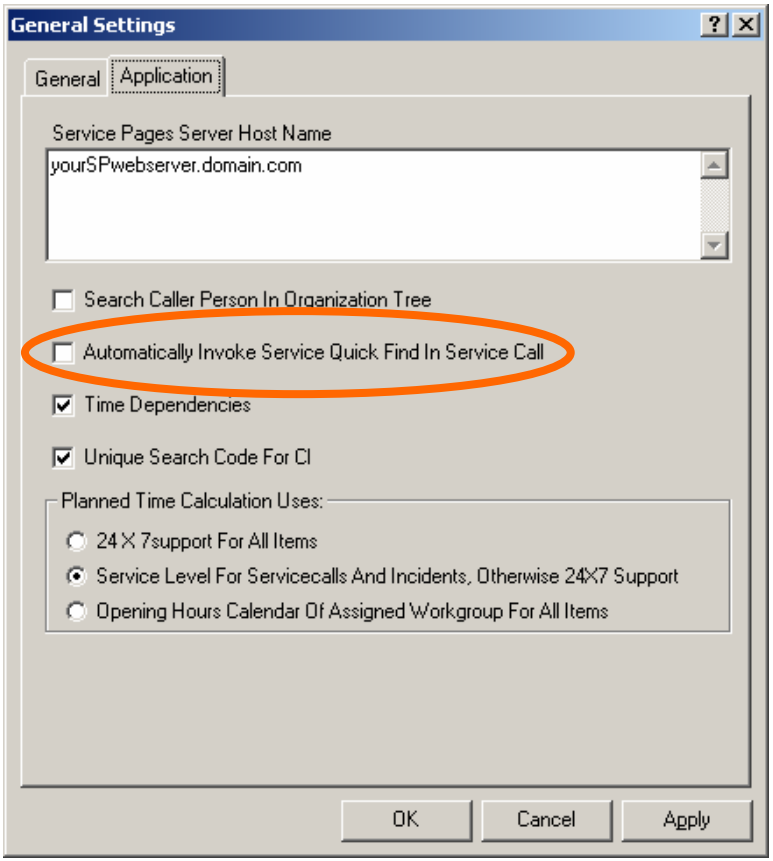
Figure 3.1 General Settings – Selecting Caller



Disable Auto-Pop up of Services Quick-Find Dialog Box

To disable the functionality that allows users to pick a service from the Quick Find window, go to the `Applications` tab of the `General Settings` window and de-select the `Automatically Invoke Service Quick Find In Service Call` check box. See the following figure:

Figure 3.2 Automatic Quick Find in Service Call Switch

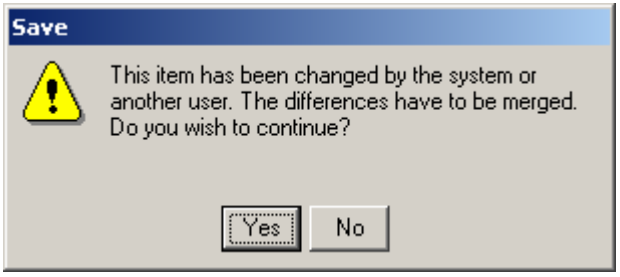


4 Automatic Merge of Concurrent Changes

In Service Pack 7 it is now possible to merge the changes made simultaneously by multiple users in one form.

For example, if two users are updating the same form and want to save their changes, a pop window (see the following figure) asks the user if they would like to merge the data:

Figure 4.1 Merge Request



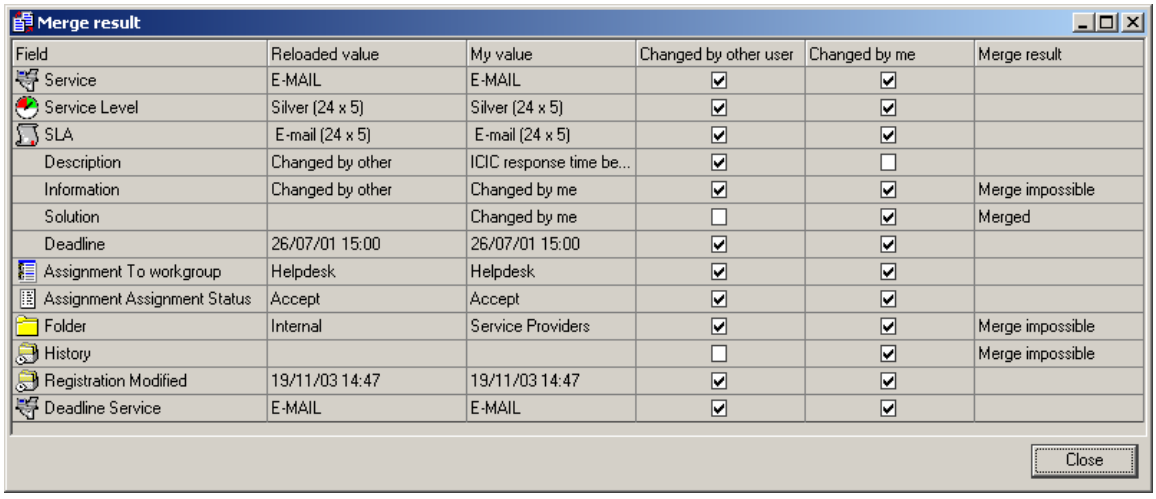
Clicking the `No` button cancels the save. If you select the `Yes` button, the fields are merged. The following table lists the possible options and outcomes:

Table 4.1 Merging of Fields

Action	Merging Outcome
User 2 changed a field and User 1 did not	User 2 details merged
User 1 changed a field and User 2 did not	User 1 details merged
User 2 and User 1 changed a field	No automatic merge possible
User 2 changed a field and User 1 made the same change	Merged (from either user)
User 2 changed a field that includes a list of other objects (for example, History lines)	No automatic merge possible

Once the merging takes place, the actions taken are presented in a list, see the following figure:

Figure 4.2 Merging Actions

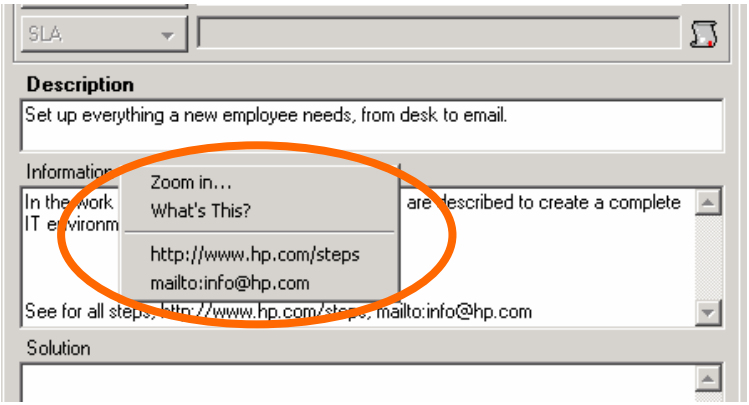


If `Merge Impossible` is listed for a field under `Merge result`, then the conflict can be resolved manually.

5 Hyperlink Support

You can now use hyper links in text fields. In Figure 5.1 Information Text box, you can see the hyperlinks when you right-click on the `Information` text box:

Figure 5.1 Information Text box



The pop up menu displays the hyperlinks as they appear in the `Information` text box.

Based upon both the formal definition of hyperlinks (go to the following URL: `http://www.ietf.org/rfc/rfc2396.txt`) and common practices within a Windows-based environment, the following strings are recognized as a hyperlink starter (the hyperlink schemes):

- `http://`
- `ftp://`
- `https://`
- `\\`[2]
- `file://`
- `mailto:`

Although the strings listed above are formally case-sensitive in Service Desk, these schemes are now recognized regardless of capitalization.

Certain special characters exist for hyperlinks. See the following table:

[2] This is Microsoft Windows specific (is a Windows synonym for `file://`); it is still supported here for convenience.

Table 5.1 Special Characters for Hyperlinks

Marks	- _ . ! ~ * ' ( )
Reserved characters	; / ? : @ & = + $,};
Escaped characters	%
Excluded characters	< > # { }| \ ^ [ ] ` “

Apart from excluded characters (they are filtered out), you are not allowed to use special characters to add a hyperlink.

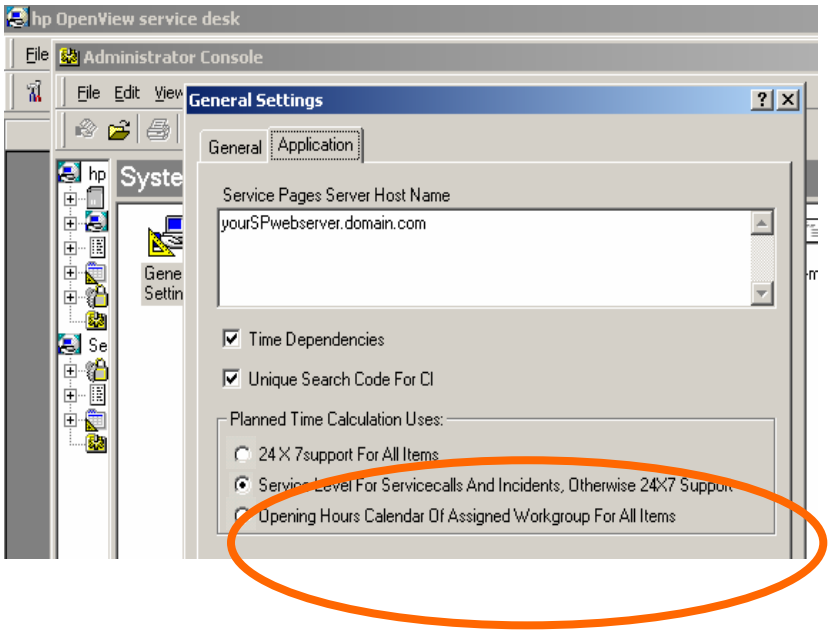
When a hyperlink is selected in the pop-up menu, the linked file is opened up in the application registered to open that type of file (for example, a HTML page is opened in a browser).

Empty hyperlinks are discarded from the list of hyperlinks.

6 Workflow Calculation per Workorder

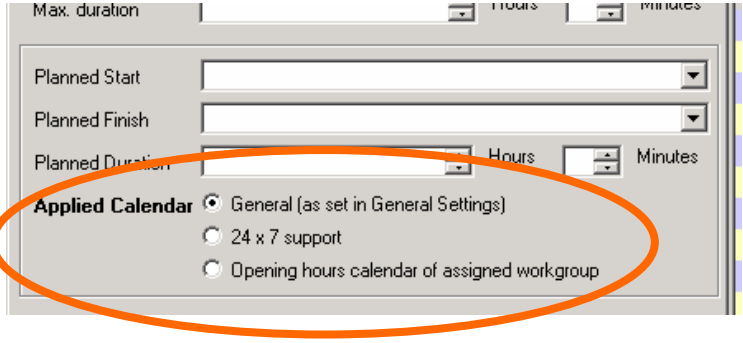
In Service Desk 4.5, determining how Workflow calculations are performed (the calculation of Planned Start, Planned Finish and resulting Planned Duration values) was performed at the (highest) level of the overall application. For example, Changes, Service Calls, and Workorders used the same calculation base (all performed in `General Settings`):

Figure 6.1 Workflow Calculations



`General Settings` functionality is now available at the Workorder level, so you can now perform a more detailed Workflow calculation for Workorders. See the figure below:

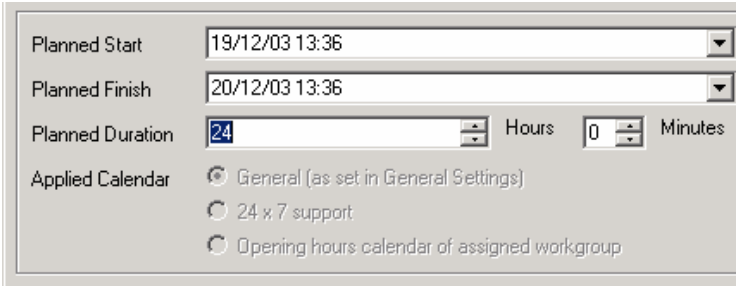
Figure 6.2 Applied Calendar Settings



By default, the first option is selected. If other options are selected, the calculation base setting for a Workflow is done on a per-Workorder basis.

When all the Workflow-related fields have been set, the new check boxes are disabled. See the figure below:

Figure 6.3 Check box Disabled



7 Minor Functionality Updates

This chapter describes some minor functionality updates.

Is Current Operator

This unary operator checks whether one of the operands is Person or Account (only available for those two field types). It can be used in three places:

- Advanced find: in a criterion under the `Advanced` tab
- View filter: in a criterion under the `Advanced` tab
- UI Rule: in a criterion in the condition

There is a related `set to current` operator in the Update Data action.

Variables

To increase the flexibility of conditions, you can now refer to the context of the evaluation for rule criteria, find criteria, and table filters. To do this, refer to variables within the value text field in these locations:

- Right-side of a criterion in a View filter and in Advanced Find
- Right-side of a criterion in a UI Rule
- Literal text expression in an Update Data action
- Command Exec action `Parameters` text box
- Smart Action `Parameters` text box
- Email action subject or message
- User Notification Message action message
- `Parameters` text box in the Update data from an external system action
- Inbound Email integration confirmation texts

The variables listed in the table below can be used at any place within the text box:

Table 7.1 All Text Variables

Name	Description	Client/ Server	Example
`$APPLICATIONSERVER$`	Current application server host name		`jupiter.neth.hp.com`
`$APPLICATIONSERVERPORT$`	The port of the current application server		`30999`
`$HOSTNAME$`	The host name of the current machine	X	`obelix.neth.hp.com`
`$IPADDRESS$`	The ip address of the current machine	X	`115.162.123.144`
`$ACCOUNTNAME$`	The Service Desk login name of the logged on user		`bobby`

Name	Description	Client/ Server	Example
$ITEMTYPE$	The unlocalized entity name of the current action (when available)		Servicecall
$DATAFOLDER$	The defined data folder	X	C:\Documents and Settings\bobby\Application Data\Hewlett-Packard\OpenView\Service Desk
$PRODUCTFOLDER$	The Service Desk installation folder	X	C:\Program Files\Hewlett-Packard\OpenView\service desk 4.5\client
$TIME$	The current time in UTC (Coordinated Universal Time), in the fixed dotted format: HH.MI.SS		14.21.02
$DATE$	The current date in UTC in the fixed dotted format: YYYY.MM.DD		2003.12.21
$DATETIME$	The current date and the current time in UTC, in the fixed dotted format: YYYY.MM.DD.HH.MI.SS		2003.12.21.14.21.02
DD	Current day of the month in UTC		21
MM	Current month in UTC		12
$YYYY$	The current year in UTC		2003
HH	The hour number (24h) in UTC		14
MI	The minute number in UTC		21

Name	Description	Client/ Server	Example
SS	The number of seconds in UTC		02
D	The day of the week (1–7, Sun–Sat)		3
WW	The week of the year (definition of Java.Calendar. WEEK_OF_YEAR)		13
Z	The name of the time zone for the current session		CET
ZZ	The time zone offset of the current session		+01:00
%<JAVA_VARIABLE>%	All Java properties of the class System. This is dependent on the used Java version. See appendices for an example[3].	X	%java.version% => 1.1.4
%<OSVARIABLE>%	All available OS variables on the platform. It is the responsibility of the customer what variables can be used. See appendices for an example[3].	X	%OS% => Windows_NT %USERNAME% => bobby

[3] If the variable (either Java or OS) can not be found, the variable is just left unchanged in the text.

Buttons in Forms

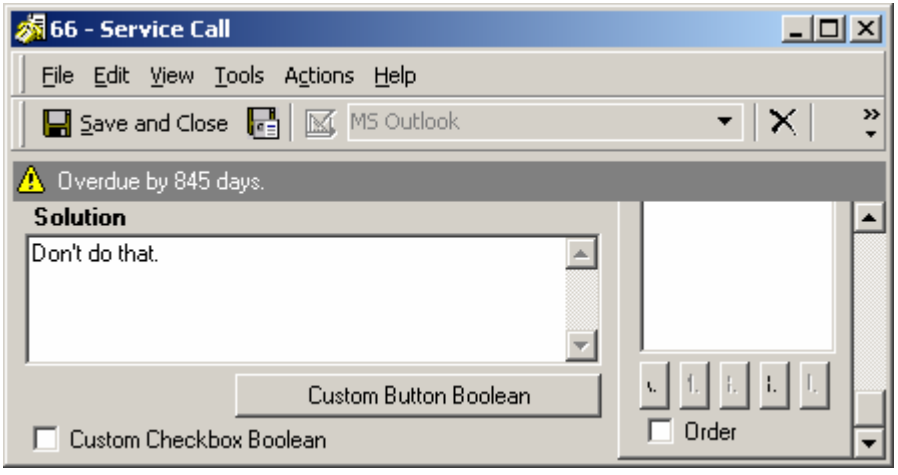
You can now add buttons to forms and these buttons can be used to trigger UI Rules (which in turn execute actions). These buttons are a special display form of `Boolean` fields. An UI rule is created for each button that triggers once the state of the button changes value to `On`. The first action to execute is always to reset the status of the button to `Off` so that the button is not grayed out[4].

For example, you can use a button on a form to trigger a Data Update action to retrieve extra data, or a Command Exec action that starts a telnet session to an affected CI. Also, data from related forms can be copied into the current form.

Adding a Button to a form

To place a button on a form, a Boolean custom field has to be activated, given a usable name, and set to the display format `Button`. After that, the button can be added to a form by dragging the custom field in the form designer. See the figure below for some example fields on a form:

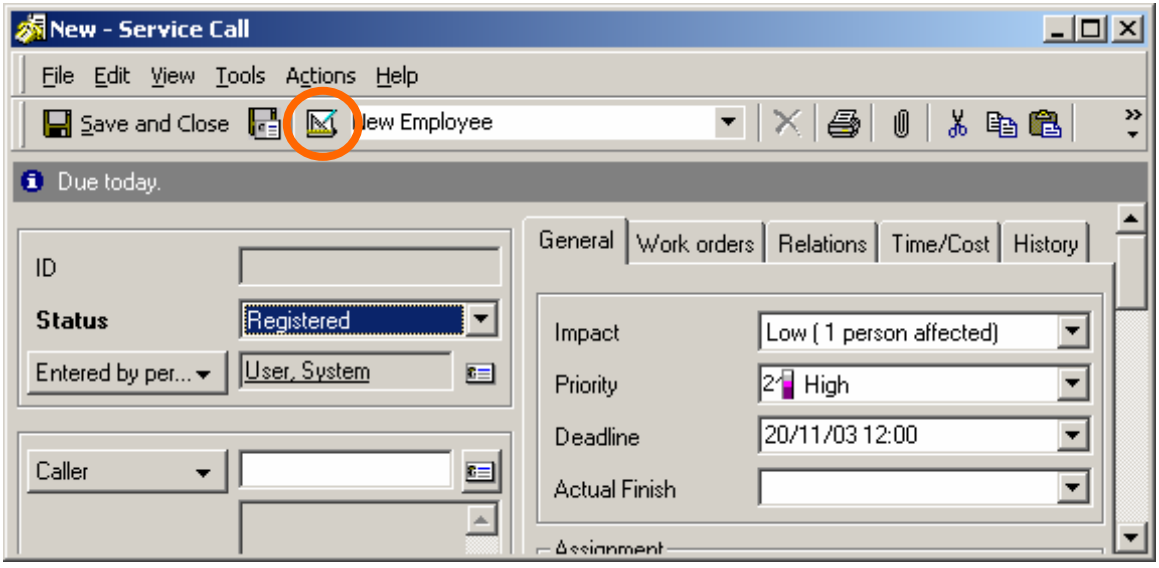
Figure 71 Button Boolean and Normal Boolean



Choose Template Dialog

In Service Desk you can use templates to partially fill a form when creating new items in the database by selecting a template from a drop down list (see the figure below). However, when there is a large number of templates this menu is difficult to use and a more structured approach is needed. In the this release, there is an extra button next to the drop-down menu that opens the `Choose Template` dialog. This makes it easier to choose the correct template.

Figure 7.2 Creating a New Service Call - Choosing a Template



[4] Immediately resetting the status of the button to `Off` right after it has been clicked, will make this button click-action not appear in the History Lines. This is default audit behavior.

Deadline Service and Service Level Weight Fields

Two new fields have been added: `Deadline Service` and `Service Level Weight`.

Deadline Service Field

This field is used by Incidents and Service calls. These objects both have a deadline calculated from a Service and a Configuration Item. This field indicates the service associated with the SLA whose deadline will be violated first by the Incident or Service call. If the deadline is empty or entered manually (in other words, it is not a result of a deadline calculation), the deadline field is empty.

The Deadline Service Field field is read-only since it is always calculated by Service Desk. It is added to the form by the Service Desk administrator.

Service Level Weight Field

The Service Level Weight is a property of the service level. The weight of a service level encodes the importance of that service level. It is used to sort the Service Impact View entries when two services have identical deadlines. Therefore a Gold SLA can appear before a Silver SLA.

The Service Level Weight field is a numeric (integer) field and must be unique. Multiple weights can be empty but non-empty weights have to be different. The Service Level Weight field can only be added to the form by the Service Desk administrator.

The Service Level forms can be edited by selecting:

```
Administrator Console>hp OpenView
servicedesk>Security>Access>Role>Access to all
forms>Service Level>Advanced>Forms
```

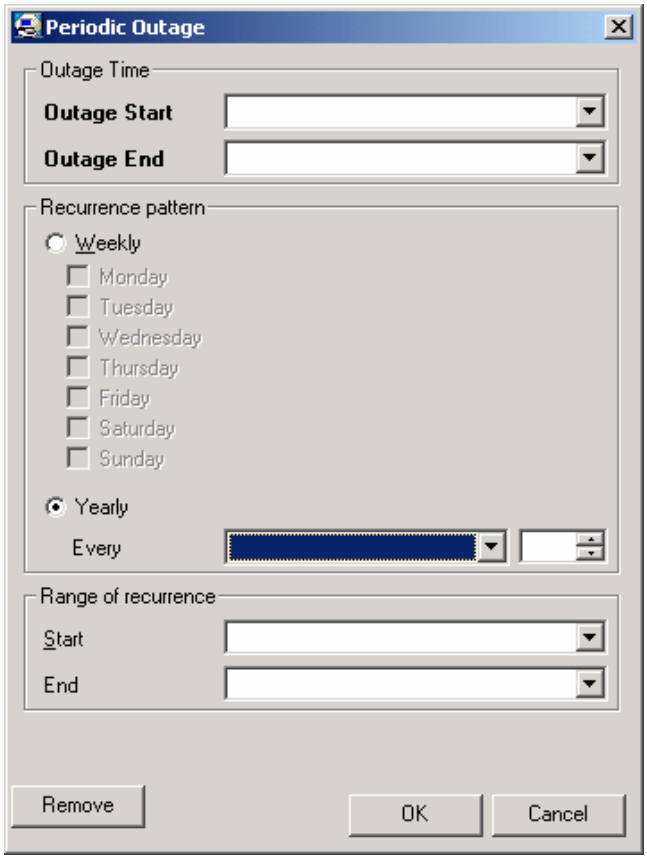
8 OVO for UNIX Integration

This chapter describes some specific tasks that have been updated or added to the OVO for Unix Integration.

Defining a Periodic Outage for a Configuration Item

The section “Suppressing Messages in Service Desk” of the *OpenView Operations Integration Administrator’s Guide* provides a general overview of how to configure a periodic outage. This section provides detailed steps how to use this functionality:

Figure 8.1 Periodic Outage Form



The fields `Outage Start` and `Outage End` (the two fields defining the `Outage Time`) are mandatory.

NOTE Times entered in Service Desk are *always* in the time zone of the Service Desk client. When defining outage on an OVO management server in a different time zone, you should:

- Enter begin and end times as they are in the time zone of the client during the outage,
- And *not* begin and end times as they would appear in the time zone of the OVO management server.

A periodic outage can occur every week or every year, not monthly. The `Recurrence Range` allows you to define a date range during which the recurrence is active. When the `Recurrence Pattern` is set to `Yearly` and no value is specified, the `Start` and `End` of the recurrence range is subject to interpretation:

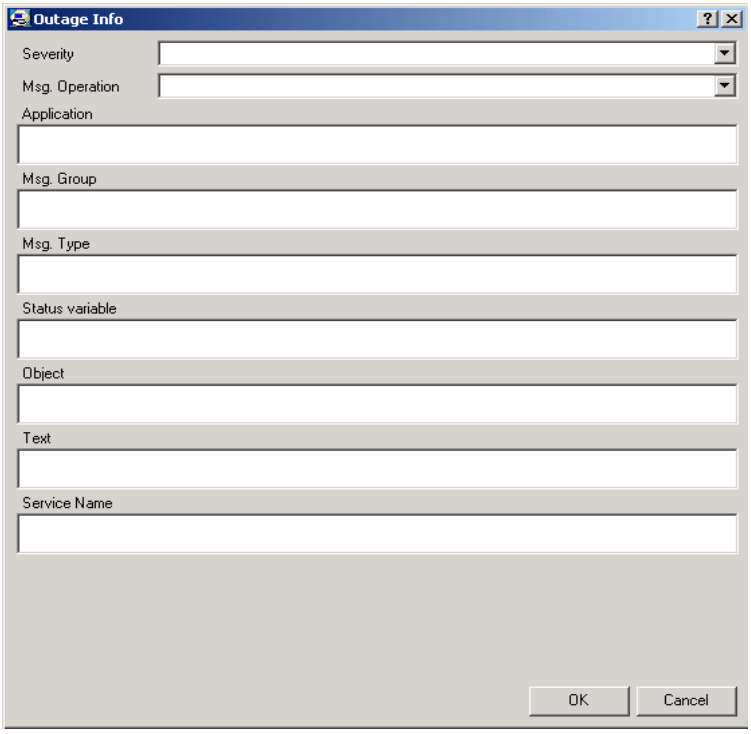
- A missing `Range of Recurrence Start` date is interpreted to mean the date specified in the `Every` field. A missing `Range of Recurrence End` is interpreted as the `Range of Recurrence Start` date plus 25 years.
- If both the `Range of Recurrence Start` and `End` fields are not defined, the yearly outage definition is interpreted as applying to every year (past, present, and future).

A definition of periodic outage does not completely suppress all messages coming from OVO for UNIX. Next to the outage definition, OVO also needs to know which messages are subject to the outage. This is defined in the `Outage Info` form, which is explained in the next section.

Specifying Messages Subject to an Outage Definition

This section describes how to specify what OVO for UNIX alerts are subject to an outage definition. This applies to both occasional as well as periodic outages.

Figure 8.2 Outage Info Form



The above screenshot shows all available fields for Outage Info forms. In Service Pack 7, this new form definition has been added for Outage Info that includes them all (not necessarily in the order shown above). This form is called “OutageInfoSP7”.

The `Outage Info` Form is located under the `SN and Outage` tab for Configuration Items and Work Orders. The fields on displayed form correspond to attributes in OVO for UNIX alerts. OVO for UNIX matches

the OVO alerts subject to the outage definition based on the values entered in these fields.

The `Msg. Operation` field contains important information. The following table outlines how the OVO management server handles alerts based on the value of the `Msg. Operation` field:

Table 8.1 Msg. Operation

Parameter	Description
INSERVICE	If the message condition matches and the time Template condition does *not* match, VPO sends messages to the Pending Messages Browser until the unbuffer time condition is matched or until the message is unbuffered manually.
LOGONLY	Sends matching messages to the history browser.
SUPPRESS	Deletes messages. Message-related actions triggered by the VPO management server are *not* started if the `SUPPRESS` option is defined.

More information on the `Outage Info` form fields in can be found in the OVO Administrator Reference Vol. 1 in the chapter “Configuring VPO”, in the section “Keywords for Flexible Management Templates”. The message properties `Object`, `Text`, `Service Name` and `Msg. Type` are new to SD 4.5 Service Pack 7.

Defining Database Rules to Forward Outage Definitions

The section “Configure Service Desk for Outage Planning” in the *OpenView Operations Integration Administrator’s Guide* describes how to set up the database rules used to forward changes in the outage planning to OVO for UNIX. In Service Desk 4.5 Service Pack 7, the implementation of those database rules has changed.

A database update file is provided with SD 4.5 Service Pack 7. It contains the database rules used with outage planning with the changes described in this section.

Do the following to import these database rules into your default Service Desk database:

For UNIX platform:

```
$ cd /opt/OV/sd/server/bin
$ ./sd_dbconfwizard -CREATEDB
"/opt/OV/sd/server/repo/tables/upgrade/dbrules-
mkoutage.dsd;/opt/OV/sd/server/repo/tables/upgrade/
dbrules-mkoutage.rsd"
```

For Microsoft Windows platform:

```
> cd C:\Program Files\Hewlett-
Packard\OpenView\service desk 4.5\server\bin
> sd_dbconfwizard.bat -CREATEDB "C:\Program
Files\Hewlett-Packard\OpenView\service desk
4.5\server\repo\tables\upgrade\dbrules-
mkoutage.dsd;C:\Program Files\Hewlett-
Packard\OpenView\service desk
4.5\server\repo\tables\upgrade\dbrules-
mkoutage.rsd"
```

Before importing this file, remove the existing database rules for outage planning from your Service Desk configuration. You are advised to do this *only* when performing *new* installations of the OVO integration. For existing installations, it is easier to manually modify the rules already present in the administrator console, as you may have made other changes to the database rules in addition to what is described here in order to tailor the database rules to your environment.

The following two sections detail the setup of the database rules for forwarding outage definitions. These definitions overrule the ones by the same name in the OpenView Operations Integration Administrator's guide.

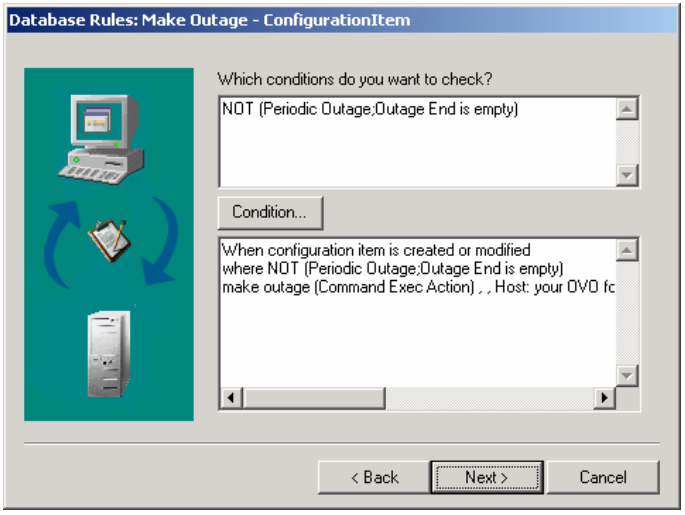
Creating a Database Rule for Periodic Outage Information

The section "Creating a database rule for Periodic Outage Information" in the chapter "OVO UNIX and Service Desk Configuration" of the *OpenView Operations Integration Administrator's Guide* describes how to set up a database rule to send periodic outage information to OVO for UNIX. The information in this section no longer applies once you install this service pack. Please use the following instructions to set up this database rule:

1. Open the Database Rule Wizard: go to the `Administrator Console`, select `Business Logic`, `Database Rules`, `Configuration Item`, and select the `mkoutage` rule if it exists in the Service Desk database; otherwise a new `mkoutage` rule.

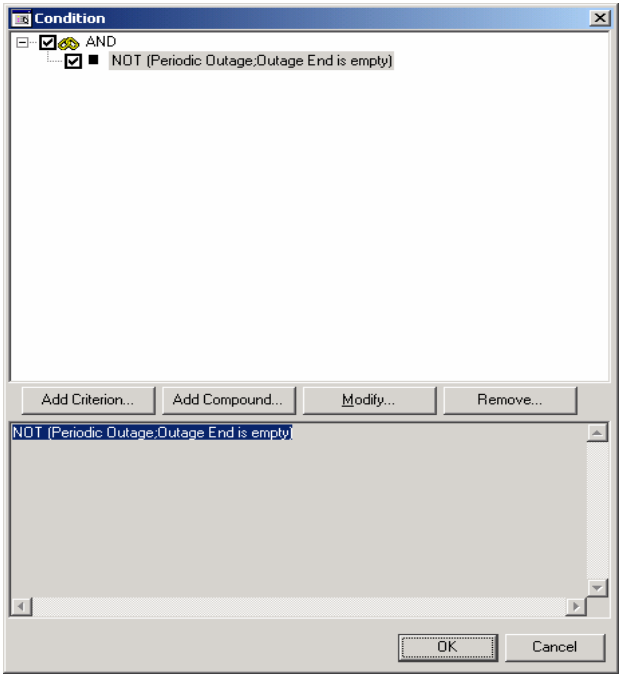
2. Select: When `configuration item is created or modified`, then press `Next`. The following window is displayed:

Figure 8.3 Defining a Condition



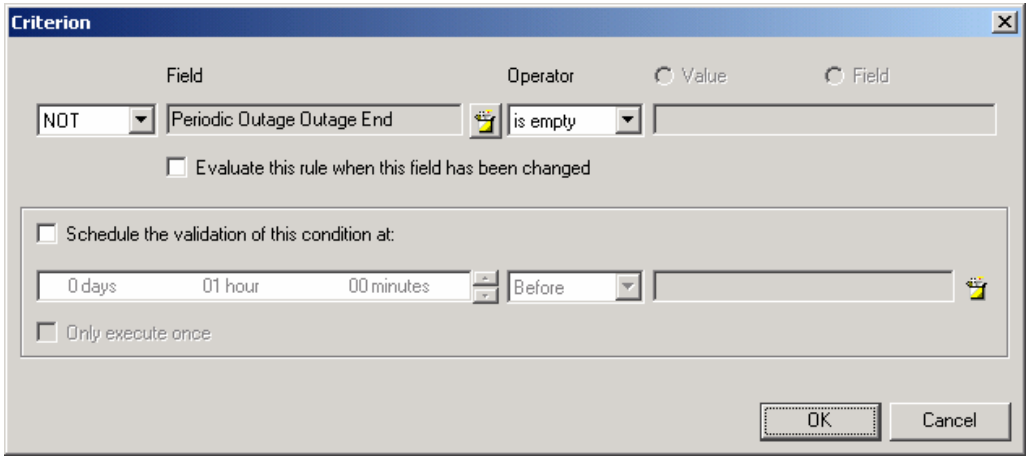
3. Press the `Condition` button to set up a new condition, the following window opens:

Figure 8.4 Condition Window



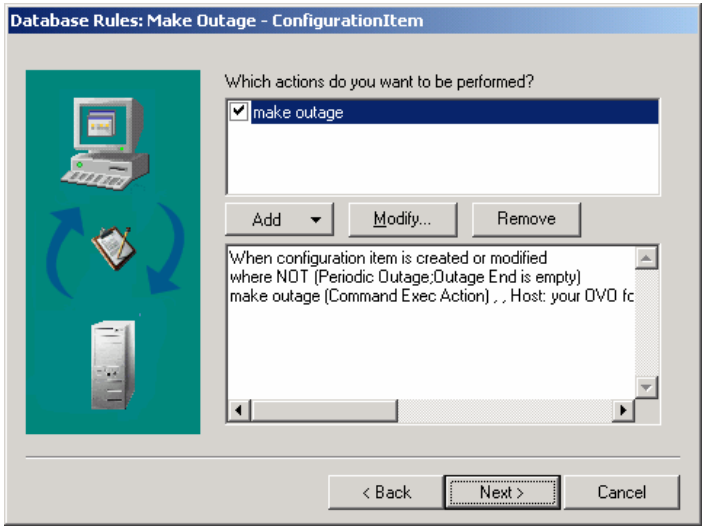
4. Press the `Add Criterion` button to define a criterion that limits the number of times the database rule is triggered, see the figure below:

Figure 8.5 Criterion Window



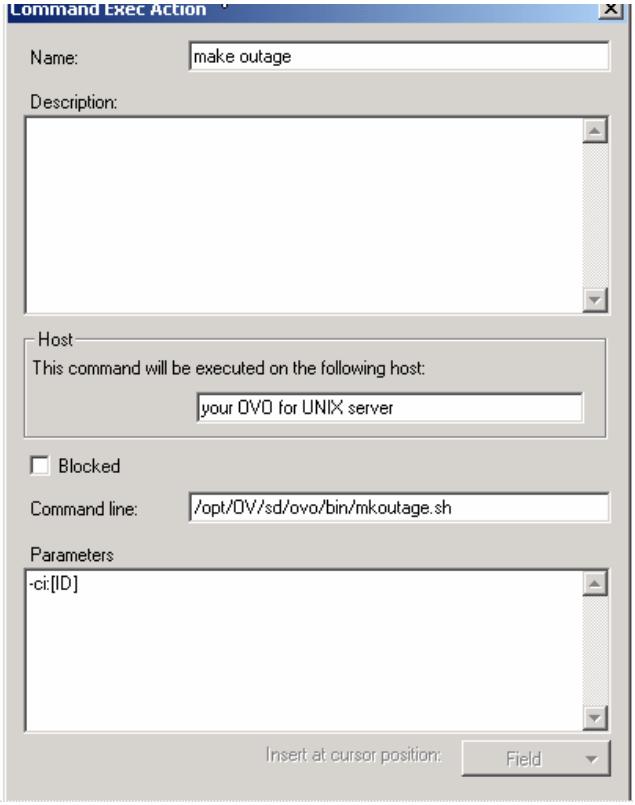
5. Define the criterion according to the definition seen in the screenshot above, then press `OK`. Follow the onscreen instructions. You then return to the screenshot seen before that one, on which you will have to select `Next` to go forward to the next step of the database rule setup, which is seen below:

Figure 8.6 Make Outage Configuration Item



Press the Add button to define a database rule action, see the figure below:

Figure 8.7 Command Exec Action Form

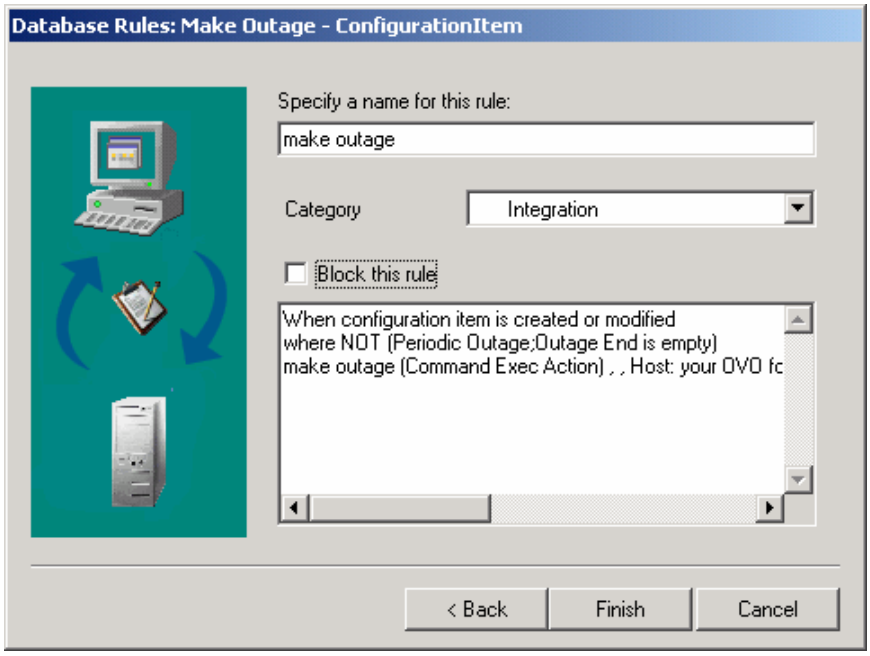


6. Create a database rule action according to the settings you see above. In the `Host` field, enter the fully qualified name of the OVO for UNIX management server on which the integration of Service Desk with OVO for UNIX is installed, and press the `OK` button when you are finished. This will return you to the previous window, in which you have to press the `Next` button to proceed to the final screen of the database rule wizard.

7. In this final window of the database rule wizard, you should select a name for the rule and a category before you can press `Finish`. Do not

select the `block this rule` check box, otherwise the rule will remain inactive. See the figure below:

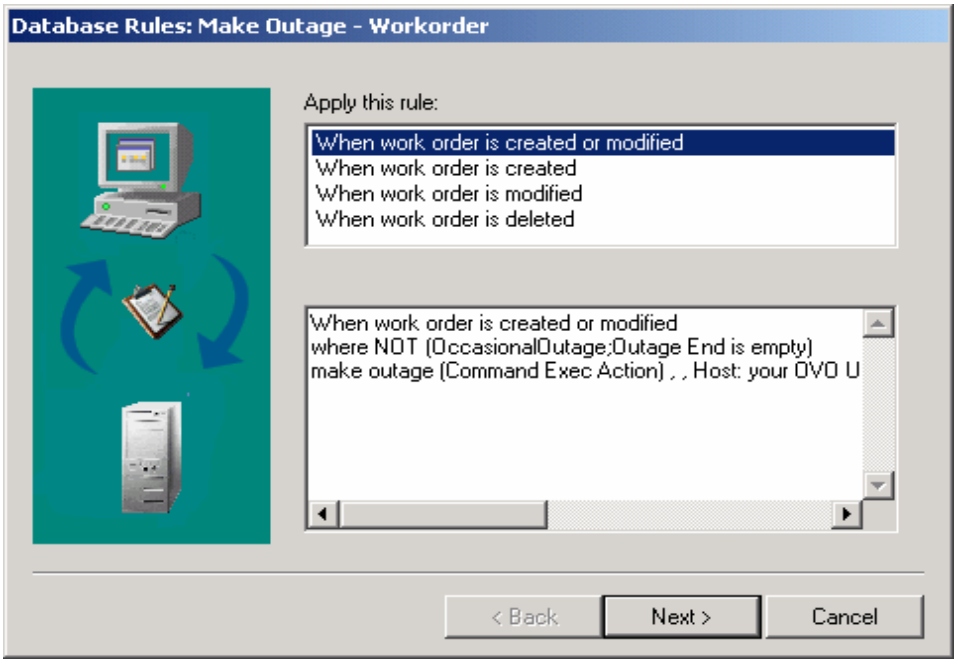
Figure 8.8 Final Database Wizard



Creating a Database Rule for Occasional Outage Information

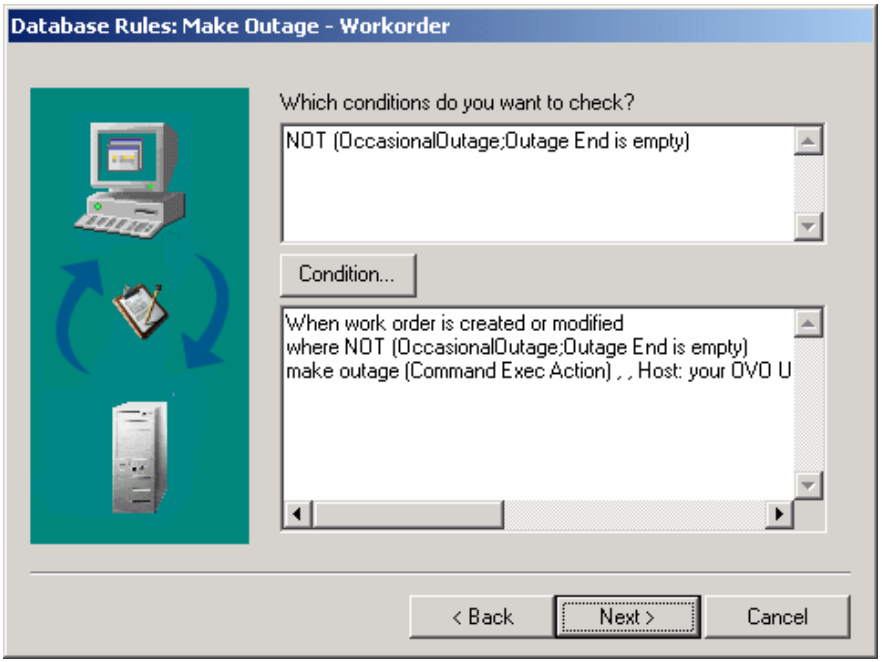
1. Open the Database Rule Wizard: go to the Administrator Console, select `Business Logic`, `Database Rules`, `Work Order`, and select the `mkoutage` rule if your Service Desk database already contains one or create a new one in case it does not.

Figure 8.9 Applying the Rule – Workorder



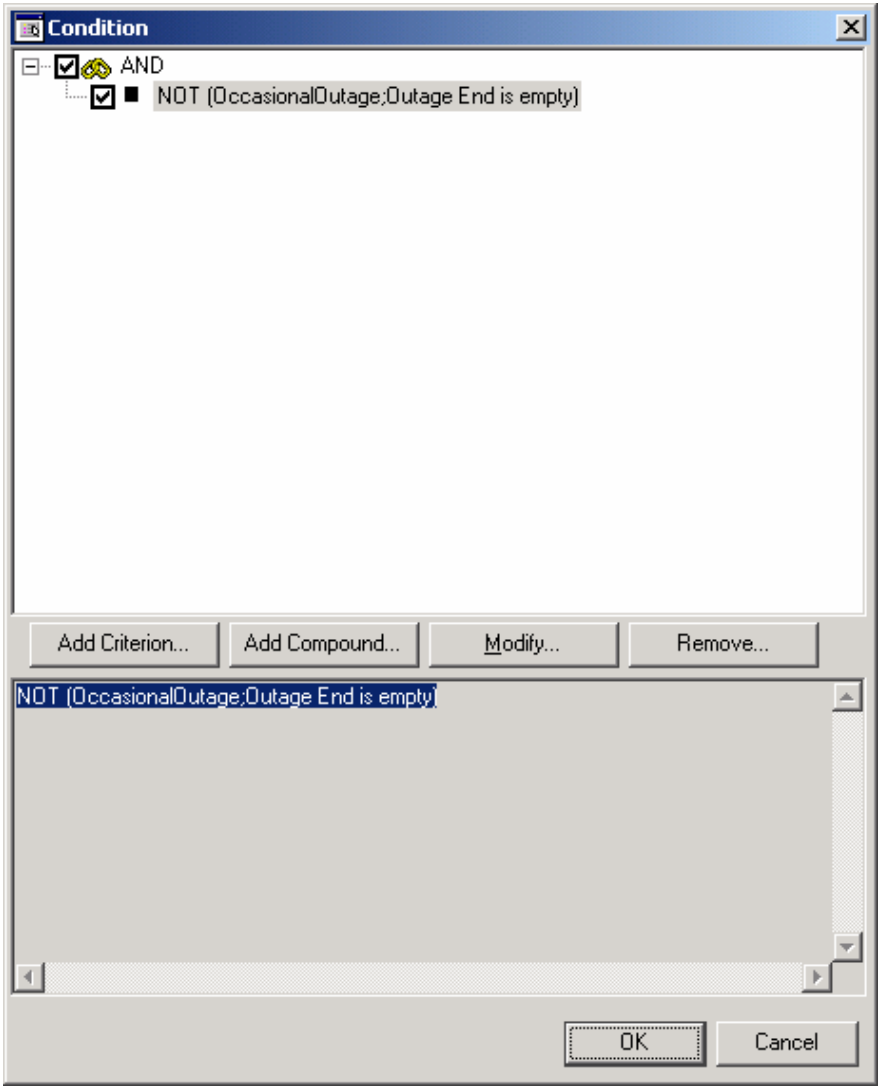
2. Select: `When work order is created or modified`, then press `Next`.

Figure 8.10 What Conditions to Check



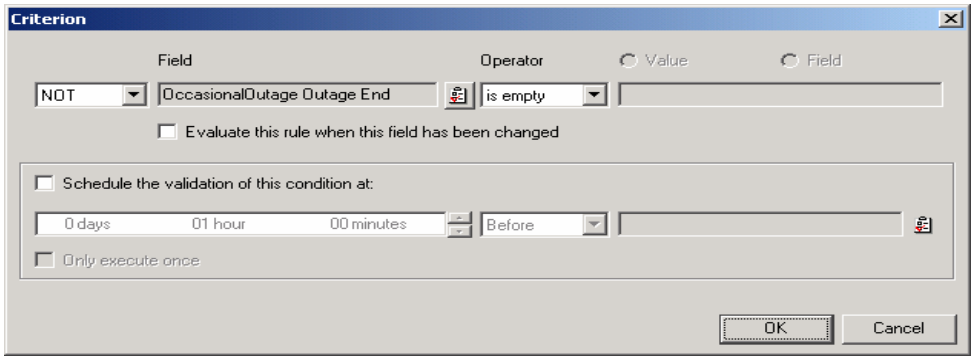
3. Press the `Condition` button to set up a new condition, the following window is opens:

Figure 8.11 Condition Window



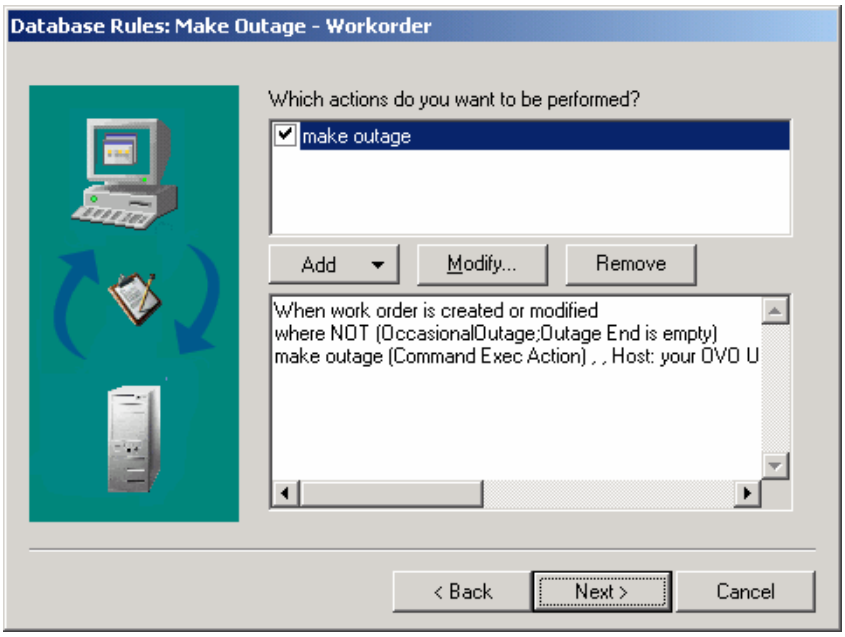
4. Press the `add criterion` button to define a criterion that limits the number of times the database rule is triggered. This opens a window similar to the following:

Figure 8.12 Criterion Definition Window



5. Define the criterion according to the definition seen in the screenshot above, then press OK. Follow the onscreen instructions until you get to the following window:

Figure 8.13 Actions to Perform



6. Press the Add button to define a database rule action. This will open the following window:

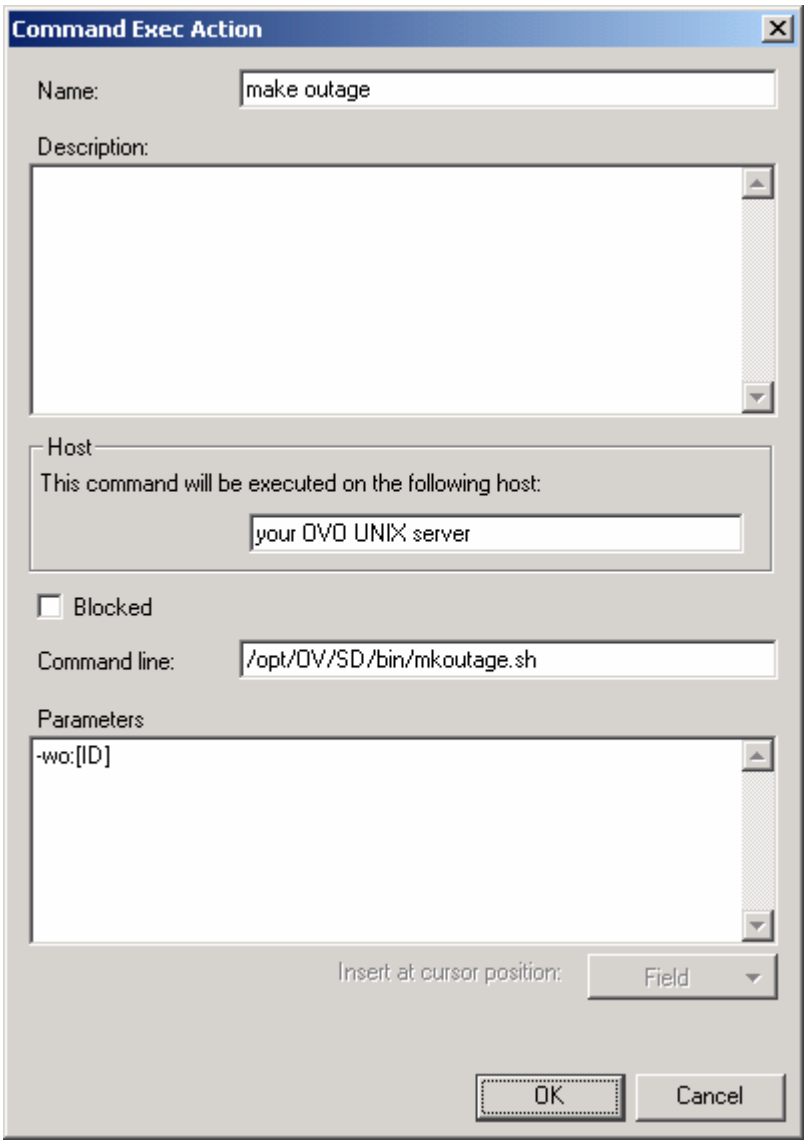
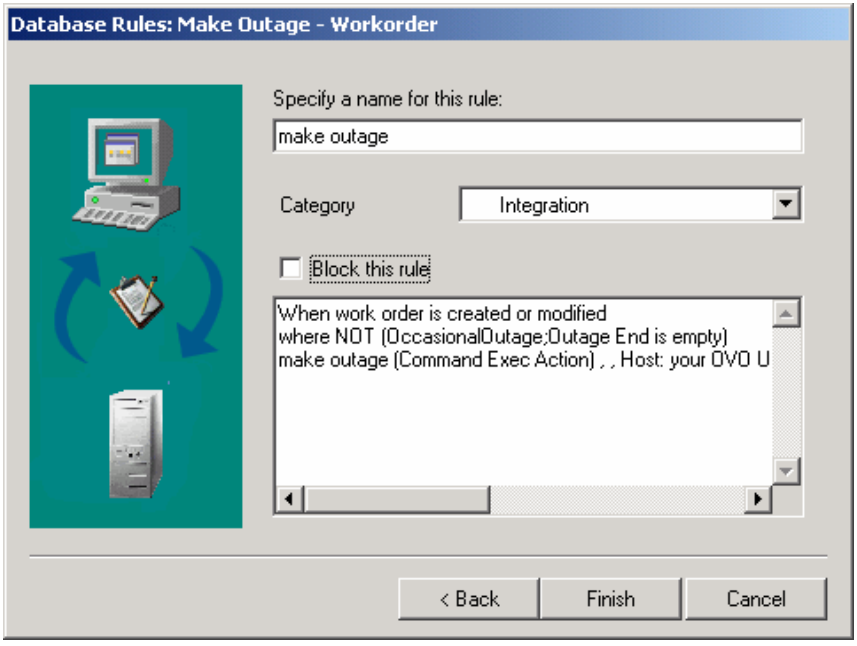


Figure 8.14 Command Exec Action

7. Create a database rule action according to the settings you see above. In the `Host` field, enter the fully qualified name of the OVO for UNIX management server on which the integration of Service Desk with OVO for UNIX is installed, and press the `OK` button when you are

finished. This returns you to the previous window, in which you will have to press the `Next` button to proceed to the final screen of the database rule wizard.

Figure 8.15 Specifying a Rule Name



In this final window of the database rule wizard, enter a name for the rule and a category, then click the `Finish` button. Do not select the `block this rule` check box, otherwise the rule will remain inactive.

Importing Operational Level Services into Service Desk

The section “Importing Services from OpenView Operations for UNIX” in the chapter “OVO UNIX and Service Desk Configuration” of the OpenView Operations Integration Administrator’s guide describes how to create configuration items in Service Desk to represent services in OVO for UNIX. For an overview of the steps to perform this operation, refer to that section of the OVO Integration guide.

NOTE It is only possible to extract services from OVO for UNIX when Service Navigator is installed.

As explained in the OVO Integration guide, you can only extract the data from the file produced by Service Navigator (on top of OVO for UNIX) through an ODBC source on a Service Desk client using the extractor file `ovoservices.ini`. It is not possible to perform this task using the extractor file on UNIX unless you have a JDBC text driver and reconfigure the extractor file to use this driver. More information on configuring the extractor for JDBC can be found in the section Defining a JDBC link in this document.

Using the new services extraction script

Service Desk 4.5 Service Pack 8 contains a new script called ovo_services.pl that replaces the following files in the OVO for UNIX integration:

- `ovo_services`
- `ovo_services.awk`
- `ovo_services_relations.awk`

The ovo_services.pl script improves on the scripts above in that it keeps track of all the search codes for services it generates in one run and tries to prevent that two distinct services are assigned the same search code.

To extract a list of services from Service Navigator in the same way as described under “Importing services from OpenView Operations for UNIX” in the OpenView Operations Integration Administration guide,

run `ovo_services.pl` without any arguments. This will create a file named ovo_services.txt, same as the ovo_services script used to do.

Before using the script, verify that the shebang is valid and edit it if necessary. The shebang is the first line of the script that starts with the characters "#!" (without the quotes), followed by the path to a Perl interpreter (version 5.6 or higher) on your system. The default is `/usr/bin/perl`, but it may be different in your environment.

Changing the import mapping using a database update file

In addition to the new script, Service Desk 4.5 Service Pack 8 contains an enhancement to the import mapping "ovounixservices", which is used to load the services into Service Desk after extraction from Service Navigator. The import mapping now defines the relationships between the resulting configuration items as defined by Service Navigator. A database update file that contains the changes described in this section is provided with SD 4.5 Service Pack 8.

Before importing this file, you must remove the following items from the administrator console of your Service Desk configuration:

- `Data/Templates/Configuration Item/OVOSERVICE`
- `Data/Data Exchange/Data Exchange Task/ovounixservices`
- `Data/Data Exchange/Import mapping/ovounixservices`

Before removing the existing items, verify that you have not made any changes that would be undone by updating.

You are advised to do the database update *only* when performing *new* installations of the OVO integration. For existing installations, it is easier to manually modify the rules already present in the administrator console, as you may have made other changes to the database rules in addition to what is described here in order to tailor the database rules to your environment.

Do the following to import the changes into your default Service Desk database:

For UNIX platform:

```
$ cd /opt/OV/sd/server/bin
$ ./sd_dbconfwizard -CREATEDB
"/opt/OV/sd/server/repo/tables/upgrade/dbrules-
ovoservices.dsd;/opt/OV/sd/server/repo/tables/upgrade/ovo
services.rsd"
```

For Microsoft Windows platform:

```
> cd C:\Program Files\Hewlett-Packard\OpenView\service
desk 4.5\server\bin
> sd_dbconfwizard.bat -CREATEDB "C:\Program
Files\Hewlett-Packard\OpenView\service desk
4.5\server\repo\tables\upgrade\ovoservices.dsd;C:\Program
Files\Hewlett-Packard\OpenView\service desk
4.5\server\repo\tables\upgrade\ovoservices.rsd"
```

Manually changing the ovounixservices Import Mapping

To change the ovounixservices import mapping in the administrator console instead of through the database update file, do the following:

Figure 8.16 Changes to the `ovounixservices` Import Mapping: Services

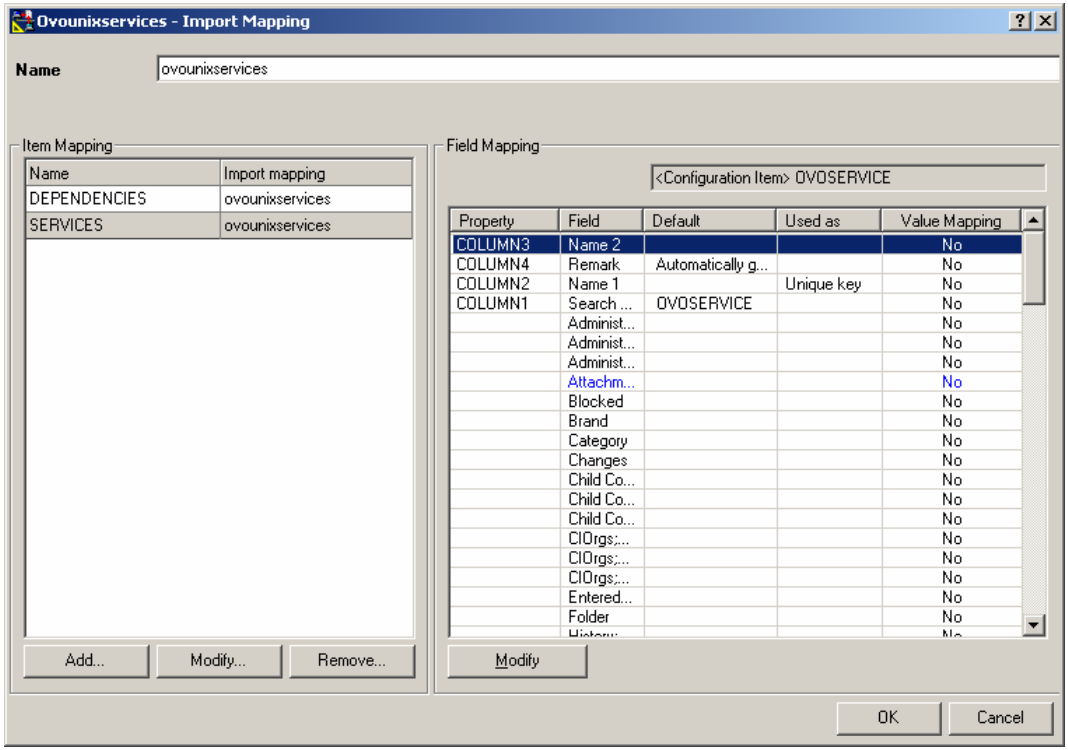
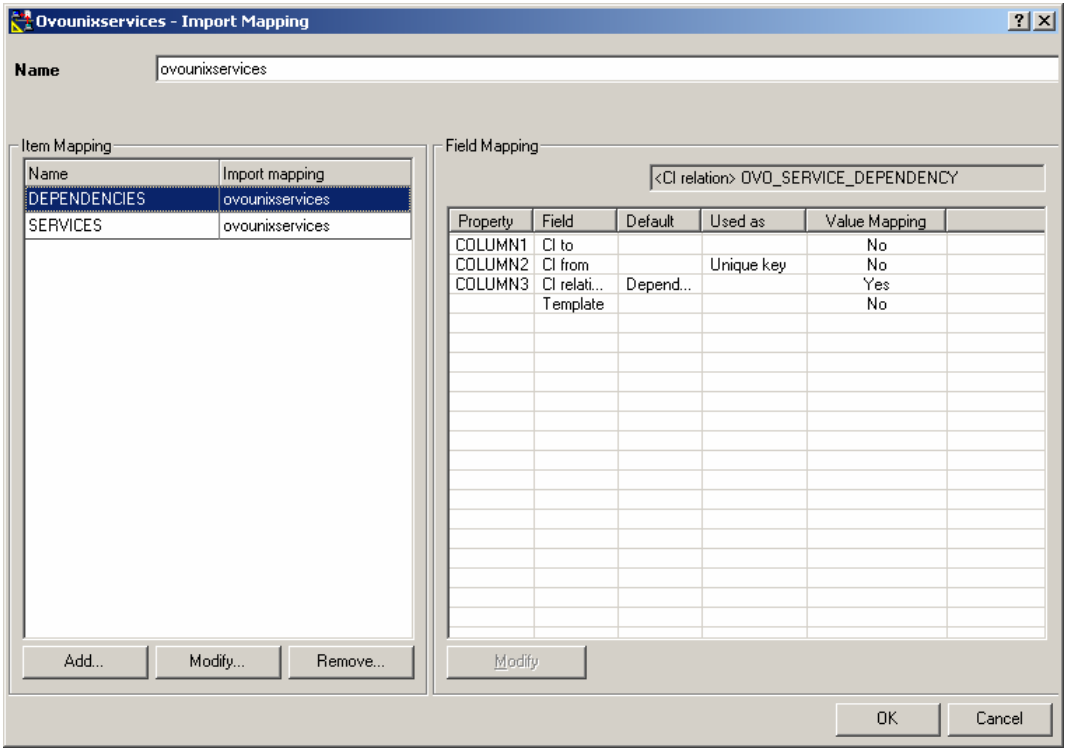


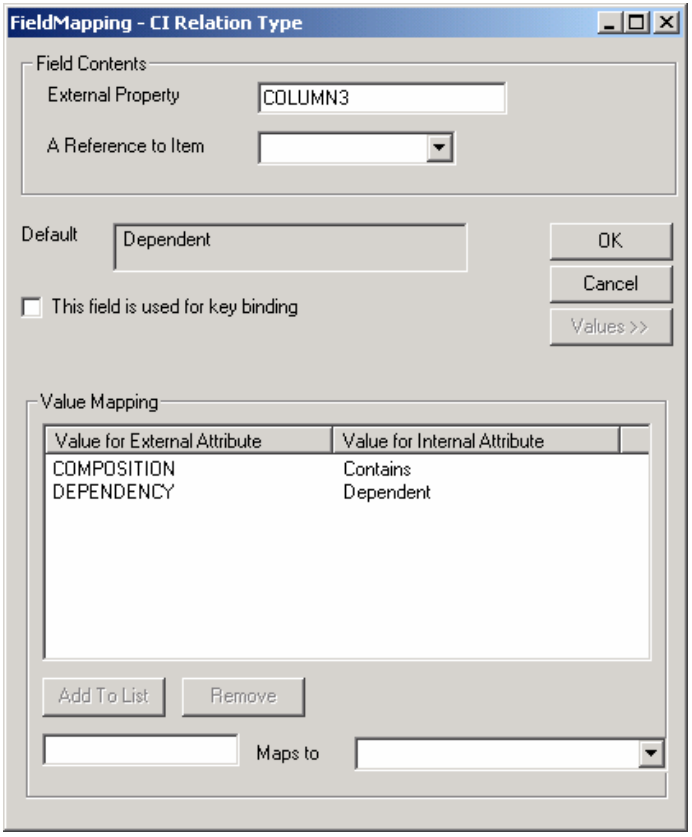
Figure 8.17 Changes to the `ovounixservices` Import Mapping: Dependencies



To change an existing `ovounixservices` import mapping to match the example seen above, press the `Add` button under the left pane. Create a new item mapping called `DEPENDENCIES`, referencing the item `CI relation` and using template `OVOSERVICE`. In the right pane, map properties according to the following table:

Property	Field	Part of Primary Key	Value Mapping
COLUMN1	CI to	No	No
COLUMN2	CI from	Yes	No
COLUMN3	CI from	No	Yes

Figure 8.188 Value Mapping of COLUMN3



The setup of the `ovounixservices` Data Exchange task does not change. However, the Data Exchange task must be redefined when a

completely new import mapping is introduced, as is effectively done with the database update file. This is because Data Exchange tasks base their relation to an import mapping on properties not visible to the end-user. This way, a Data Exchange Task can maintain a relation with an import mapping even when the import mapping's name changes.

Exporting Nodes for Import into a UNIX SD Application Server

The section “Importing nodes into Service Desk” in the chapter “OVO UNIX and Service Desk Configuration” of the OpenView Operations Integration Administrator’s guide describes how to create configuration items in Service Desk for all nodes managed by OVO for UNIX. Creating these Configuration Items is done through the Data Exchange functionality of Service Desk.

The examples given in the Integration guide are based on running Data Exchange on a Windows system. When your Service Desk application server is running on UNIX, you might prefer to use the command line versions of the Data Exchange program `sd_export` and `sd_import` directly instead of having to perform these actions from a Windows client.

In order to avoid the problem of extracting the data from OVO for UNIX (which relies on the existence of an ODBC data source), and possibility of your UNIX system not supporting this type of connection, you can manually configure `sd_export` to use JDBC instead.

Service Desk 4.5 Service Pack 7 contains an example of the `sd_export` configuration file (also referred to as extractor file in the Service Desk manuals). It is located in `<SD Product Path\client\data exchange\config>` on Service Desk client installations and in `<SD Product Path/server/data exchange/config>` on application server installations.

The file `jdbc_oracle_ovounixci.ini` configures `sd_export` to connect to the OVO for UNIX Oracle database using a JDBC connection. If your SD application server is on the same host as your OVO management server, the connection works first time; the `opc_op` account in OVO for UNIX has the default password, and your OVO for UNIX Oracle database instance has the default name `openview`.

In order to adapt the extractor configuration file for use in your particular environment, edit `jdbc_oracle_ovounixci.ini`. To change the user name and password used to connect to Oracle, change the `USR` and `PWD` parameters in the `JDBC` section. To change the Oracle connection information, change the three values between colons after the @ character in the `URL` parameter in the `JDBC` section. The first value is the Oracle host name, the second is the port that the Oracle TNS listener is configured to use on this host, the third is the instance name of the OVO for UNIX Oracle database.

The configuration file `jdbc_oracle_ovounixci.ini` contains comments that elaborate on the above instructions and should help you complete the configuration of the UNIX version of `sd_export` to use JDBC for the extraction of nodes from OVO for UNIX.

NOTE Next to the import of nodes, the integration of Service Desk with OVO for UNIX has an option to import services. The integrations manual describes this as a task to perform on a SD client in part for the `sd_export` step. The reason this has not been changed to work with JDBC just like the import of nodes is that `sd_export` needs to use a text driver in this context. Currently, a JDBC text driver cannot be provided with Service Desk. For more information see the section “Finding Suitable JDBC Drivers”.

Enhancements to the Service Desk message source templates in OVO

The integration of Service Desk with OVO contains OVO message source templates that configure OVO to interact with Service Desk in a number of ways. Among other things, these templates configure OVO to monitor log files generated by Service Desk. In Service Pack 7, the monitoring templates have been updated to work out of the box on new installations where OVO for UNIX, the Service Desk application server and the integration between the two are all installed on the same machine.

In case the assumptions documented above do not hold true for your environment, you will need to modify the message source templates for monitoring Service Desk log files prior to deploying them. The section "Deploying the message source templates" in the chapter "OpenView Operations for UNIX" in the Service Desk – OVO for UNIX integrations application guide identifies the names and locations of the log file monitoring templates in OVO:

Table 8.2 OVO Message Source Templates to monitor SD logs

Message Source Template Name	Function
SD_VP	Monitors event forwarding log on OVO management server
SD_VP_ACK	Monitors SD agent log on OVO management server (for acknowledgements returning from SD to OVO)
SD_VP_APP_SERVER_LOG	Monitors SD application server logfile

The updated message source templates need not be re-deployed or changed in any other way on existing installations of the integration between Service Desk and OVO for UNIX, as the changes to the templates most likely have already been made to get Service Desk log file monitoring part of the integration working.

9 Data Exchange

This chapter describes the changes to the data exchange functionality within Service Desk for this service pack.

Defining a JDBC link

The section “Defining a JDBC link” in the chapter “Exporting Data” of the Data Exchange Administrator’s guide contains a brief description on how to configure an extractor configuration file to use a JDBC connection. This section provides some more detail on the setup process.

NOTE You need to have Java experience to implement JDBC drivers.

Finding Suitable JDBC Drivers

The first step in configuring an extractor configuration file to us a JDBC connection, is to locate a suitable JDBC driver the source you want to connect to. If your database product is not delivered with a JDBC driver and your database vendor can not provide you with a driver, there is often a third-party implementation available, either commercially or through open source.

A JDBC driver is only suitable for use with `sd_export` if it is functionally complete (you may need to test this with `sd_export`). For example, the open source JDBC Text driver `csvjdbc` is currently not ready for use with Service Desk. Any attempt to use this driver with `sd_export` will cause the program to fail with a Java error message about the implementation of the `prepareStatement` method.

Service Desk includes JDBC drivers for two of the most common database products: Oracle and SQLServer. You can find examples of their use in the JDBC extractor example in the Data Exchange Administrator’s guide.

Using JDBC Drivers

The Data Exchange documentation advises you to place the JDBC driver in your Java CLASSPATH environment variable. Alternatively, sd_export can automatically load the JDBC driver if you place the .jar file in which the driver is implemented in the Service Desk lib directory. The lib directory is located under the home directory of the Service Desk client and/or the Service Desk server. You can use the environment variables SD_CLIENTHOME and SD_SERVERHOME to find the SD client and SD server home directories. The sd_export program is supplied with the client and the application server, so it is important to know what application you are using. If you are unsure, place the .jar file in both lib directories.

In order to configure an extractor to connect to a JDBC data source, you also have to define a JDBC section in your configuration file containing four parameters: DRIVER, URL, USR and PWD. The values to use for these parameters can be determined from the JDBC driver documentation.

The values for the USR and PWD parameters are the most straightforward to determine: they are the username and password used to access the external data source.

The DRIVER parameter should be set to the Java class that implements the JDBC driver (this is usually mentioned explicitly in the JDBC documentation). Additionally, most JDBC documentation shows the value of the DRIVER parameter in Java code examples that demonstrate its use. Look for a code section that registers the driver with the JDBC DriverManager, which is usually done through a Java method call of Class.forName(), using the DRIVER class as an argument.

The format of the connect URL is described in the JDBC documentation, and is usually shown in Java method examples that demonstrate the use of the driver. In the code, look for Java method call DriverManager.getConnection(). This method call uses the URL (and optionally USR and PWD) as arguments.

Properties Objects

Many JDBC drivers can accept settings defined in Properties objects that control their behavior. These Properties objects can, for example, define timeouts, whether or not to automatically commit on exit, how long to wait until a connection times out, or how caching should be handled. In Java programs, these properties can be assigned to the JDBC driver in a number of ways. In extractor configuration files, you can only set Properties if your JDBC driver allows you to embed them in the connection URL. If this is not possible, there is no other way to control the properties.

10 Appendix A UI rules Example

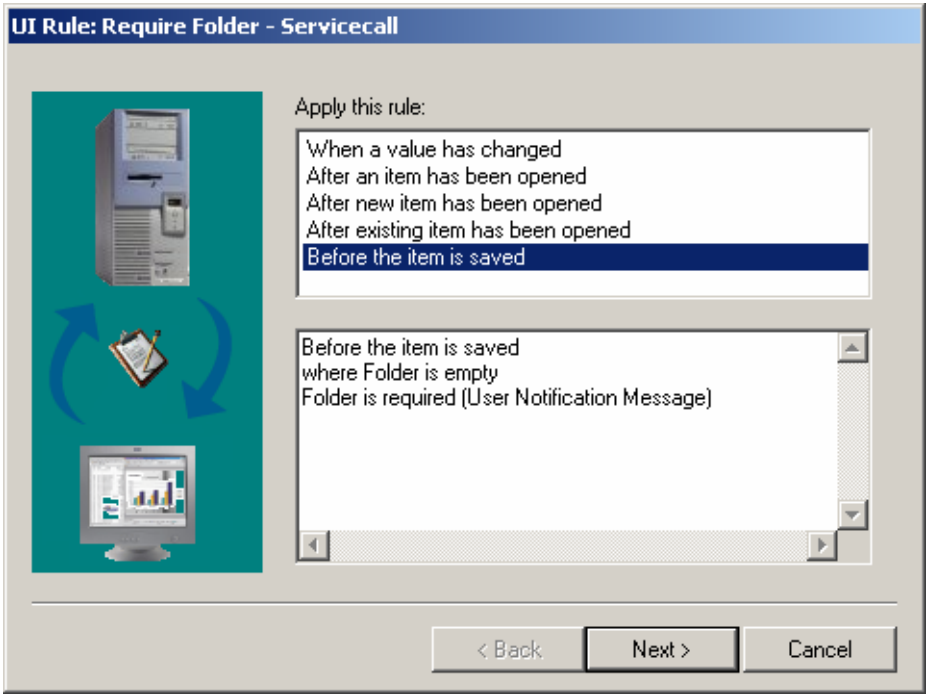
In this chapter a UI Rule is shown that forces the user to fill a field (or cancel an operation) based on the condition of other fields.

Making Form Items Required

You can force the user to either fill a field or abandon the changes made in the form using the new `Error Notification Message`. By showing it during an `On Save` trigger, it cancels the Save and keeps the window open. To create a UI rule, do the following:

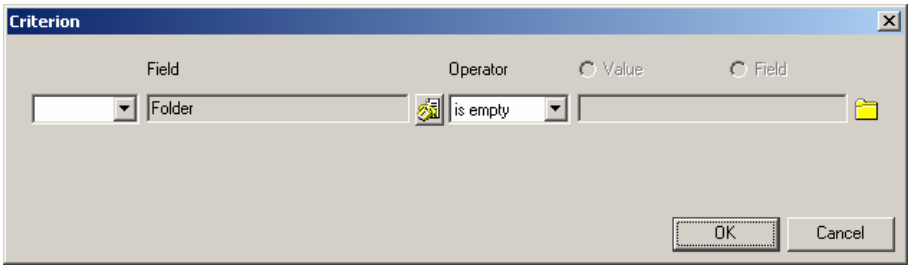
1. Choose the Before Save trigger:

Figure 10.1 Choosing the Before Save Trigger



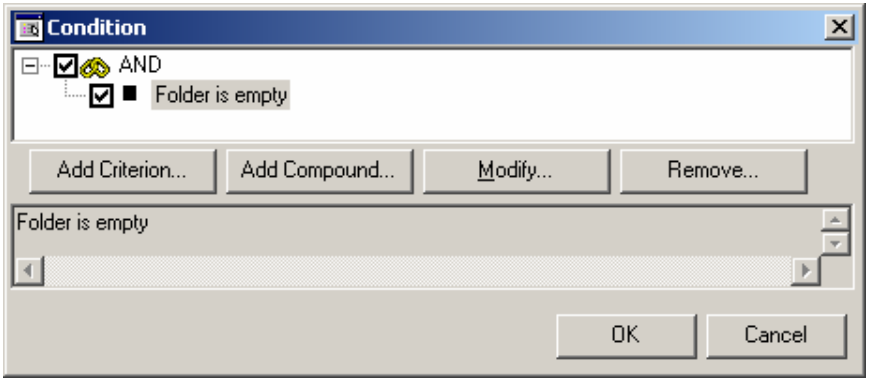
2. In the Condition text box, create the criterion shown in Figure 10.1 Choosing the Before Save Trigger above:

Figure 10.2 Check for an Empty Folder



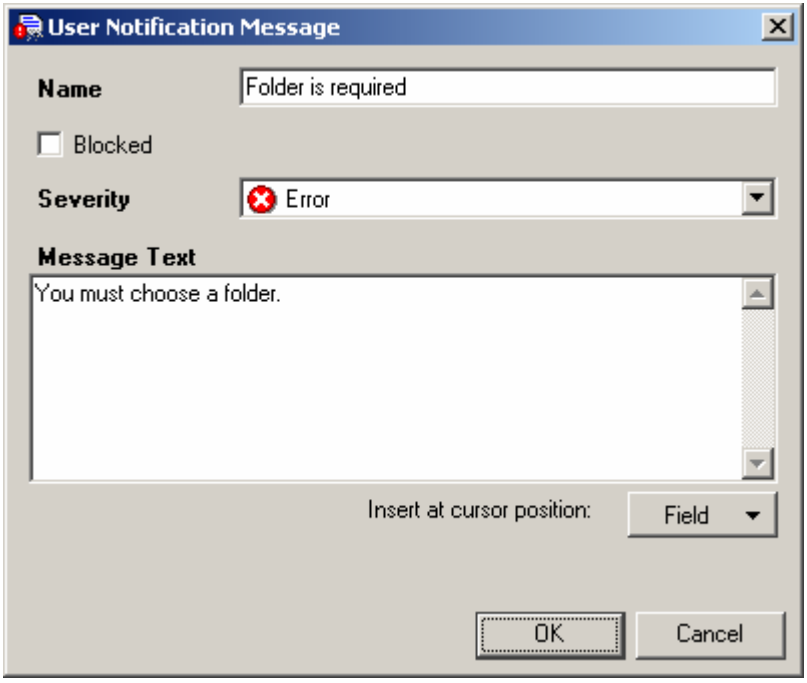
Which results in the condition shown in the figure below:

Figure 10.3 Folder is Empty Condition



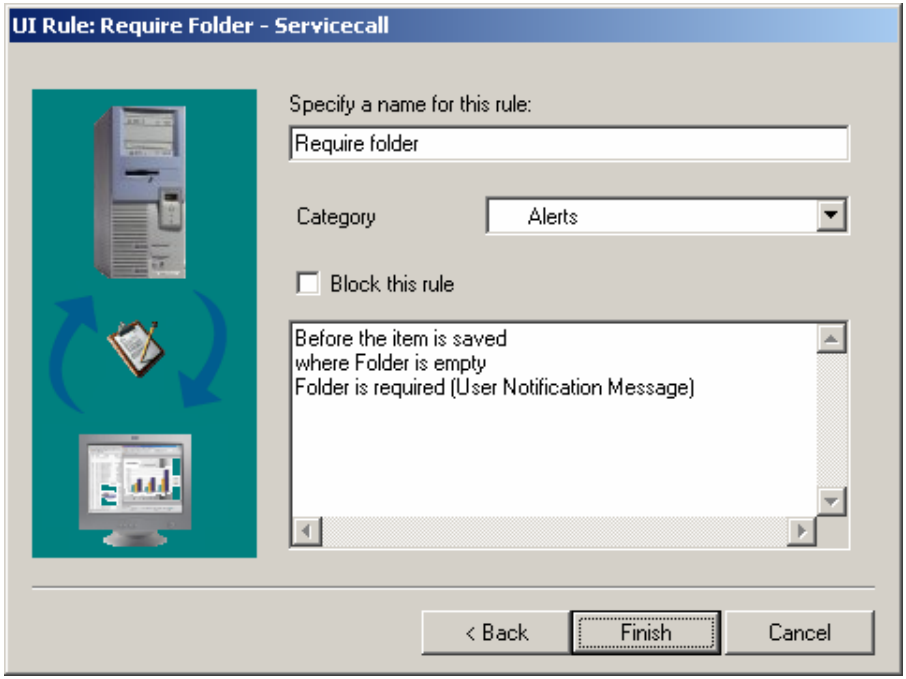
3. When creating the Message, make sure it has the `Error` severity (otherwise it will fail to cancel the Save action). See the figure below:

Figure 10.4 Preventing the User From Saving the Form



4. The rule needs a name and maybe a category for locating it later on, see the figure below:

Figure 10.5 Finished UI Rule



When the user tries to save a Service call where the folder is not filled in, this message is displayed:

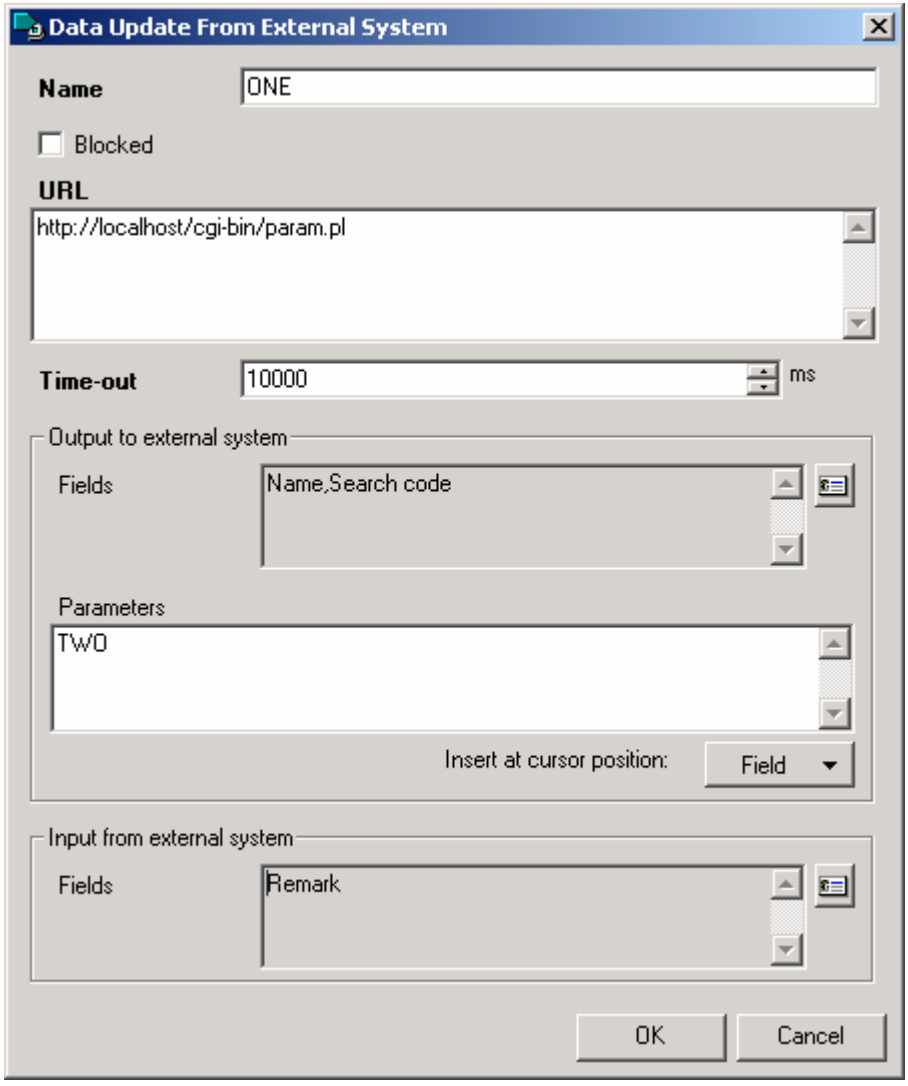
Figure 10.6 User Notification Message

After clicking OK, the form is not saved and remains open.

11 Appendix B Data Update Protocol

Introduction to External Data Update

External Data Update is an option that allows you to send a HTTP POST request to a URL on a web server, similar to what happens on many websites when you enter a form. External Data Update is configured in the UI-rules or DB-rules wizard in a dialog that looks like the one seen below:



The Name field is used to describe the action in Service Desk. It is mandatory by default. The URL field is also mandatory. It identifies how to get to the external program. Time-out is the time in milli-seconds that Service Desk should wait for an answer to be returned by the external web-program. The Fields box allows you to select fields to send to the external web-program as arguments and parameters allows you to combine fields and text as an additional parameter to the external program. The input from external system

allows you to identify the fields that the answer from the external web-program should be mapped to.
The sections “External Data Update Request” and “External Data Update Response” detail how the fields entered above are translated to a request and what kind of response is expected from the external web-program.

External Data Update Request
When the rule is triggered, Service Desk sends a POST request from the Service Desk application server to the URL entered in the URL field of the dialog.
The parameters that are sent with the POST request are taken from the other fields of the dialog. To include them with the request, the parameters are URL encoded (Content-type: application/x-www-form-urlencoded). This means that non-alphanumeric characters are replaced by % plus two uppercase characters specifying the ASCII code in hexadecimal. The request parameters are derived from the values as entered in the dialog according to the following specification:

session_id	Number that identifies current client session in SD internally
rule_name	Name field of the external data update dialog
action_name	Name field of the DB or UI rule from which the external data update was invoked.
entity_id	Number identifying current thing being manipulated (e.g. Incident, Person, Problem, Organization) in SD internally.
entity_name	Name identifying current type of thing being manipulated (e.g. Incident, Person, Problem, Organization) in SD internally.
primary key	Number identifying a field (also referred to as attribute) that uniquely identifies an particular item of type entity_name in SD internally.
parameters	The field named parameters as seen in the external data update dialog.
n_output_fields	Number of fields entered under Output to External System

If any field is entered under Output to External System, these are translated into parameters for the web-program. Starting at i=0, each field is send in the following form:

output_field_name_i =<name of the field>
output_field_id_i=<SD internal number of the field>
output_field_value_i=<current value of the field>

If any field is entered under Input from External System, thse are translated into parameters for the web-program that take the following form for each field (starting at i=0):

Input_field_name_i=<name of the field>
Input_field_id_i=<SD internal number of the field>
Input_field_value_i=<current value of the field>

Translating Service Desk attribute values to field values for the External System

The parameters sent by Service Desk as specified under Output to External System are passed on as text. However, Service Desk may not render these attribute values as text when seen in the Service Desk client. The following table shows how Service Desk attribute values are translated into text:

Type	Example Attribute	String format	Example Values
Boolean	Frequently Asked Question	`0` for false, `1` for true	`0, 1`
Date Time	Actual Start	`yyyy.MM.dd.HH.mm.ss` The date is in the timezone UTC	`2003.02.12. 17.39.00`
Entity Reference	Status, Configuration Item	The object id	`28147831406 6123`
Entity Set Reference	History lines	Not possible	
Double, Float, Integer, Long, Currency	Actual Cost	A number with a dot (.) as decimal symbol	`4.56`
Duration	Actual Duration	Number of days	`0.253472222 2222222`
String	Solution	String, the max length is checked	

Example of External Data Update request:

Take the request seen in the screenshot on the first page. Now imagine it is carried out on the Service Desk entry of a Person with Name “Babbott, Charles” and search code “BABBC” (this is a record that actually can be found in the Service Desk Demo database). The equivalent of the request made by Service Desk when entered in a web-browser would be:

```
http://localhost/cgi-bin/param.pl ?rule_name=MY DEMO RULE&action_name=ONE&entity_name=Person&parameters=TWO&n_output_fields=2&output_field_name_0=Name&output_field_value_0=Babbott,+Charles&output_field_name_1=Searchcode&output_field_value_1= BABBC
```

In the example, the expected return parameters have been omitted. In the request, there is little difference between the series of

parameters prefixed by “output_field” and its counterpart prefixed by “input_field”, other than the name itself.
Furthermore, all ID fields have been removed from the parameter list in the example as there is little that script-writers can utilize these values for. Users of the Service Desk Web-API may have a use for them, however.

External Data Update Response

Service Desk expects the external web-program to respond with a message of which the parameters are encoded as content-type: application/x-www-form-urlencoded. Service Desk does not check that you use this as content-type for the response, it just expects the body of the HTTP response to be formatted in that way. It is important that the HTTP header Content-length is included and set correctly in the response header because otherwise, Service Desk will report an IOException when the rule is executed, as it may try to read more data than actually returned by the external web-program.

The number of return parameters must be equal to the number of fields entered under “Input from External System” otherwise, Service Desk will report the error “The results array of the External Data Provider has a wrong size” after the rule is executed. The external program is expected to first specify the number of fields returned using the following parameter (where i is the actual number of fields):

n_fields=i

Then, starting at i=0, the value for each entered field is returned through parameters taking the form:

field_i=<return value>

Translating return values from the External System to Service Desk

The fields returned by the external web-program consist of nothing but text. These fields are to be translated to attributes in Service Desk of a different nature (like a checkbox or a date). The following table outlines how the string values sent by the external web-program are interpreted by Service Desk.

Type	Example Attribute	String format	Example Values
Boolean	Frequently Asked Question	When `1`, `true`, `yes` or `on` then `TRUE` else `FALSE`	`1, 0, true, FALSE, on`
Date Time	Actual Start	A datetime string. This depends on the date and time format adjustments of the current account,0 => today, `+n` => number of days after today, `-n` => number of days before today or a date in the `format"yyyy.MM.dd.HH.mm.ss` The date is parsed as local time for the current account.	`0`, `+1`, `-2`,`11-11-2002 14:20`, `2003.03.15.14.20.59`, `NULL`
Entity Reference	Status, Configuration Item	An existing object id or an unique value of the Search attribute of the Related Entity	`28147831406612 3`, `Waiting`, `NULL`
Entity Set Reference	History lines	Not possible.	
Double, Float, Integer, Long, Currency	Actual Cost	A valid number. This value depends on the number format adjustments of the current account. The use of "," or "." depends on the number adjustments of the current account.	`2.5`, `2323`, `NULL`
Duration	Actual Duration	Number of days. The use of "," or "." depends on the number adjustments of the current account.	`1.5`, `NULL`
String	Solution	String, the max length is checked.	unicode text, `NULL`

Example of response:
Given that Output to External System for the example request seen earlier is set to Remark and Identification Number and the values returned by the program are nothing but the field names with ValueFor prefixed, the response sent back by the web-server is:

```
Content-length: 38

n_fields=1&field_0=ValueForRemark&field_1=ValueForIdentificationNumber
```

In Service Desk, the field Remark would then be set to "ValueForRemark" and the field Identification Number to "ValueForIdentificationNumber".

Example program:
Put the following program in the CGI directory on an Apache web-server that also has Perl installed. Change the first line (the shebang) so it is set to the actual location of the Perl interpreter on the local file system of the web-server.

Next, create a UI rule using the settings from the screenshot at the top of this appendix.

```
#!c:/perl/bin/perl
# Script : edu_param.pl
# Author : Max Hirschfeld
#
# For use with External Data Update feature of UI and DB rules as
introduced
# in Service Desk 4.5 Service Pack 7. Put it in a web server's
cgi directory,
# correct the shebang to match local settings, and configure a UI
or DB rule
# that utilizes it.
#
# This script returns all parameters sent to it by SD External
Data Update.
# Map 1st return field in External Data Update to a big text
field
# to see the result.
#

use CGI qw/:standard/;


sub get_sd_external_data_update_fields
######################################################################
```

```
###############
# reads fields as sent by the External Data Update interface
# introduced in Service Desk 4.5 SP7
#############################################################################
###############
{
    my $n_fields = param("n_output_fields");
    my %fieldhash       = ();

    foreach (0..$n_fields - 1) {
        $key             = param("output_field_name_$_");
        $value           = param("output_field_value_$_");
        last if not defined $key or not defined $value;
                            $fieldhash{$key} = $value;
                        }
                        return %fieldhash;
                    } # end get_sd_external_data_update_fields


sub write_sd_external_data_update_response(@)
#############################################################################
###############
# writes responses in the format expected by the External Data
Update interface
# introduced in Service Desk 4.5 SP7.
#############################################################################
###############
{
   my $fieldidx     = 0;
   my @fields       = map { 'field_'.$fieldidx++.'='.$_; } @_;
   my $nr_of_fields = scalar @fields;
   my $message_body = "n_fields=$nr_of_fields&". join("&",
@fields);

   print "Content-length: ". length($message_body)
."\n\n$message_body";
} # end write_sd_external_data_update_response


#############################################################################
###############
# main
#############################################################################
###############
{
   my %fields_from_sd = get_sd_external_data_update_fields();
   my $output         = '';

   foreach (keys %fields_from_sd) {
       $output .= "$_=$fields_from_sd{$_}, ";
   }
   $output .= join(', ',
               "parameters=". param('parameters'),
               "session_id=". param('session_id'),
               "rule_name=". param('rule_name'),
               "action_name=". param('action_name'),
               "entity_id=". param('entity_id'),
```

```
                "entity_name=". param('entity_name'),
                "primary_key=". param('primary_key'));

    write_sd_external_data_update_response($output);

}
```