HP OpenView Service Desk

for the UNIX and Windows operating systems

Software Version: 4.5 Service Pack 20

Consolidated Documentation Addendum

Manufacturing Part Number: None

Document Release Date: September 2006

Software Release Date: September 2006

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

Copyright Notices

Trademark Notices

Documentation Updates

This manual's title page 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, visit the following URL:

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

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

Support

You can visit the HP OpenView support web site at:

http://www.hp.com/managementsoftware/support

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

HP OpenView online software 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 site to:

- Search for knowledge documents of interest
- Submit enhancement requests online
- Download software patches
- Submit and track progress on support cases
- Manage a support contract
- Look up HP support contacts
- Review information about available services
- Enter 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 find more information about access levels, go to:

http://www.hp.com/managementsoftware/access_level

To register for an HP Passport ID, go to:

http://www.managementsoftware.hp.com/passport-registration.html

Contents

1 Overview of Service Pack Documentation

Introduction

This document presents information previously released with Service Desk 4.5 Service Packs 7 through 19, and previously unreleased information explaining features delivered with Service Pack 20.

Audience

This document is intended for the following audiences:

- System administrators who install and maintain HP OpenView Service Desk 4.5 Service Packs
- Service Desk users who want to understand the features included with Service Desk 4.5 Service Packs

Related Documents

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.

> 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 `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.

Documented Service Pack Features

The following tables list the Service Pack features that are described in this document. Please note that Service Pack 7 features were originally documented in *HP OpenView Service Desk 4.5 Documentation Addendum for Service Pack 8*.

Operational Features

Table 1 Operational Features

Enhancement Number	Service Pack	Description
itsm004619	SP07	Embedding URLs into Service Desk text fields
itsm005173	SP17	Attachments can be sent via Database Rule now
itsm008088	SP17	Generic Relation feature is introduced to Service Desk and a new UI Rule action "Limit field value range by relations" is added based on this

Enhancement Number	Service Pack	Description
itsm007175	SP19	User can save current form by UI Rule action and UI rule now have explicit execution order
itsm008424	SP19	User can define complex filter condition using "OR" and "Negate" compounds
itsm008425	SP19	Client email function supported and email history with Service Call can be stored
itsm004456	SP20	Templates should be stored in Folders
itsm004852	SP20	User can sort order on query results with query restriction
itsm004127	SP20	User can get notified when item assigned to his group
itsm005140	SP20	Add functionality to share views between users
itsm005878	SP20	Approval view and approval mechanism enhancements
itsm008584	SP20	Custom fields GUI limitation fix
itsm008741	SP20	Notification alert upon editing the same record more than once

System Features

Table 2 System Features

Enhancement Number	Service Pack	Description
itsm004690	SP07	UI Rule to fire on Window Open or Close
itsm005577	SP07	Request to create a DB or UI rule that "appends" text to a text field
itsm006033	SP07	Quick Find should have an option to display only Services for the caller
itsm004780	SP11	Ability to audit changes made from within the administrator's console
itsm007695	SP15	Three new filter operators are added for date type fields
itsm007474	SP16	Attachment folder structure changed and requires migration of existing attachments
itsm008291	SP18	Layout of currently opened form can be changed dynamically based on values selected in the form
itsm008292	SP18	Now user is able to add extra fields to Service Desk item
itsm008332	SP18	More than one event can be defined for one Database Rule or UI Rule

Enhancement Number	Service Pack	Description
itsm004791	SP20	Auditing attachments
itsm008585	SP20	Custom Codes enhancement

Administrative Features

General Administration

Table 3 General Administration

Enhancement Number	Service Pack	Description
itsm006537	SP11	Applying a hot fix or service pack to Service Desk console clients is cumbersome
itsm008586	SP20	Provide SMTP authentication for outbound mail (PLAIN/LOGIN/CRAM-MD5)

Integration Enhancements and Support

Table 4 Integration Enhancements and Support

Enhancement Number	Service Pack	Description
itsm006849	SP12	WBEM ODBC required for OVOW integration is not available in Windows 2003 family

Platform Support

Table 5 Platform Support

Enhancement Number	Service Pack	Description
itsm007844	SP16	Support of Service Desk 4.5 client on Citrix Access Suite 4.0
itsm007869	SP16	Support of Service Desk 4.5 integrated with OVOW Cluster

2 Service Pack 7

The information in this chapter describes features that were provided with Service Pack 7 and documented in *HP OpenView Service Desk 4.5 Documentation Addendum for Service Pack 8*.

Rules

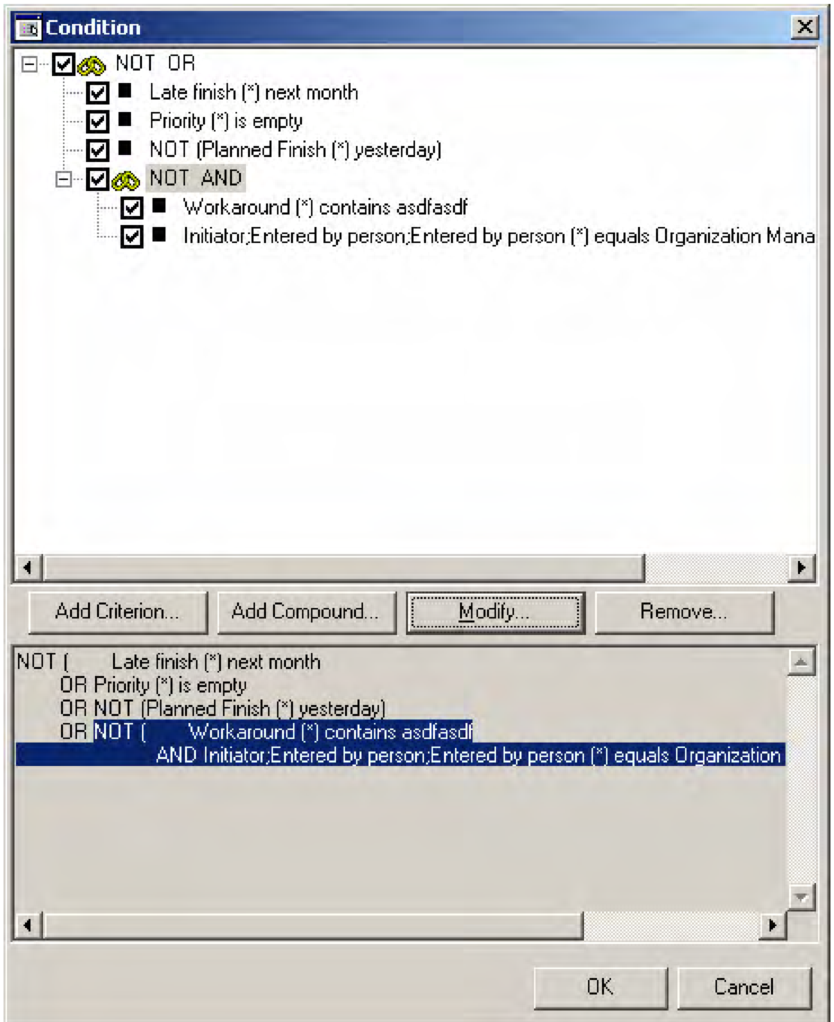
A rule consists of actions that are executed when certain specified conditions are met. They can be divided into User Interface (UI) rules and Database (DB) rules. For more information on these rules, see Chapter 4 of *Service Desk Administrator's Guide*.

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 Creating a Condition

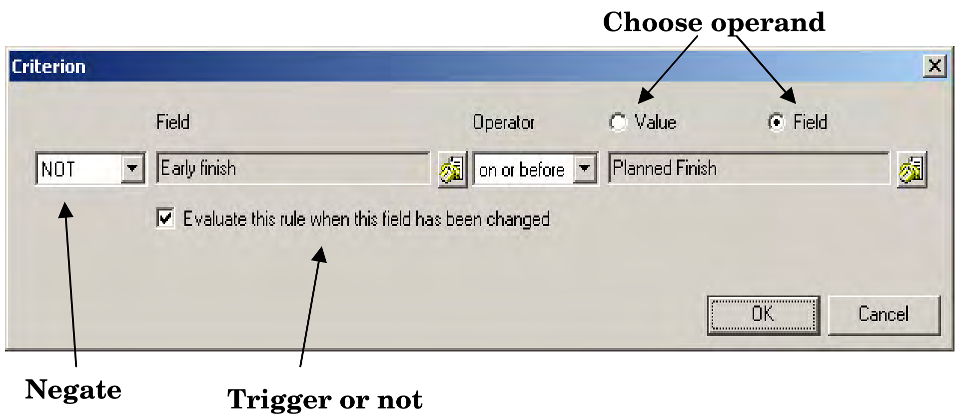


Criteria

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 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.

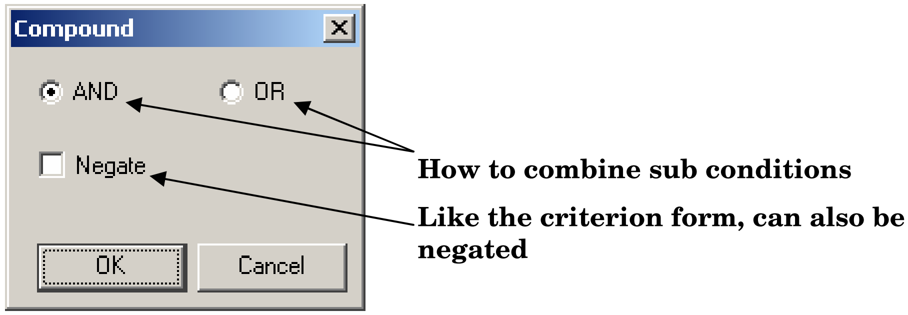
Compounds

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 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 4 A Data Update Action

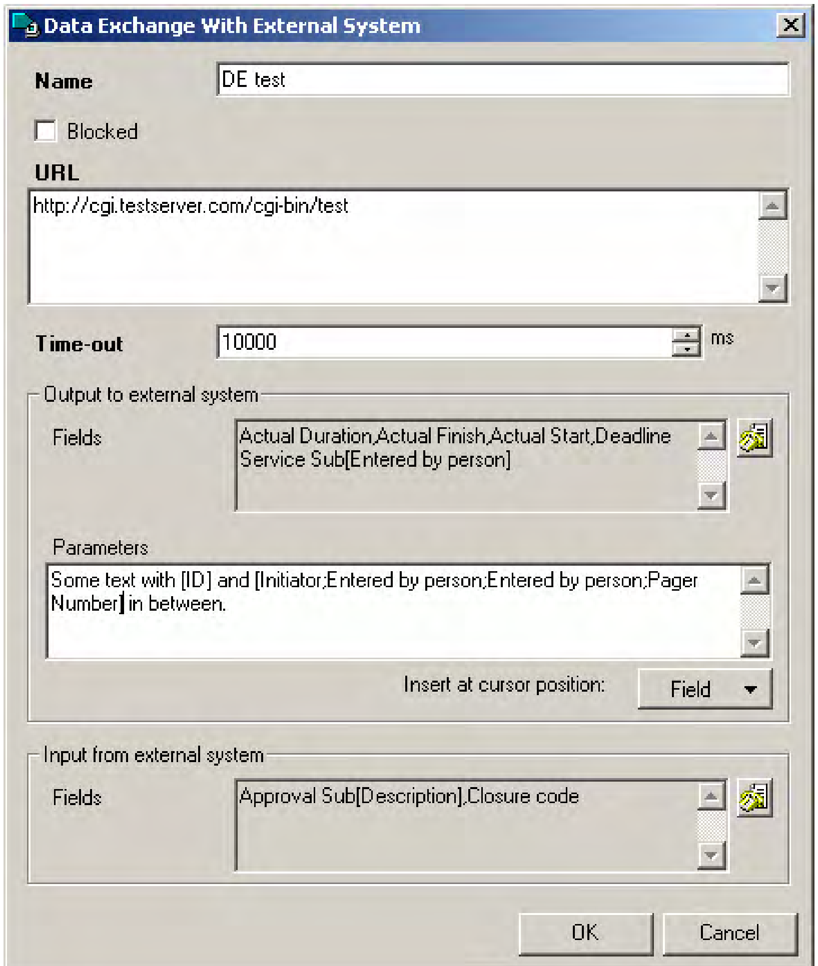
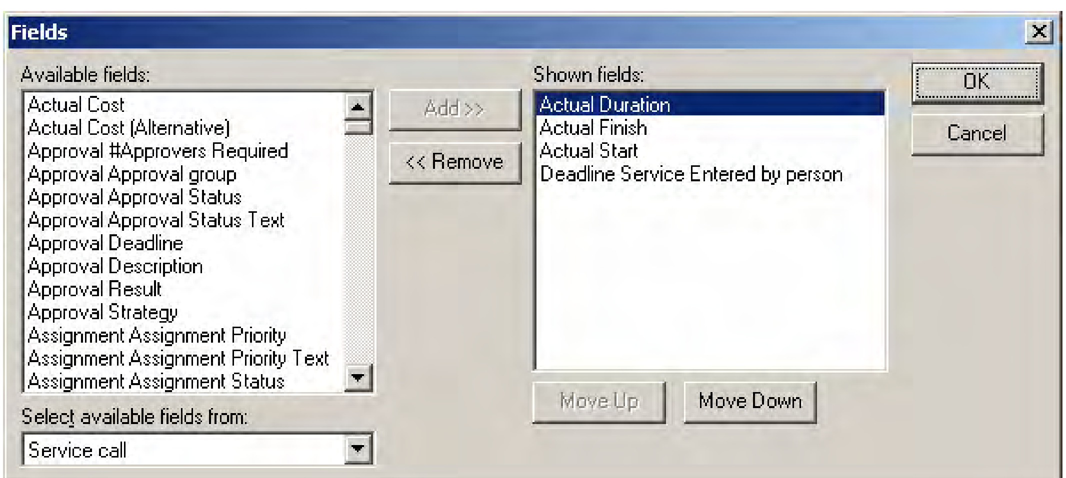


Figure 5 Selecting Output Fields for the Data Update Action



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.

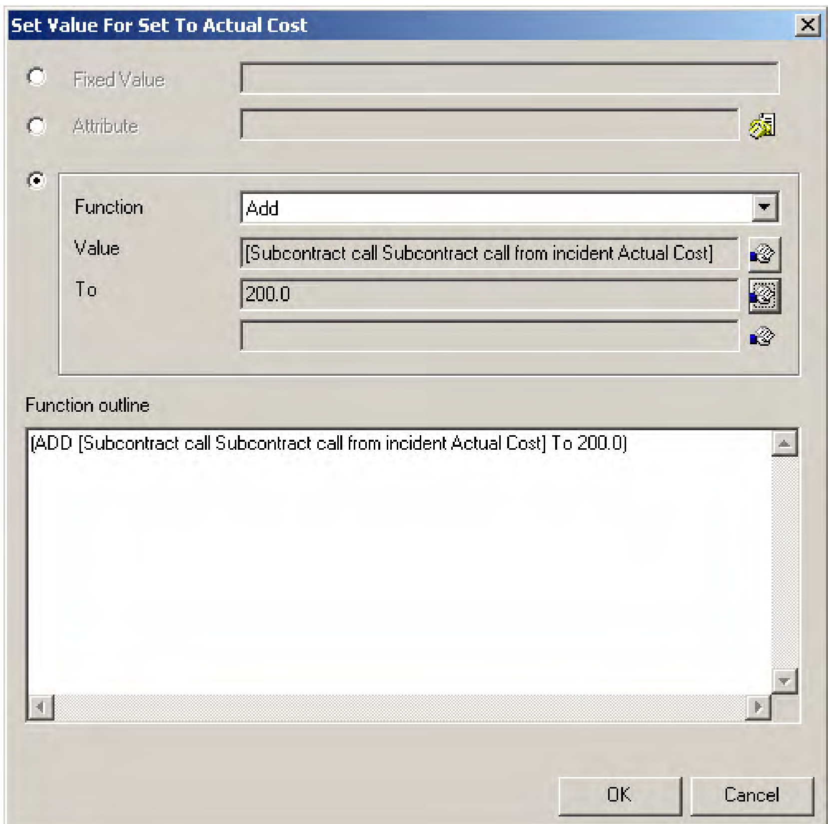
On the web server any action can be taken and new values returned to the input fields. These fields are set and in turn can execute other UI Rules.

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.

Figure 6 Function Editor



You can use the Function Editor to enter functions, see Figure 6. 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`*

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

```
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 *Special Functions*) 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.

Impacted Services View

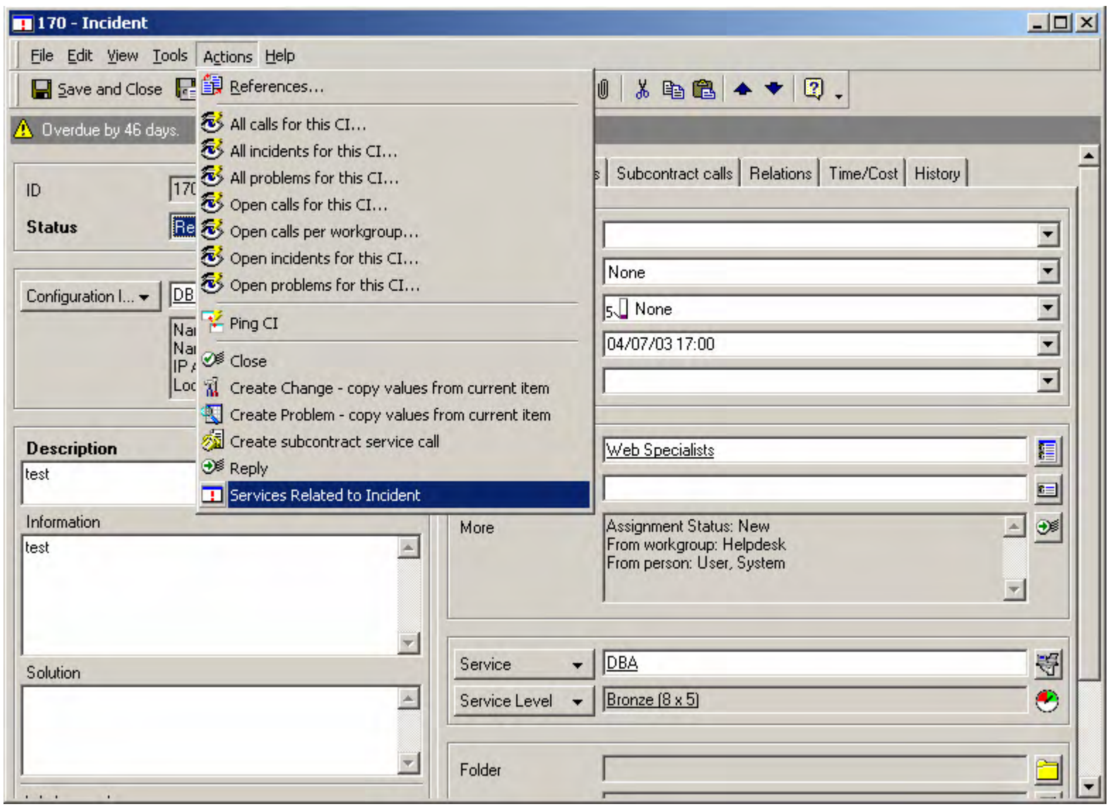
The Impacted Services View shows all Services that might be impacted by an Incident or Service Call. The primary purpose is to allow you to contact customers who may be affected by the Incident/Service Call as quickly as possible, without having to open several Service Desk Forms to find the necessary 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.

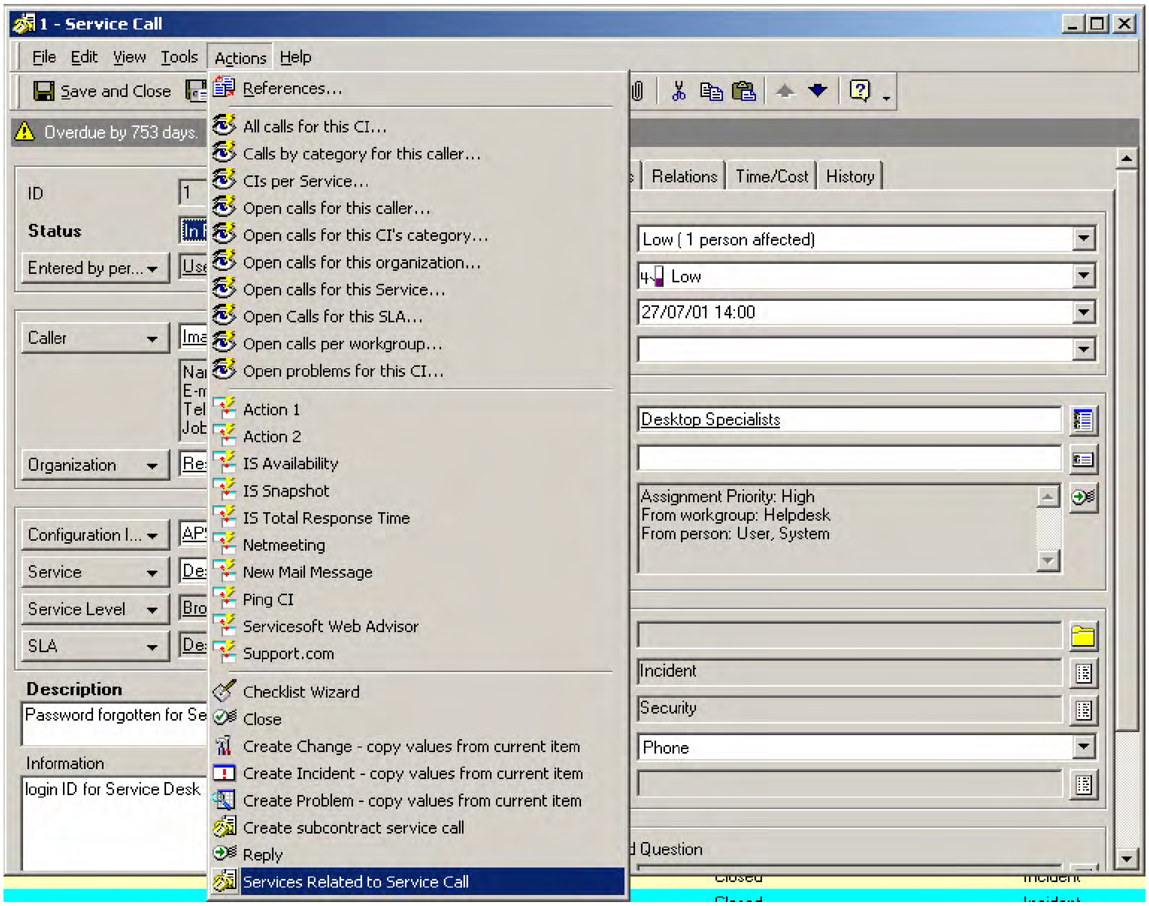
To start the view from the `Incident` form, select the `Actions` menu and then select the `Services Related to Incident` option, see Figure 7:

Figure 7 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 Figure 8:

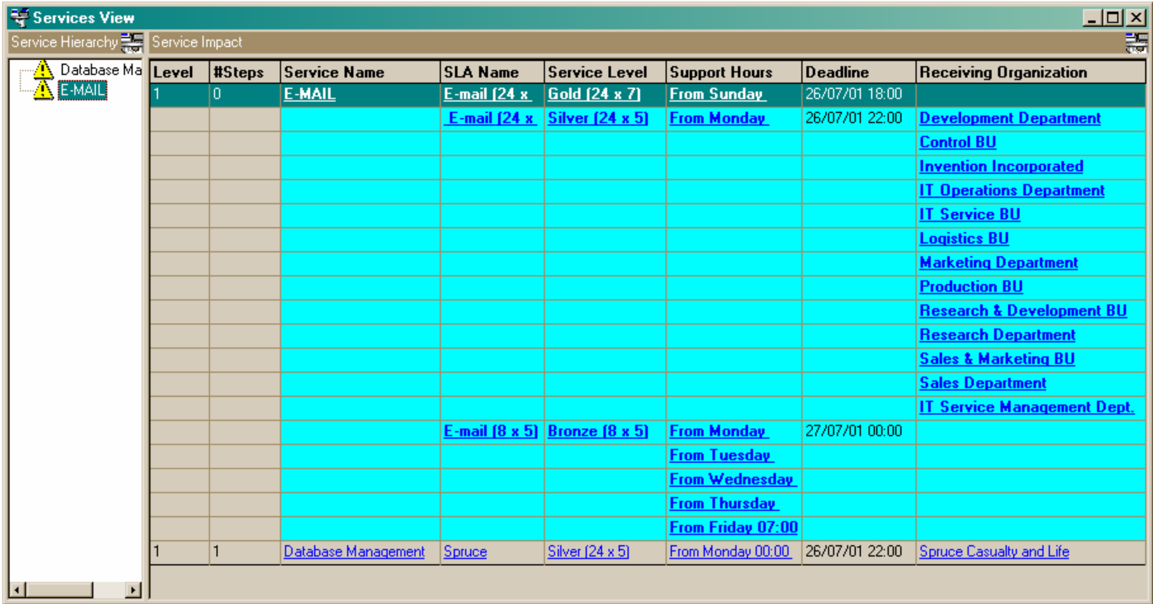
Figure 8 Impacted Services View in the Service Call Form



Using the Impacted Services View

The view consists of two parts: the `Service Hierarchy` tree and the `Service Impact` table. See Figure 9:

Figure 9 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 1 Service Hierarchy Tree Icons

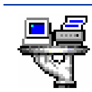
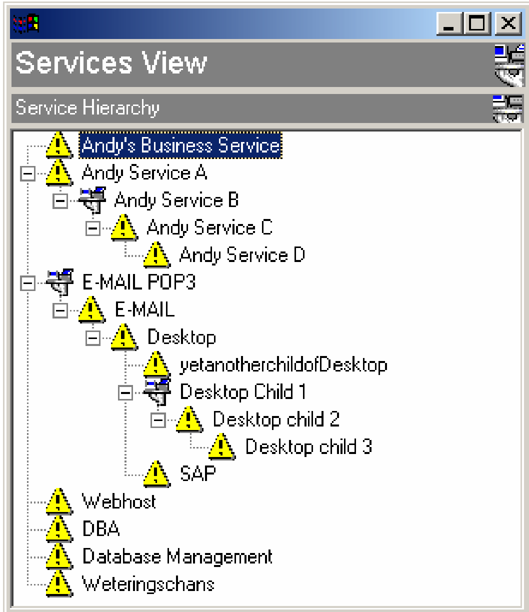
Icon	Explanation
	Impacted Services
	Non-impacted services

Figure 10 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.

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 Figure 11. 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:

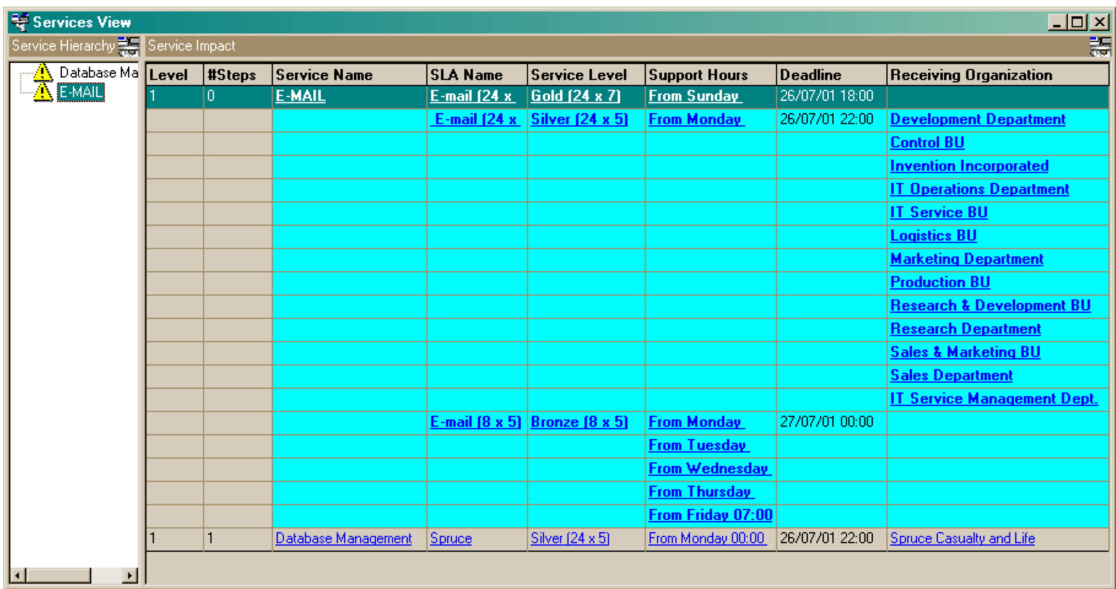
Table 2 Services Related to the CI and/or Service

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.

Level	Position in the Services Hierarchy.
`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.

If more than one SLA or Receiving Organization is associated with a Service, the information is displayed on multiple rows in the table (see Figure 11).

Figure 11 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.

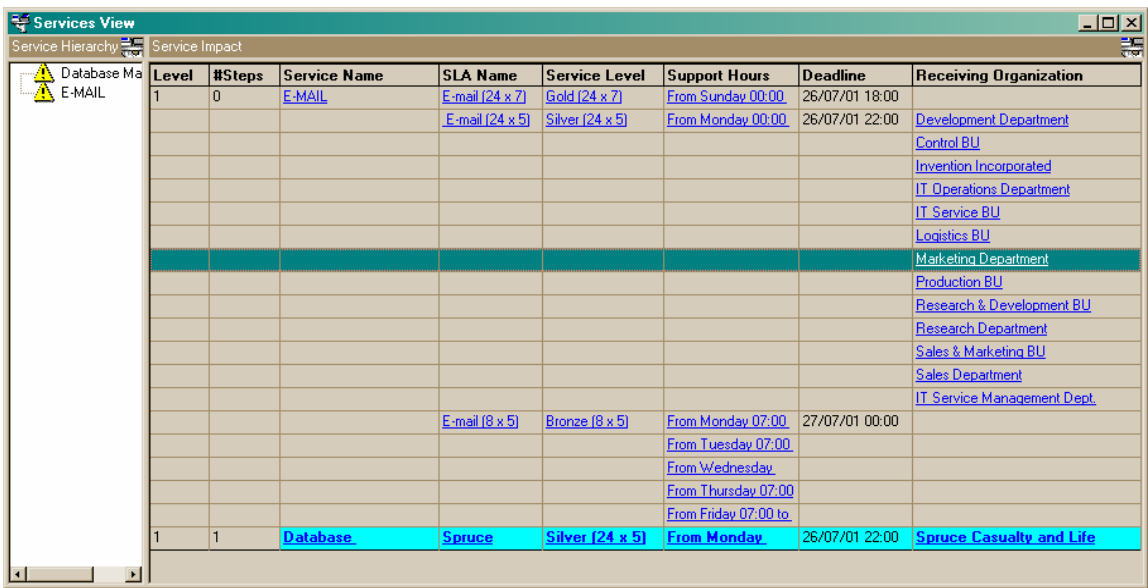
To summarize, the table has been designed so that the information at the top of the table has the greatest impact on the customer. This results in a quicker response to the customer.

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.

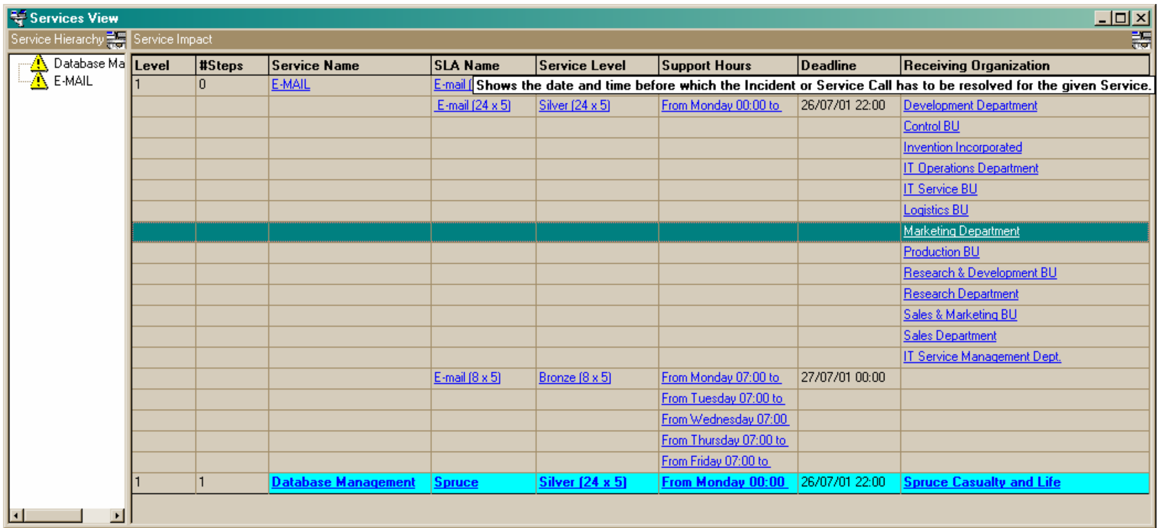
Figure 12 Hyperlinks



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 Figure 13:

Figure 13 Tool-tip



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.

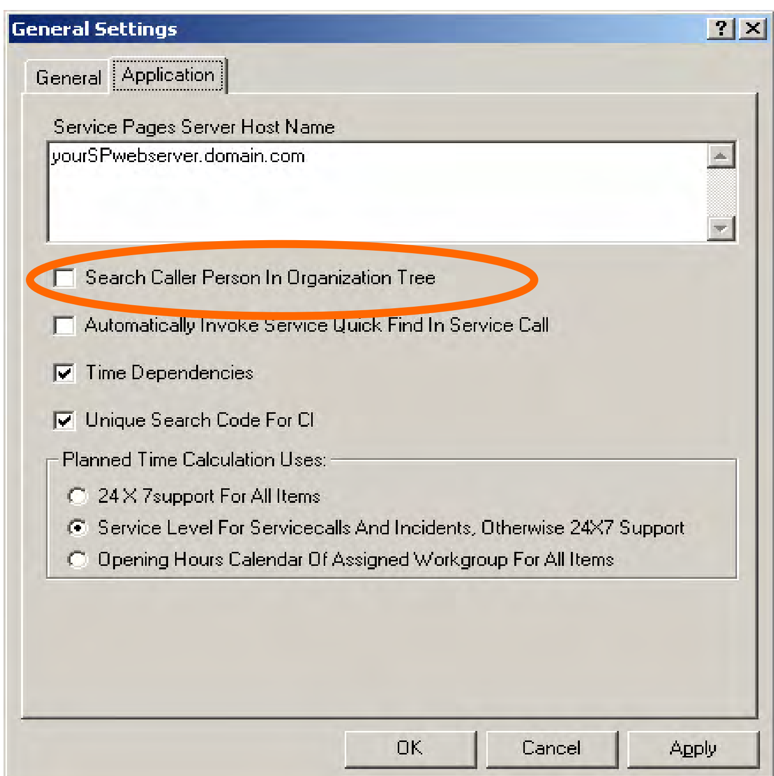
Assisted selection of service by CI

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):

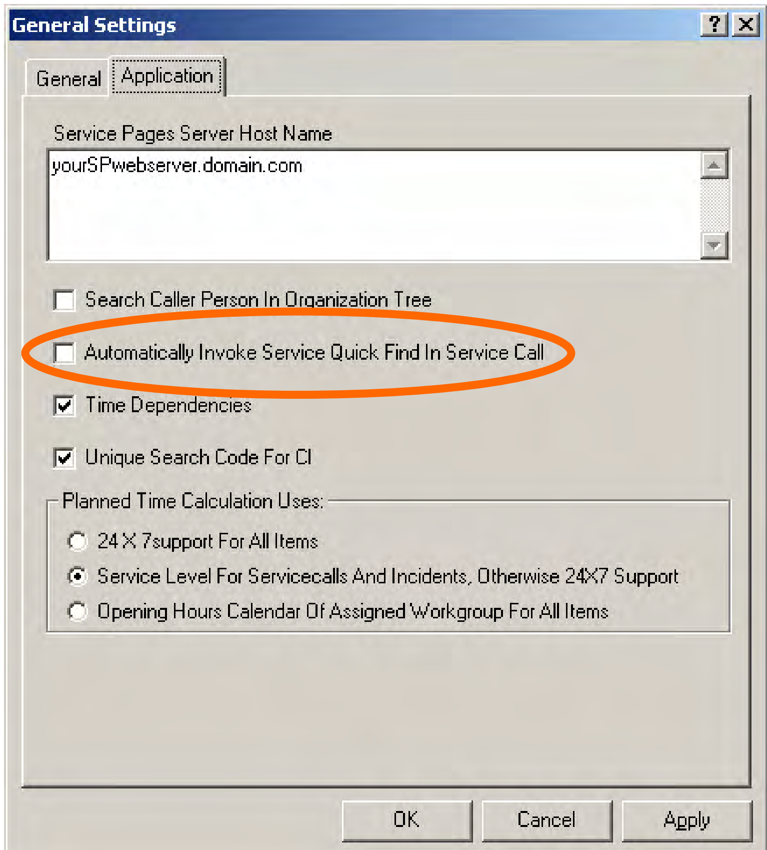
Figure 14 General Settings – Selecting Caller



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 Figure 15:

Figure 15 Automatic Quick Find in Service Call Switch

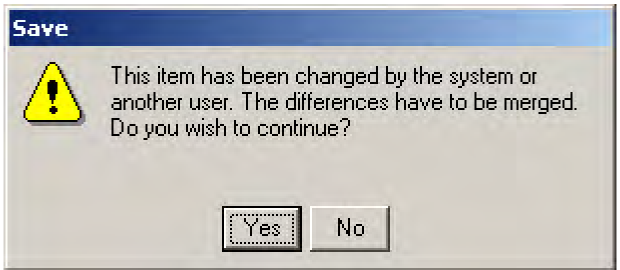


Automatic Merge of Concurrent Changes

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 Figure 16) asks the user if they would like to merge the data:

Figure 16 Merge Request



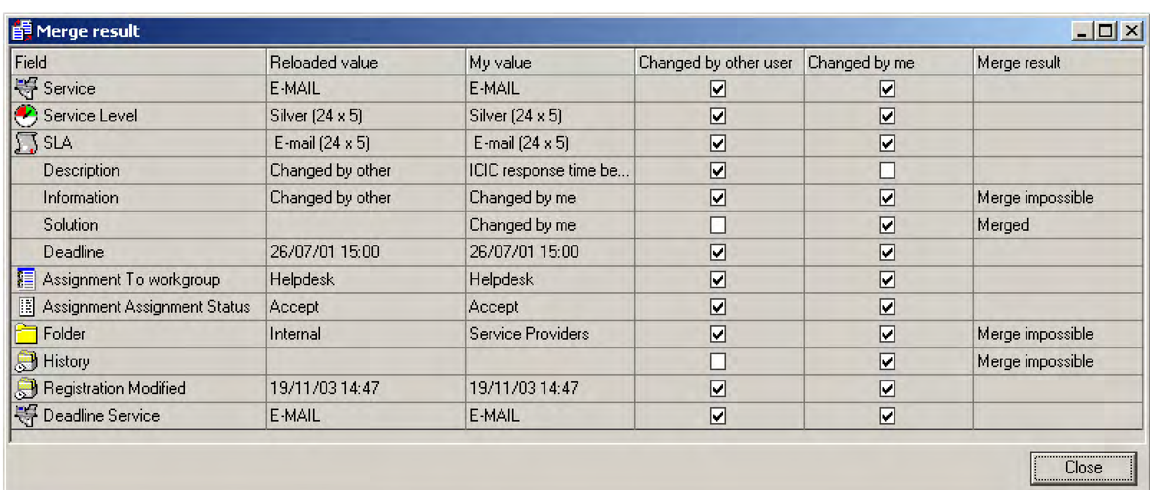
Clicking the `No` button cancels the save. If you select the `Yes` button, the fields are merged. The following table lists the possible options and outcomes:

Table 3 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 Figure 17:

Figure 17 Merging Actions



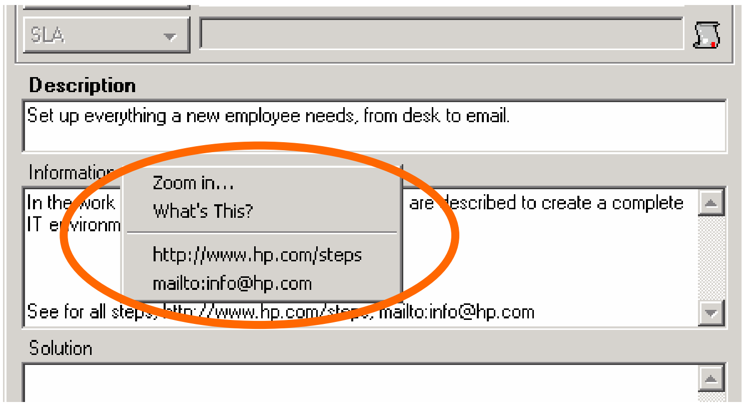
Field	Reloaded value	My value	Changed by other user	Changed by me	Merge result
Service	E-MAIL	E-MAIL	☑	☑	
Service Level	Silver (24 x 5)	Silver (24 x 5)	☑	☑	
SLA	E-mail (24 x 5)	E-mail (24 x 5)	☑	☑	
Description	Changed by other	ICIC response time be...	☑	☐	
Information	Changed by other	Changed by me	☑	☑	Merge impossible
Solution		Changed by me	☐	☑	Merged
Deadline	26/07/01 15:00	26/07/01 15:00	☑	☑	
Assignment To workgroup	Helpdesk	Helpdesk	☑	☑	
Assignment Assignment Status	Accept	Accept	☑	☑	
Folder	Internal	Service Providers	☑	☑	Merge impossible
History			☐	☑	Merge impossible
Registration Modified	19/11/03 14:47	19/11/03 14:47	☑	☑	
Deadline Service	E-MAIL	E-MAIL	☑	☑	

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

Hyperlink Support

You can now use hyper links in text fields. In Figure 18, you can see the hyperlinks when you right-click on the `Information` text box:

Figure 18 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 Table 4:

Table 4 Special Characters for Hyperlinks

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

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

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

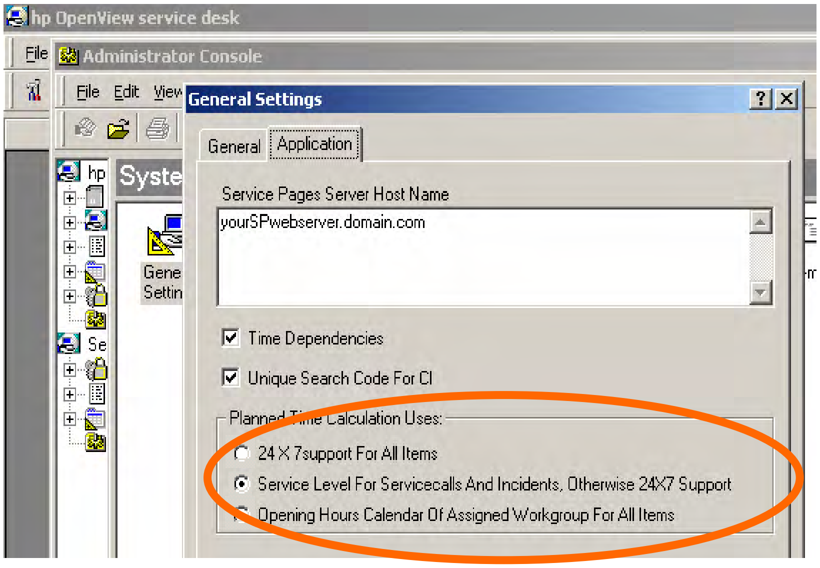
When a hyperlink is selected in the pop-up menu, the linked file is opened up in the application registered to open that type of file (for example, a HTML page is opened in a browser).

Empty hyperlinks are discarded from the list of hyperlinks.

Workflow Calculation per Workorder

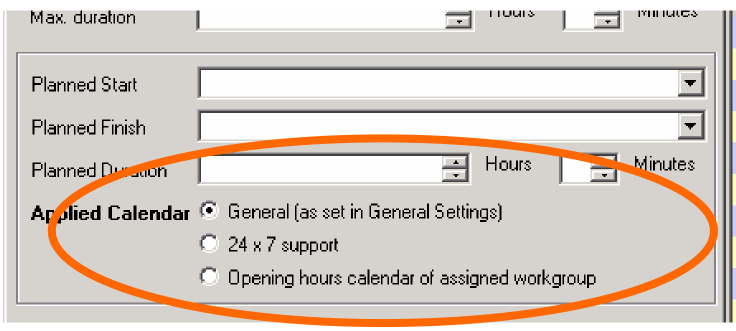
In Service Desk 4.5, determining how Workflow calculations are performed (the calculation of Planned Start, Planned Finish and resulting Planned Duration values) was performed at the (highest) level of the overall application. For example, Changes, Service Calls, and Workorders used the same calculation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

Figure 19 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 Figure 20:

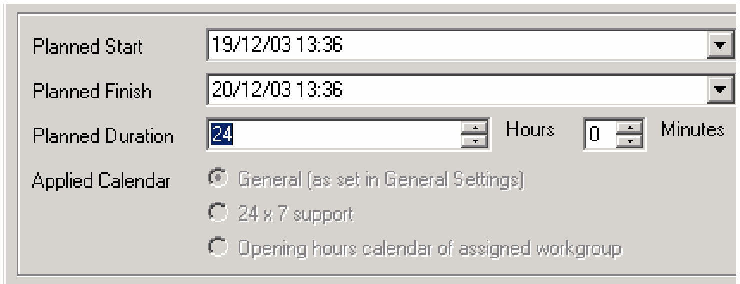
Figure 20 Applied Calendar Settings



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 Figure 21:

Figure 21 Check Box Disabled



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 5 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

Name	Description	Client/ Server	Example
$ACCOUNTNAME$	The Service Desk login name of the logged on user		`bobby`
$ITEMTYPE$	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
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

Name	Description	Client/ Server	Example
%<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

Buttons in Forms

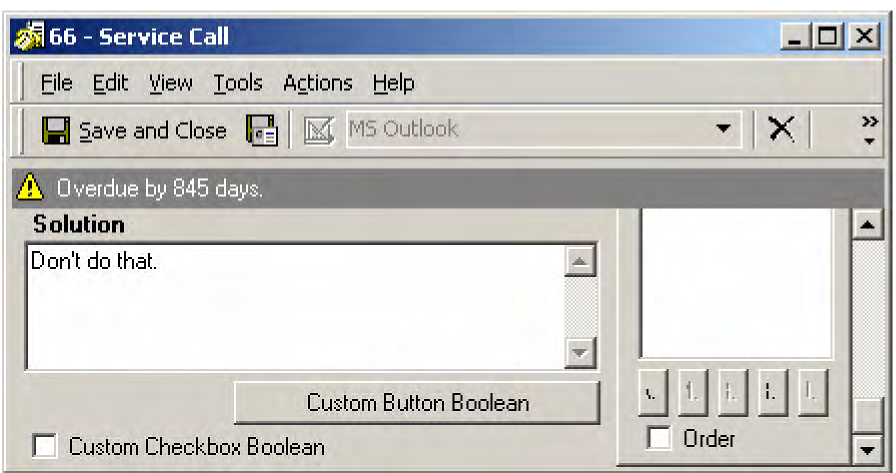
You can now add buttons to forms and these buttons can be used to trigger UI Rules (which in turn execute actions). These buttons are a special display form of `Boolean` fields. An UI rule is created for each button that triggers once the state of the button changes value to `On`. The first action to 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 Figure 22 for some example fields on a form:

Figure 22 Button Boolean and Normal Boolean



---

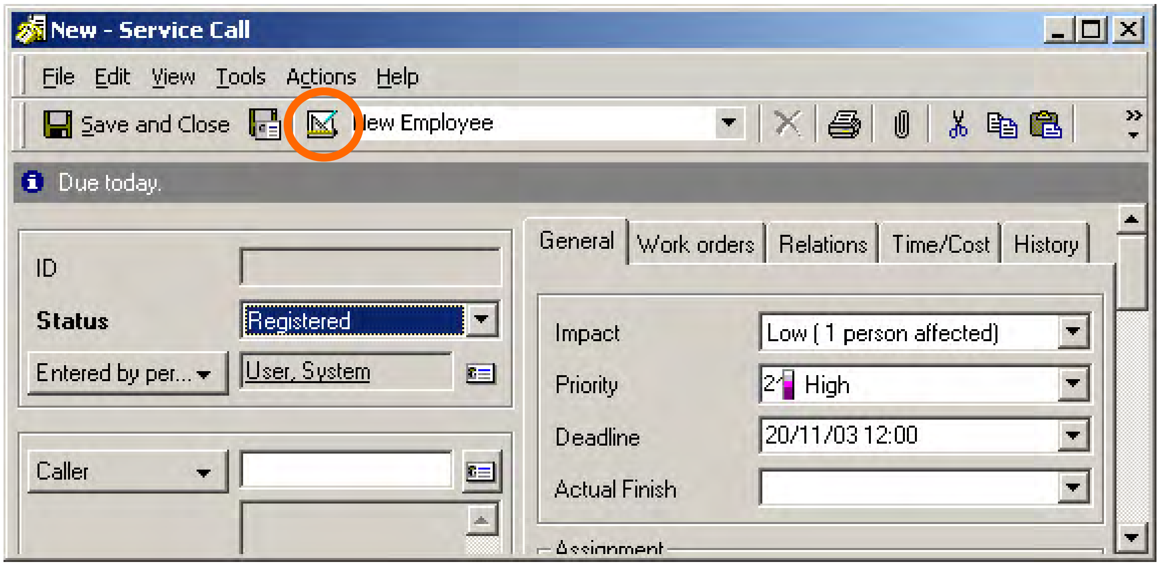
[3] If the variable (either Java or OS) can not be found, the variable is just left unchanged in the text.

[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.

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 Figure 23). 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 23 Creating a New Service Call - Choosing a Template



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
```

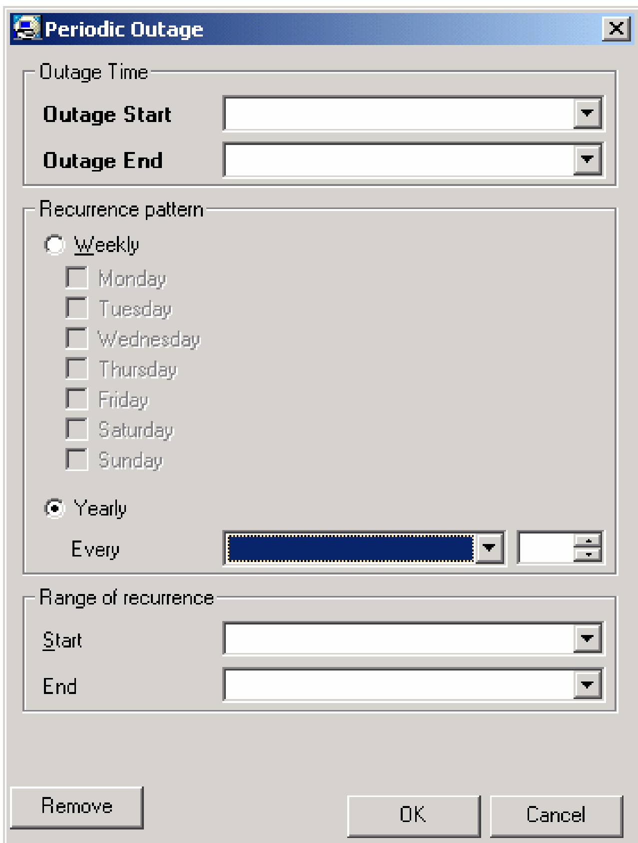
OVO for UNIX Integration

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 24 Periodic Outage Form



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

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:

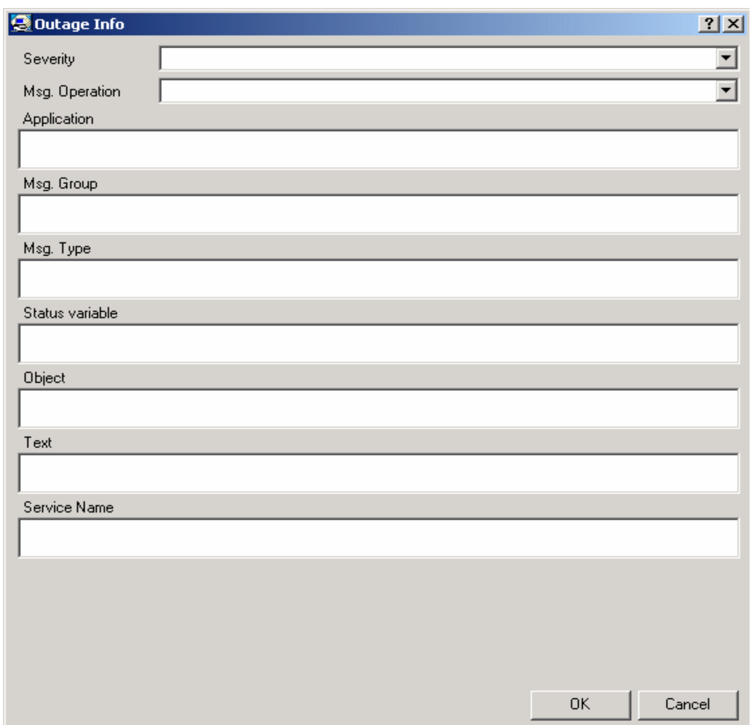
- A missing `Range of Recurrence Start` date is interpreted to mean the date specified in the `Every` field. A missing `Range of Recurrence End` is interpreted as the `Range of Recurrence Start` date plus 25 years.
- If both the `Range of Recurrence Start` and `End` fields are not defined, the yearly 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 25 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 6 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.

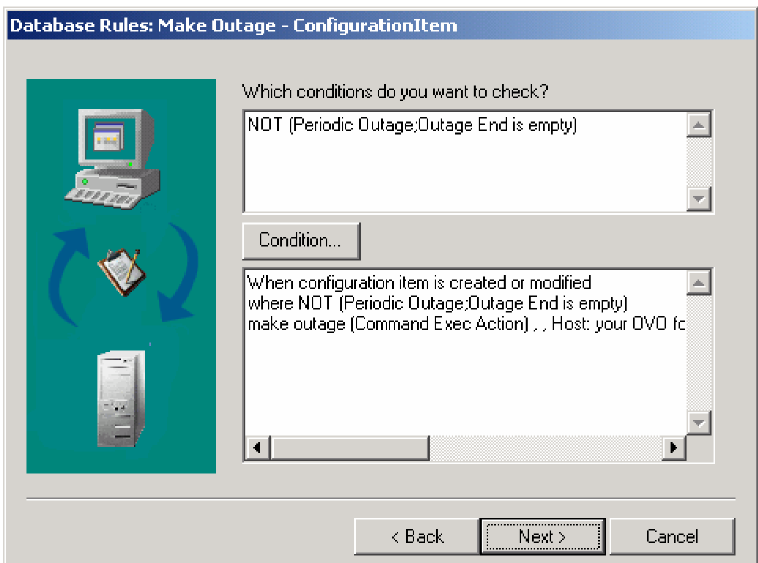
Creating a Database Rule for Periodic Outage Information

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:

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.

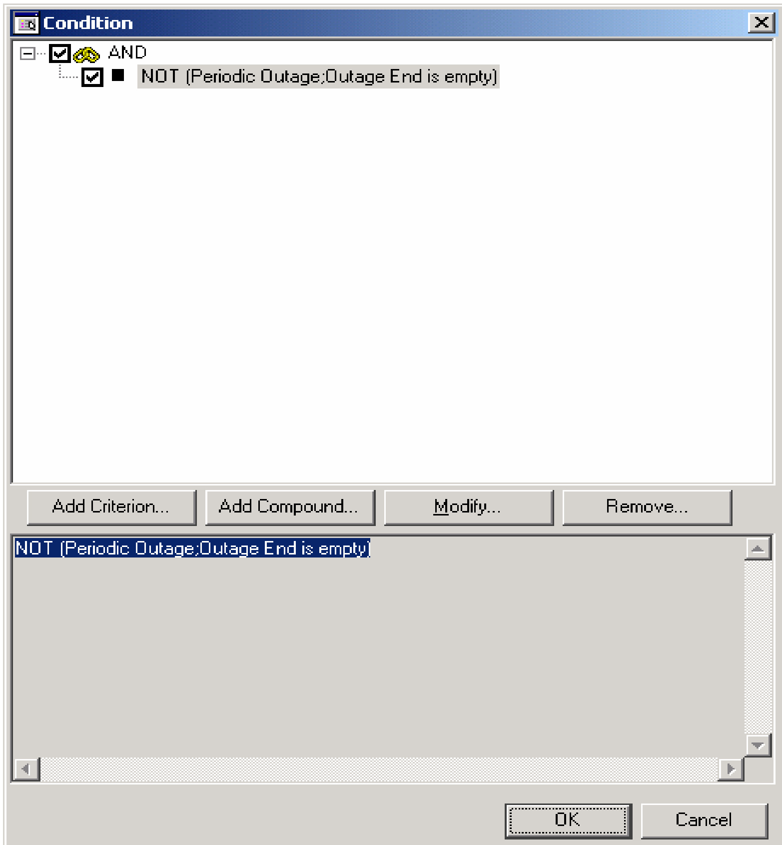
Select **When configuration item is created or modified**, then click **Next**. The following window is displayed:

Figure 26 Defining a Condition



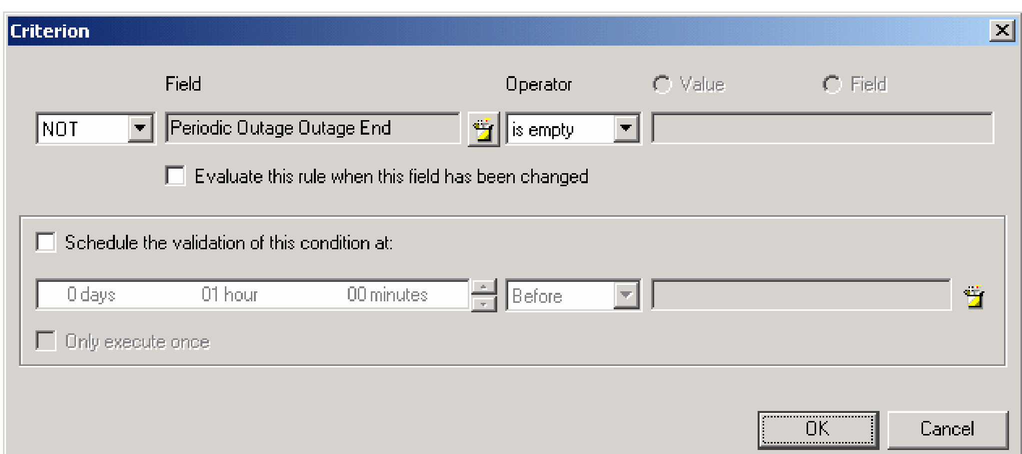
Click **Condition** to set up a new condition. The following window opens:

Figure 27 Condition Window



Click **Add Criterion** to define a criterion that limits the number of times the database rule is triggered, see Figure 28:

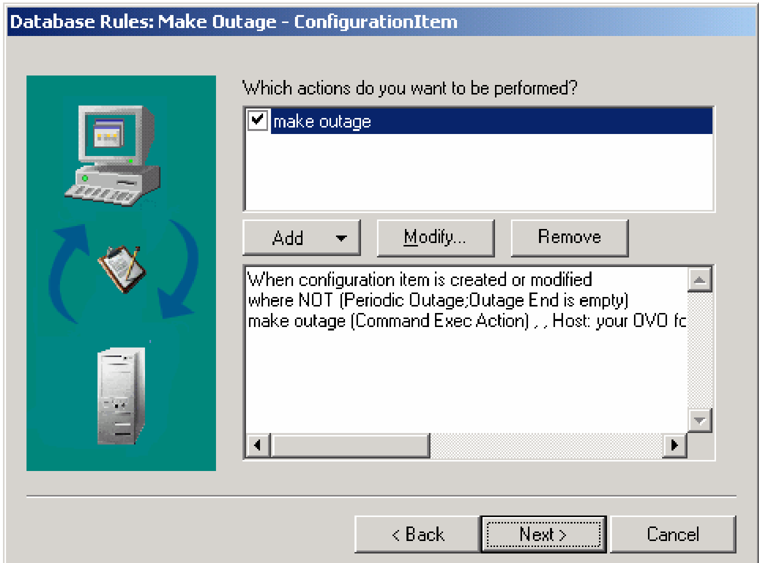
Figure 28 Criterion Window



Define the criterion according to the definition seen in the screenshot above, then click **OK**. Follow the onscreen instructions.

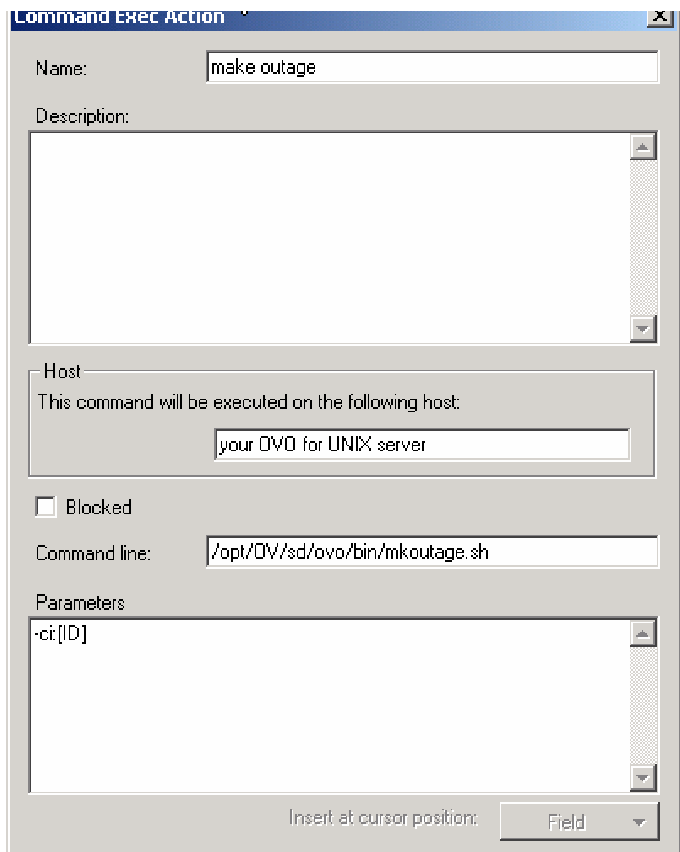
You then return to the screenshot seen before that one, on which you will have to click **Next** to go forward to the next step of the database rule setup, which is seen in Figure 29:

Figure 29 Make Outage Configuration Item



Click **Add** to define a database rule action, see Figure 30:

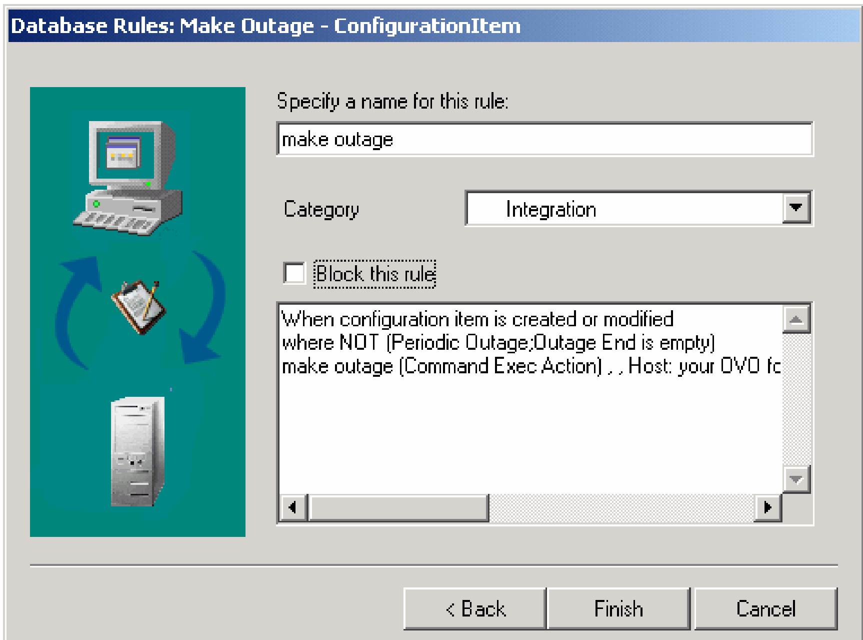
Figure 30 Command Exec Action Form



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 click **OK** when you are finished. This will return you to the previous window, in which you have to click **Next** to proceed to the final screen of the database rule wizard.

In this final window of the database rule wizard, you should select a name for the rule and a category before you can click **Finish**. Do not select the `block this rule` check box, otherwise the rule will remain inactive. See Figure 31:

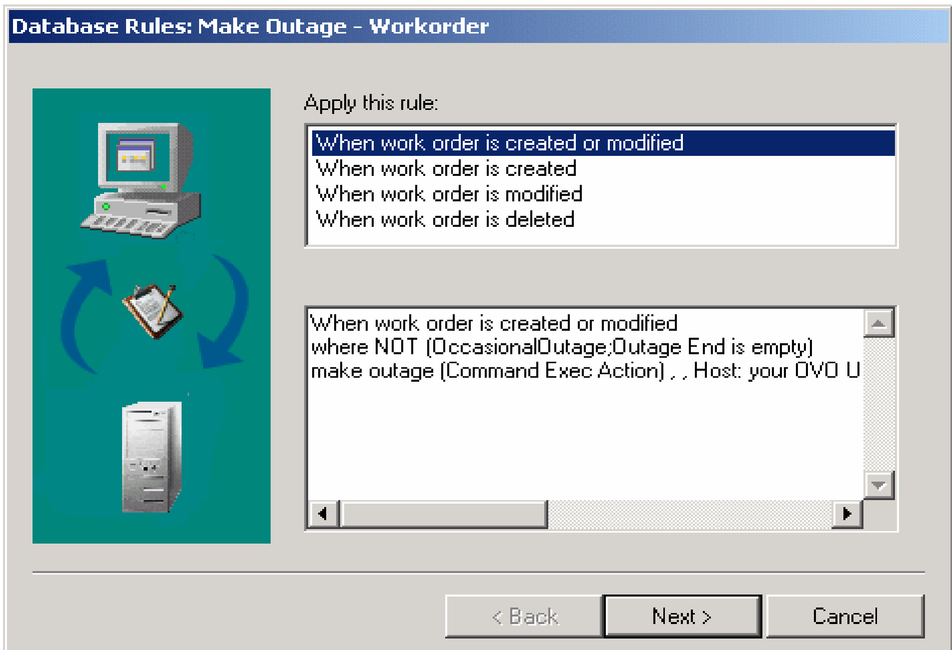
Figure 31 Final Database Wizard



Creating a Database Rule for Occasional Outage Information

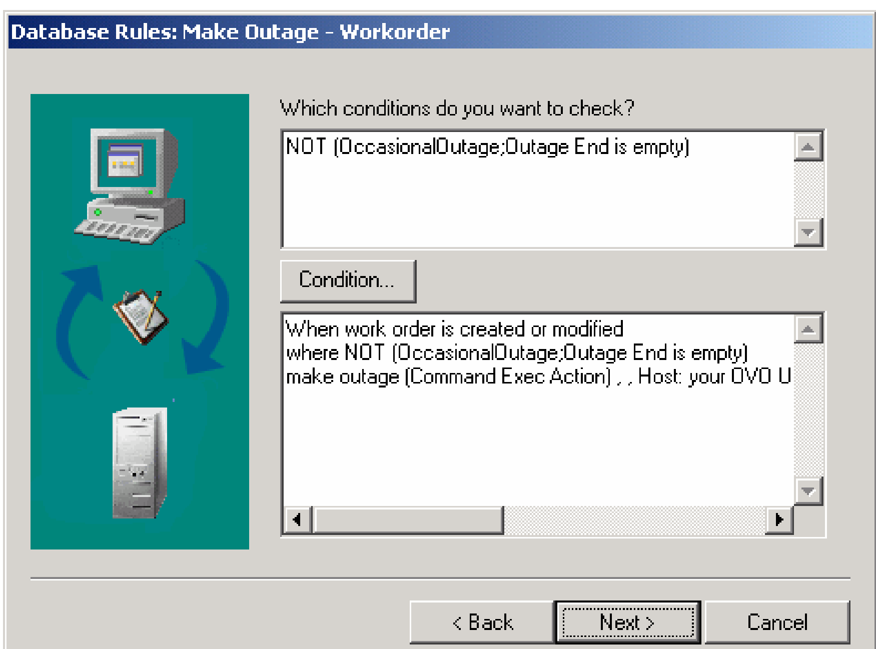
Open the Database Rule Wizard: go to the Administrator Console, select `Business Logic`, `Database Rules`, `Work Order`, and select the `mkoutage` rule if your Service Desk database already contains one or create a new one in case it does not.

Figure 32 Applying the Rule – Workorder



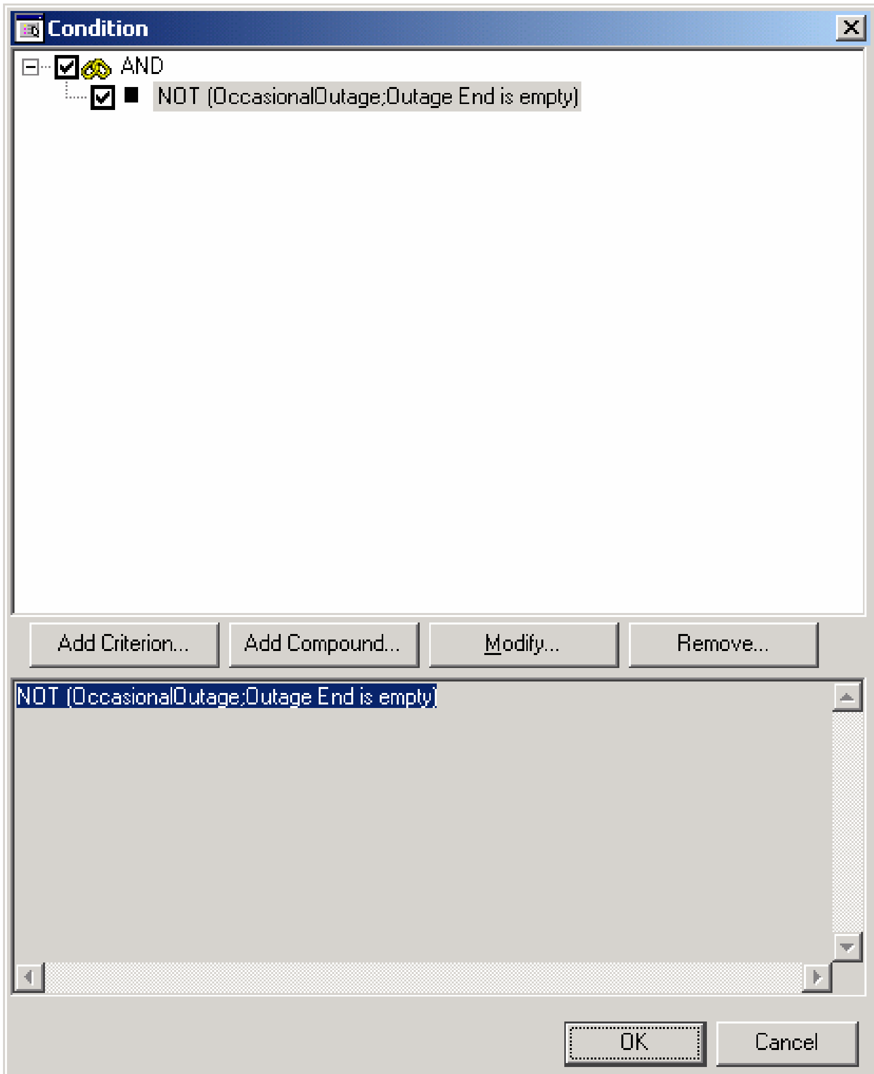
Select **When work order is created or modified**, then click **Next**.

Figure 33 What Conditions to Check



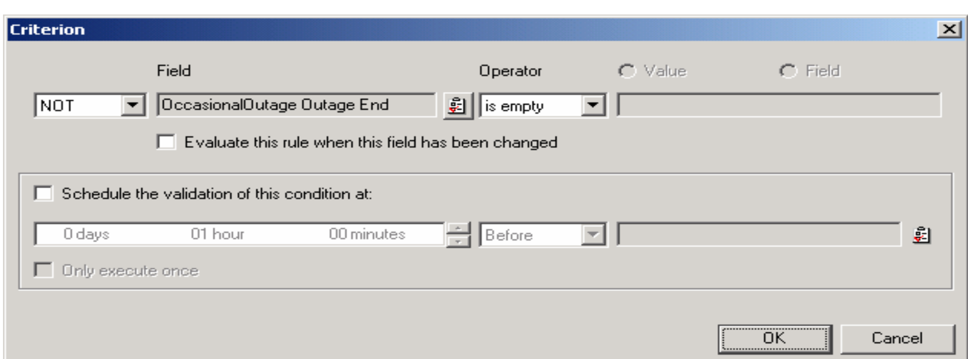
Click **Condition** to set up a new condition, the following window opens:

Figure 34 Condition Window



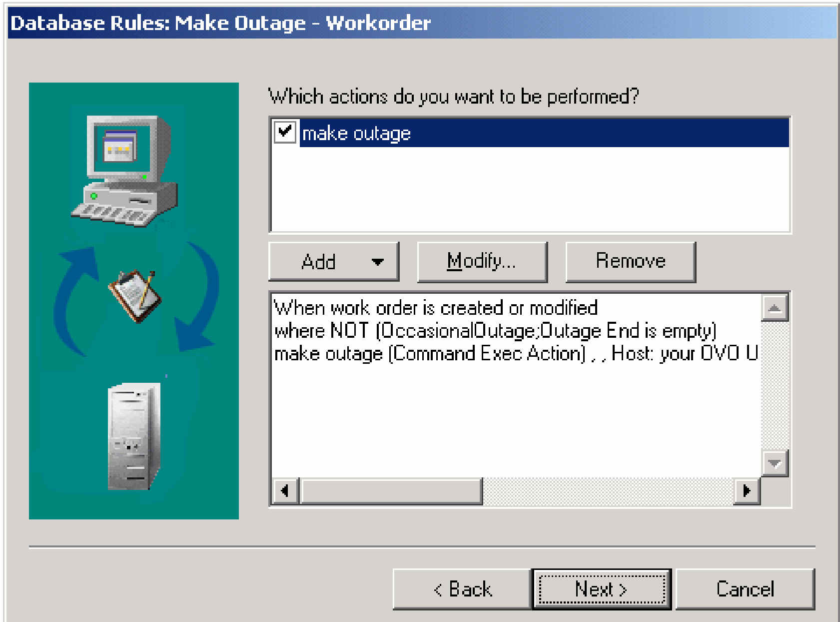
Click **Add Criterion** to define a criterion that limits the number of times the database rule is triggered. This opens a window similar to the following:

Figure 35 Criterion Definition Window



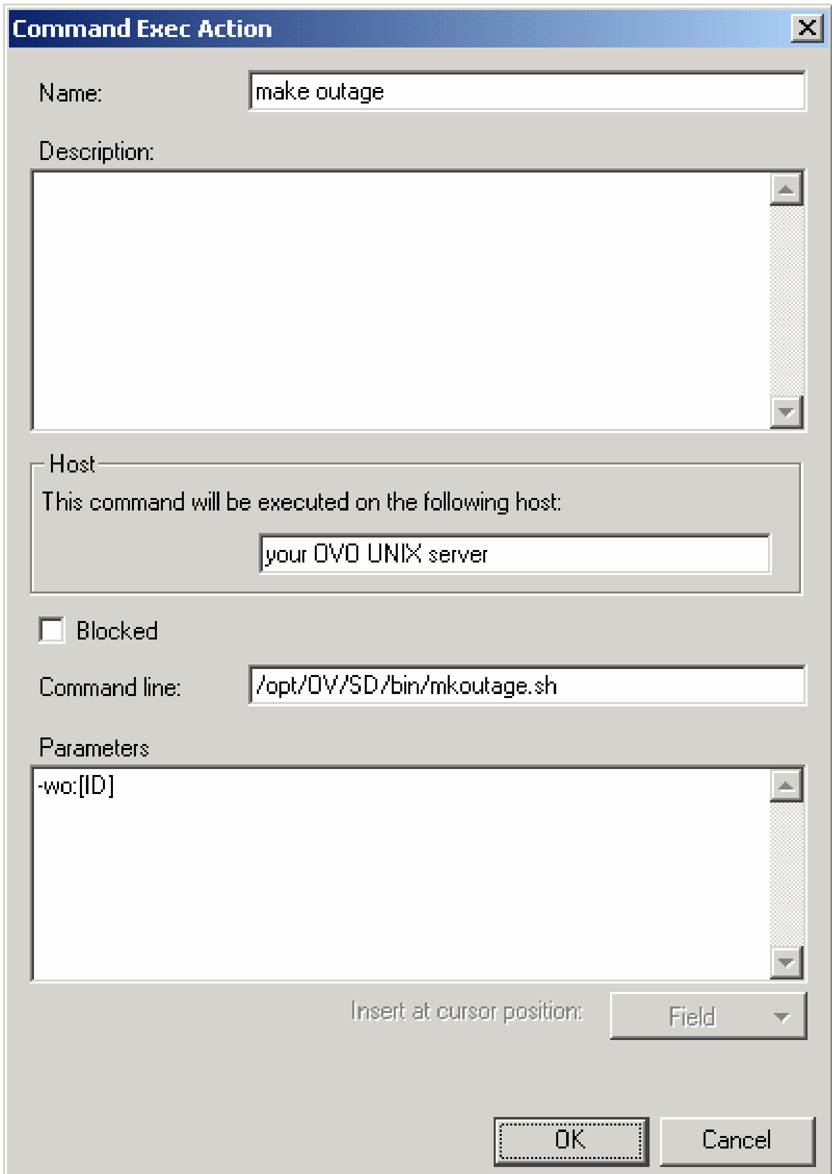
Define the criterion according to the definition seen in the screenshot above, then press OK. Follow the onscreen instructions until you get to the following window:

Figure 36 Actions to Perform



Click **Add** to define a database rule action. This will open the following window:

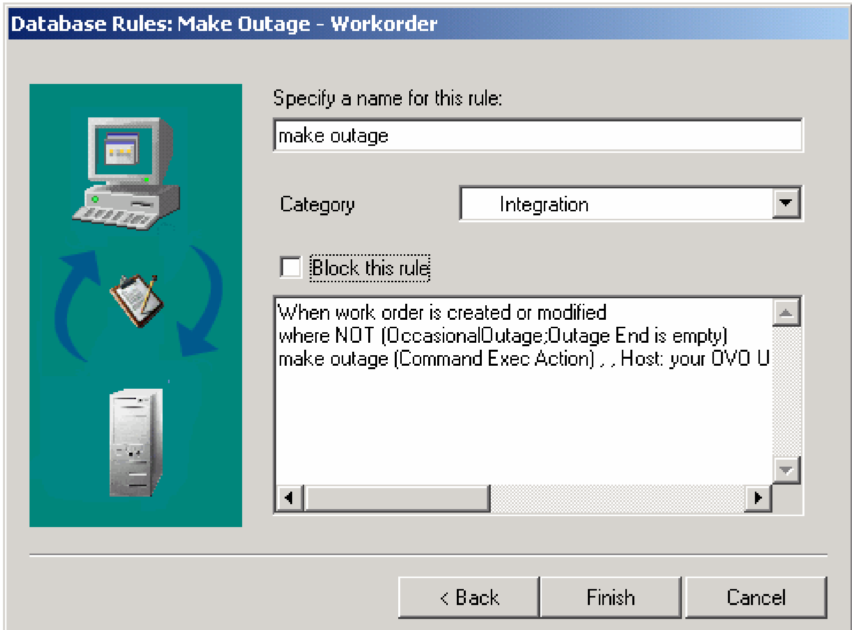
Figure 37 Command Exec Action



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 click **OK** when you are finished.

This returns you to the previous window, in which you will have to click **Next** to proceed to the final screen of the database rule wizard.

Figure 38 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.

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/ovoservices.rs
  d"
  ```

- 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 39 Changes to the ovounixservices Import Mapping: Services

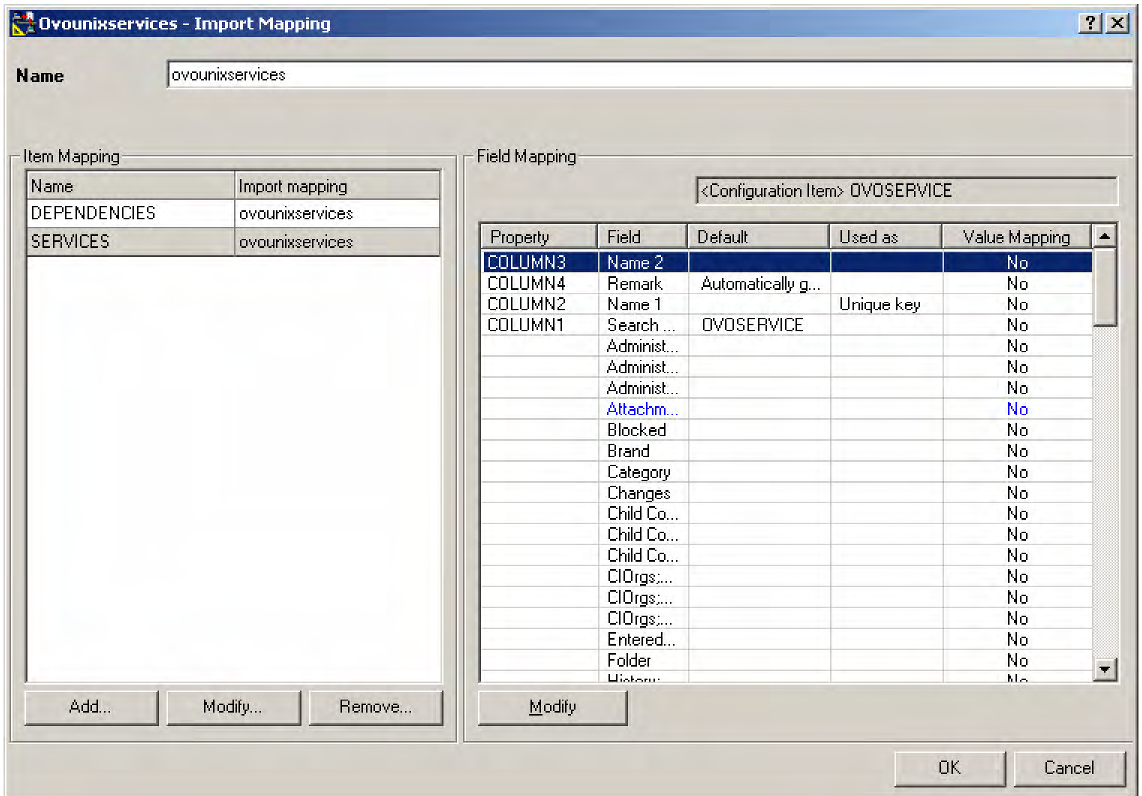
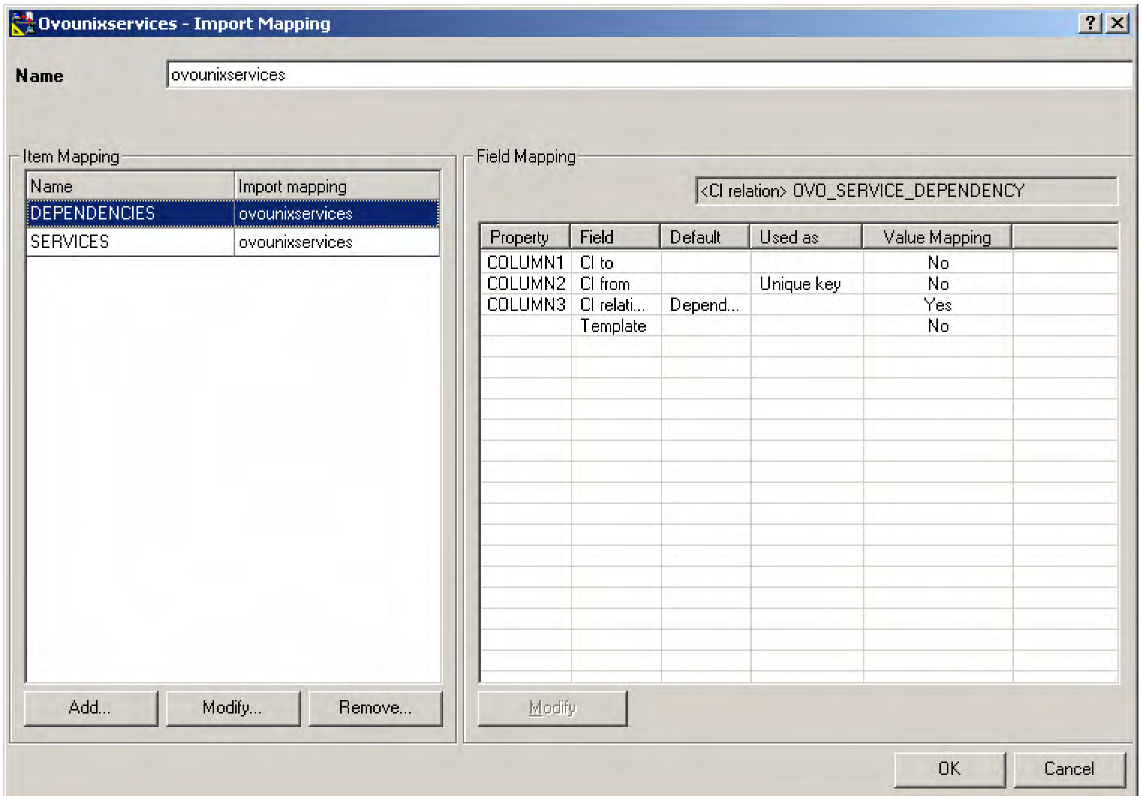


Figure 40 Changes to the ovounixservices Import Mapping: Dependencies

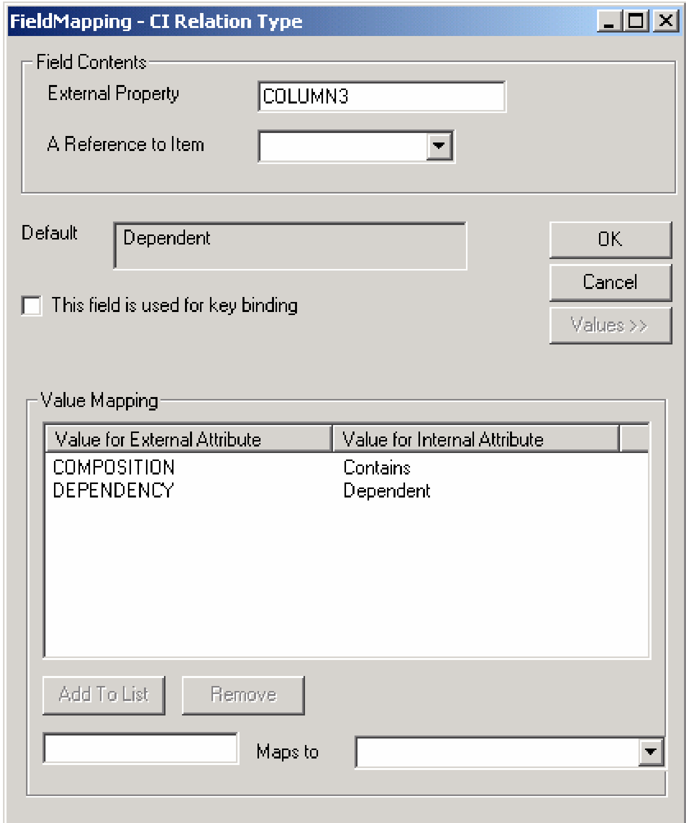


To change an existing `ovounixservices` import mapping to match the example seen above, press the Add button under the left pane. Create a new item mapping called `DEPENDENCIES`, referencing the item `CI relation` and using template `OVOSERVICE`. In the right pane, map properties according to the following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

Table 7

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

Figure 41 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.

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 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.

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.

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.

Appendix A UI Rules Example

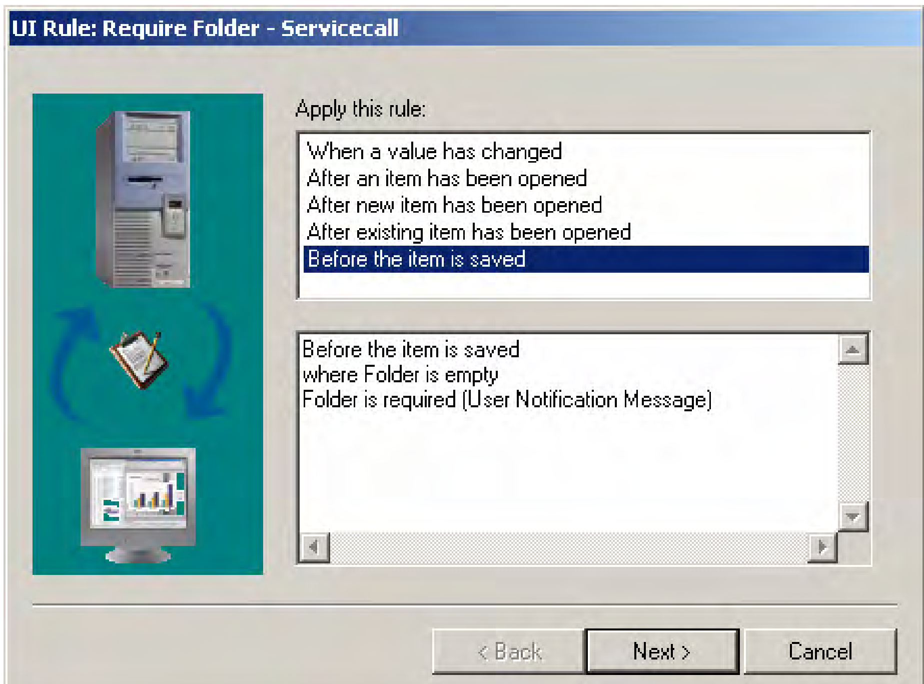
In this chapter a UI Rule is shown that forces the user to fill a field (or cancel an operation) based on the condition of other fields.

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:

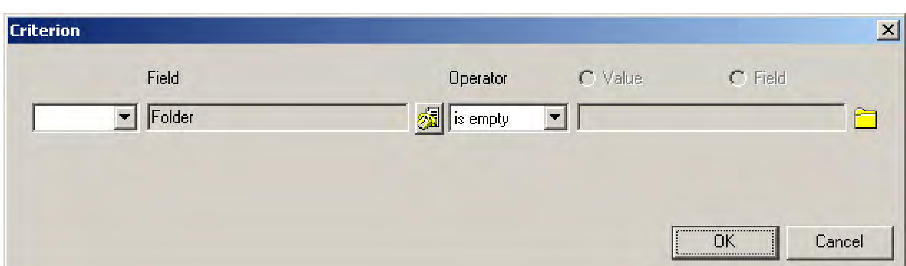
Choose the Before Save trigger:

Figure 42 Choosing the Before Save Trigger



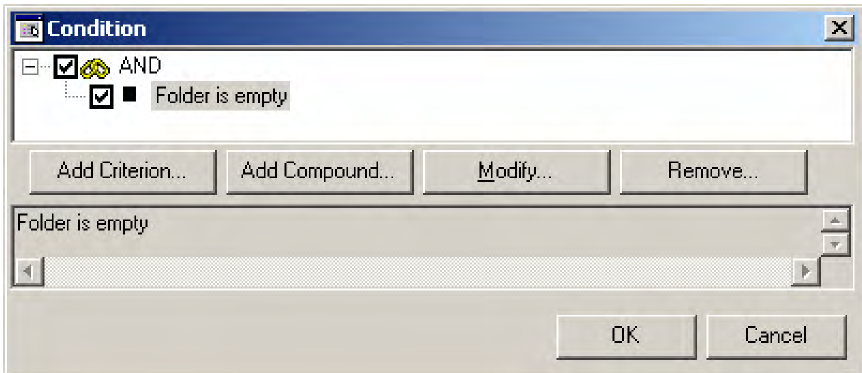
In the Condition text box, create the criterion shown in Figure 42.

Figure 43 Check for an Empty Folder



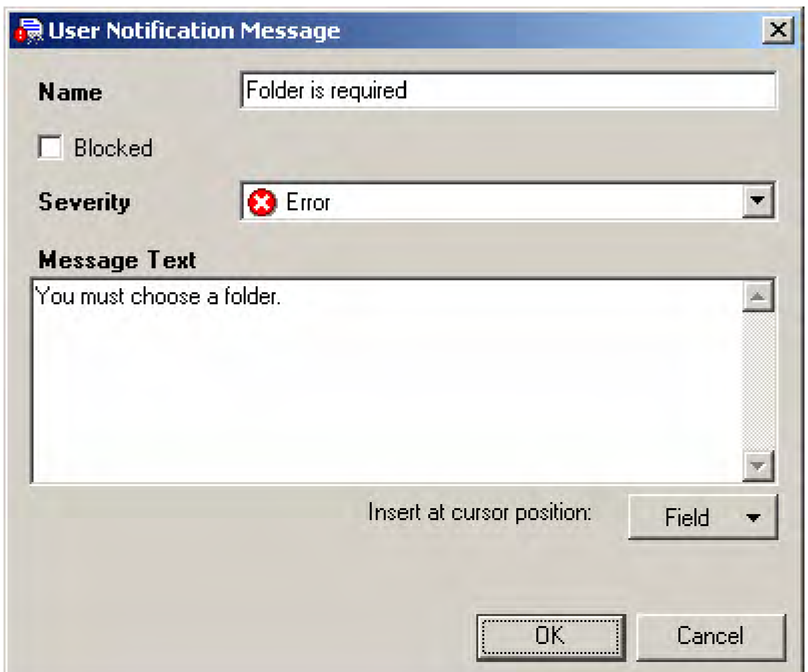
This results in the condition shown in Figure 44:

Figure 44 Folder is Empty Condition



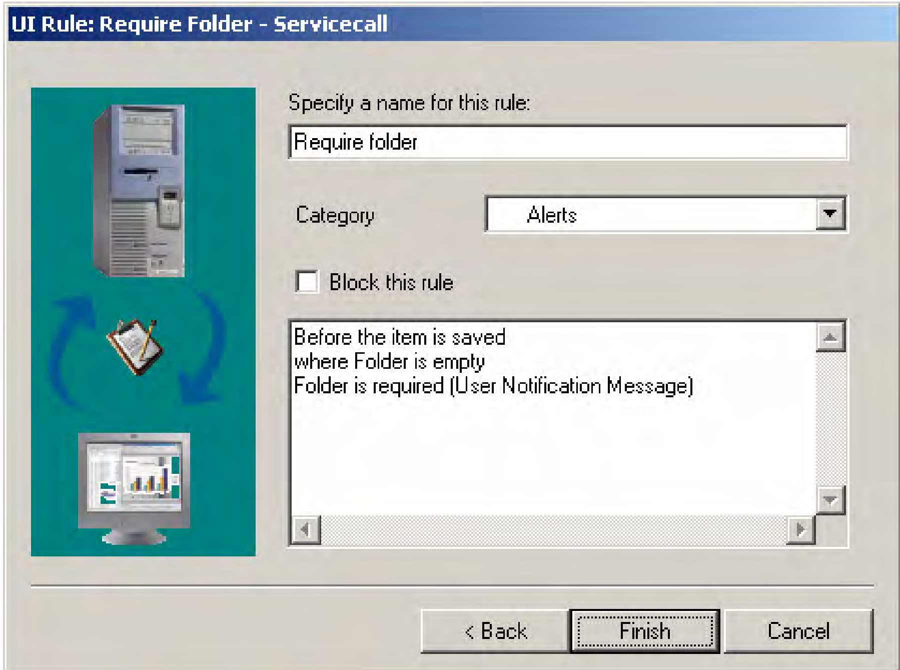
When creating the Message, make sure it has the `Error` severity (otherwise it will fail to cancel the Save action). See Figure 45:

Figure 45 Preventing the User From Saving the Form



The rule needs a name and maybe a category for locating it later on, see Figure 46:

Figure 46 Finished UI Rule



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

Figure 47 User Notification Message

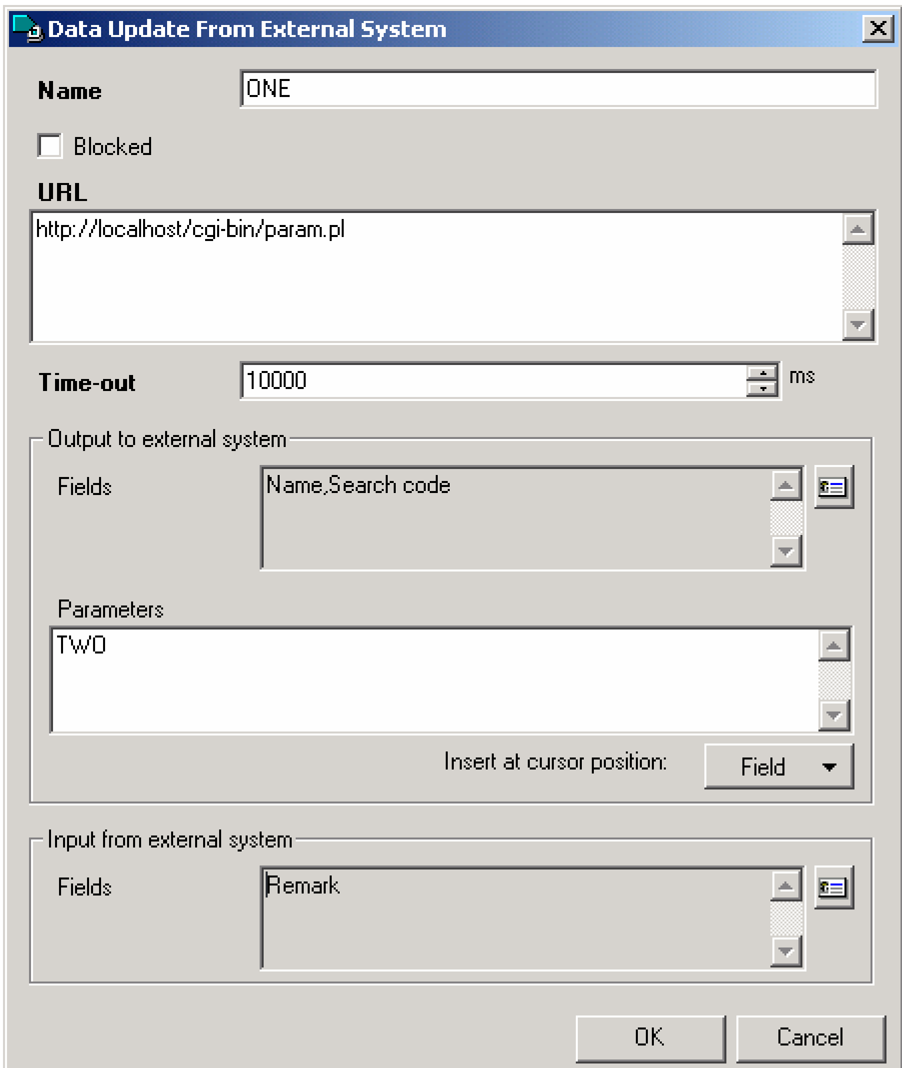
After clicking **OK**, the form is not saved and remains open.

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 in Figure 48:

Figure 48



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:

Table 9

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:

Table 10

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	`281478314066123`
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	`0.2534722222222222`

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.

Table 11

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	`281478314066123`, `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);

}
```

3 Service Pack 11

This chapter contains information previously provided with Service Pack 11.

Enhanced Auditing

Introduction

The extended auditing feature has been implemented that takes care of auditing of all administrator console items, user logon/logoff and deletion of history lines. Any user who has the auditor role assigned can configure, export and view the audit log. This auditor role can be assigned to a user by the system administrator.

Getting Started

As mentioned auditing is created so that only users with a specific role can see auditing. That is why, when the System Administrator logs in into Service desk there does not appear to be any extra functionality. The way to setup auditing is to have the System Administrator create or modify a user (i.e. Auditor) and assign the Auditor role to this user. When this user then logs in, the Auditing part of Service Desk is shown.

Initially Enhanced Auditing is paused and needs to be resumed before it is possible to configure Enhanced Auditing. Please refer to Configuration on how to resume Enhanced Auditing.

When a user who has only the auditor role opens the administrator console, the screen in Figure 1 is shown.

Figure 1

There are two items that can be opened:

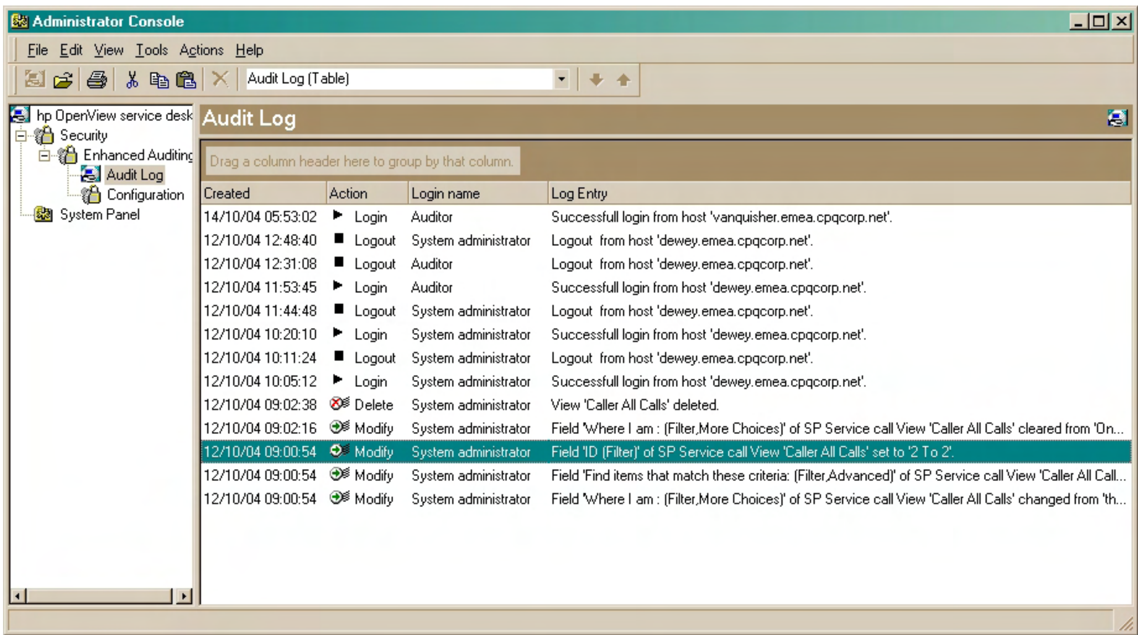
- *Audit Log*;
- *Configuration*;

Because the audit log information cannot be deleted, it is only possible to archive the audit information. See the *Archiving* section for more details.

Audit Log

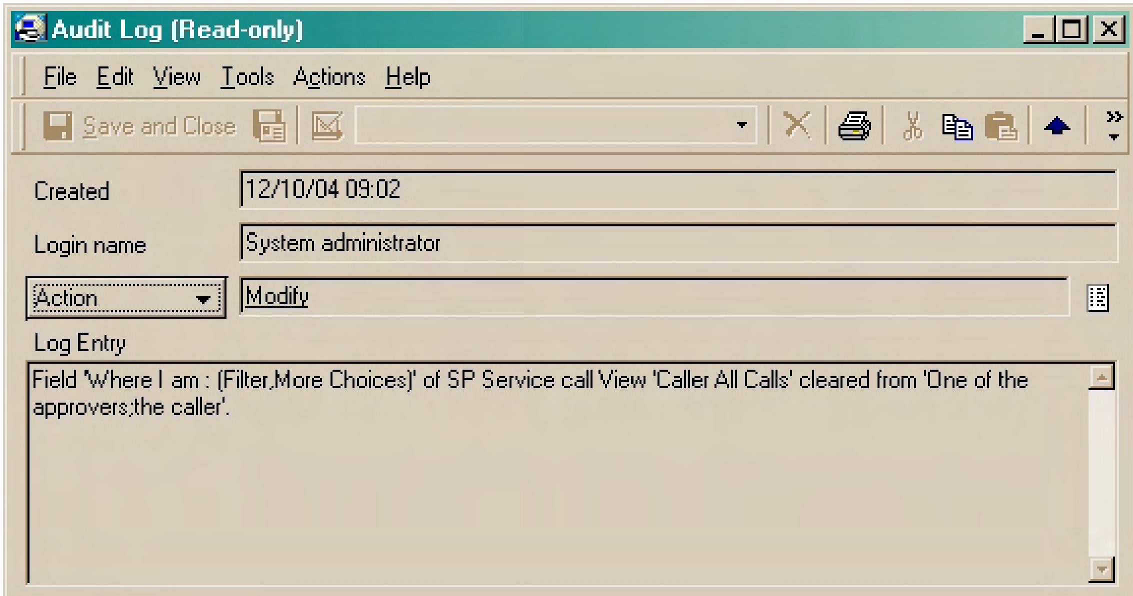
When selecting the Audit Log, the Audit Log screen is shown (see Figure 2). This window has standard Service Desk behavior. This means that columns can be sorted by clicking on their header and the view can be customized, filters can be added, etc.

Figure 2



By double clicking on one of the items, the details of the specific Audit Entry can be viewed (see Figure 3).

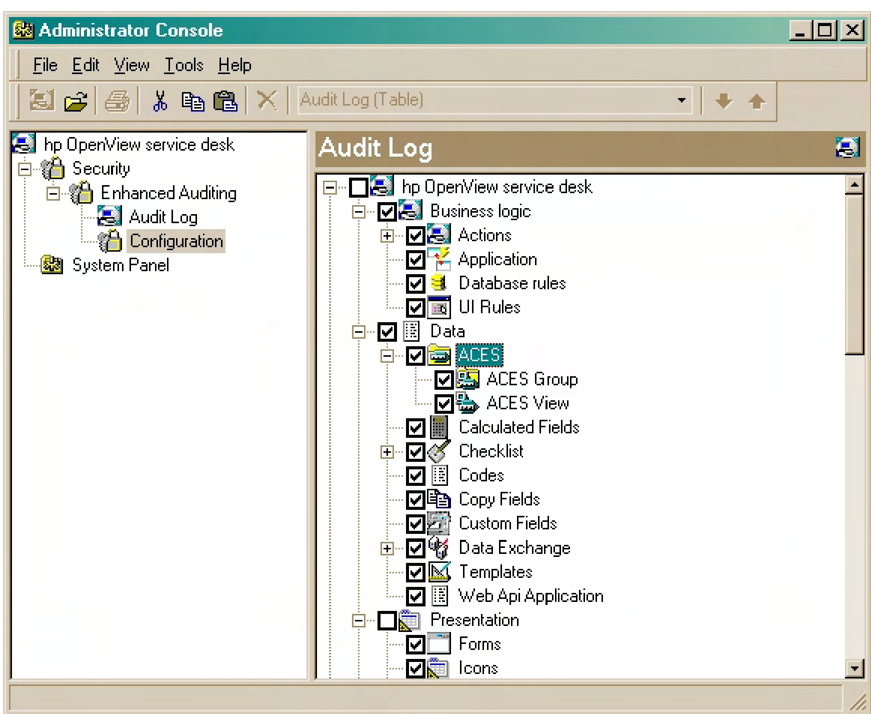
Figure 3



Configuration

Enhanced Auditing can be configured on an item by item basis. See Figure 4.

Figure 4



When right clicking on the root of the tree (hp OpenView service desk) the whole Auditing feature can be paused or resumed. See Figure 5. This is also useful when you want to halt

Auditing but actually don't want to clear your current Auditing configuration, for example when using ACES because the combination of *ACES and the Enhanced Auditing feature* can cause problems.

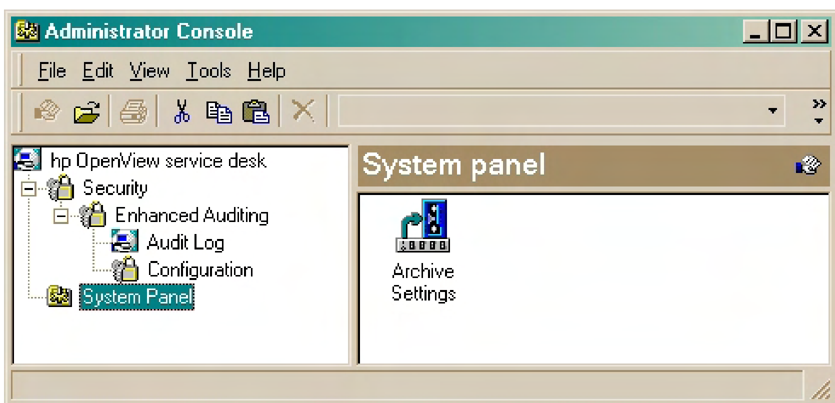
Figure 5

Deletion is never audited unless one or both of the options 'Item deletion' or 'Deletion of History Lines' is checked.

Archiving

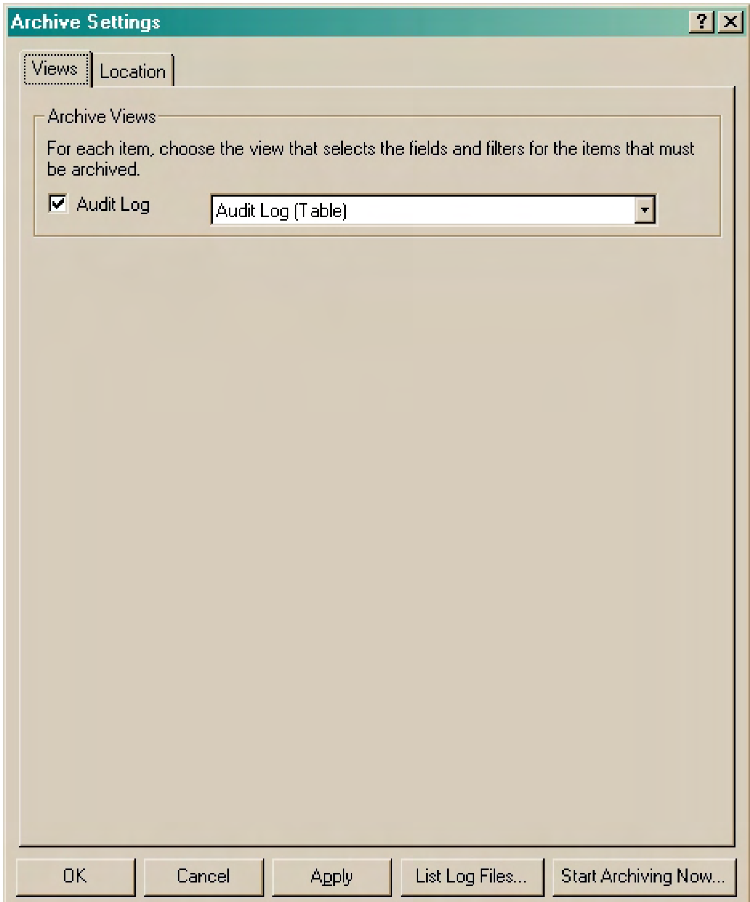
Audit entries cannot be deleted within Service Desk. They can only be archived. This is done with the Archiving functionality of Service Desk. An Auditor can open the System Panel and can select Archiving. See Figure 6.

Figure 6



When opening the Archive Settings, an Auditor can only see items related to Archiving. See Figure 7.

Figure 7



On the Location tab, the archive settings can be amended. By selecting the 'Use Subfolder For Each Item' setting the archive process logs to a specific folder which can be protected by the Operating System. Note that if a user amends any setting in this page, this setting is valid for *all items which can be archived*. By changing the 'Archive File Name' setting the Audit Log can always have a unique name.

To start archiving, click **Start Archiving Now** on the button panel. This starts archiving the views the current logged in user has access to. So if a user had Administrator rights and Audit rights, all views selected for Archiving are exported. If a user only has Auditor rights only the Audit view is archived if selected.

Known Errors or Limitations

This Auditing feature might not work in relation with ACES; it is recommended to paused auditing while performing ACES actions, and resuming this auditing afterwards.

Remote Updater

Clients updated to Service Pack 15 by remote updater will pop up with the "java.lang.NullPointerException" text when trying to open the administrator console. File `AdminConsole.atb` should be copied manually to `<client installed directory>\repo` folder.

This issue is fixed in Service Pack 16 which means the `AdminConsole.atb` will be updated automatically by remote updater.
See *ITSM007823* for more information.

Introduction

Whenever a new hot fix or service pack is released with fixes and / or features for the Service Desk console client, it has to be installed on every end user machine. In large companies different IT departments can cause delays in the distribution of a simple hot fix which usually updates no more than a few files and currently requires some sort of third party distribution mechanism or simply manually copying of the file in order to get the hot fix to the end user machine.

For service packs, the situation is similar, although it has to be said that because of the size of the service pack, the delay of installation is usually because of the test effort required to ensure that the production system can be upgraded without problems. Currently, the risk that a hot fix is required to resolve potential problems in the production environment fuels the need to perform extra testing on the service pack and the knowledge that a hot fix cannot be easily applied on all end user machines delays the implementation of a service pack even more.

Because customers are experiencing numerous difficulties keeping up with the latest fixes on the Service Desk product, it happens that when a customer is ready to upgrade to a service pack, a newer service pack is already available. This is because the implementation time between a service pack and the release of the next service pack is greater. For a customer this means that fixes for potential problems are already available, but it is practically impossible to apply a newer service pack as they just implemented a previous one. The customer then requests hot fixes for the most urgent problems, in order to bridge the gap between the release of the service pack and the implementation of that service pack.

Currently the level of effort required when distributing a hot fix in a customer environment depends on the Service Desk component that the hot fix is for. If the hot fix is only required on the application server, there is no real problem as the amount of application servers is limited, but if the hot fix is required on all the clients, the customers have great difficulties because they have to get the ‘`hotfix.jar`’ to all the client machines, of which there can be thousands. Although this can be done by copying the file, via email, or using a software distribution system, the process can be time consuming and above all error prone.

The addition of a mechanism that provides easier distribution of a hot fix and a service pack would greatly reduce the impact and existence of the above mentioned problems. Further more, it is now impossible to forget to copy or install a hot fix to the correct location, and thus ensuring that all users

Getting Started

Service Desk 4.5 will be extended in a way that enables customers to easily distribute and remove hot fixes on their clients in their environment. The main idea is that during the start up of a client console, a check is made at the application server to see if there is a new hot fix that needs to be distributed, and if so, to transfer this hot fix to the client and install it automatically.

An almost similar rule applies to client side service packs: they are downloaded automatically as soon as a new service pack is available and installed on the application server if the feature has been activated that is. If so, the current mechanism where users receive a notification screen that their version is out of date and need to upgrade by clicking on the hyperlink, is replaced by a notification that a new update has been received and that the application will be restarted. However, only the actual client `servicepack.jar` is remote updatable. Other files such as batch-files and other jars are beyond the scope of this new feature. This means that end users that require all Service Desk functionality, like export and import for example, will have to install the normal client patch, see *Known Errors and Limitations* for more details.

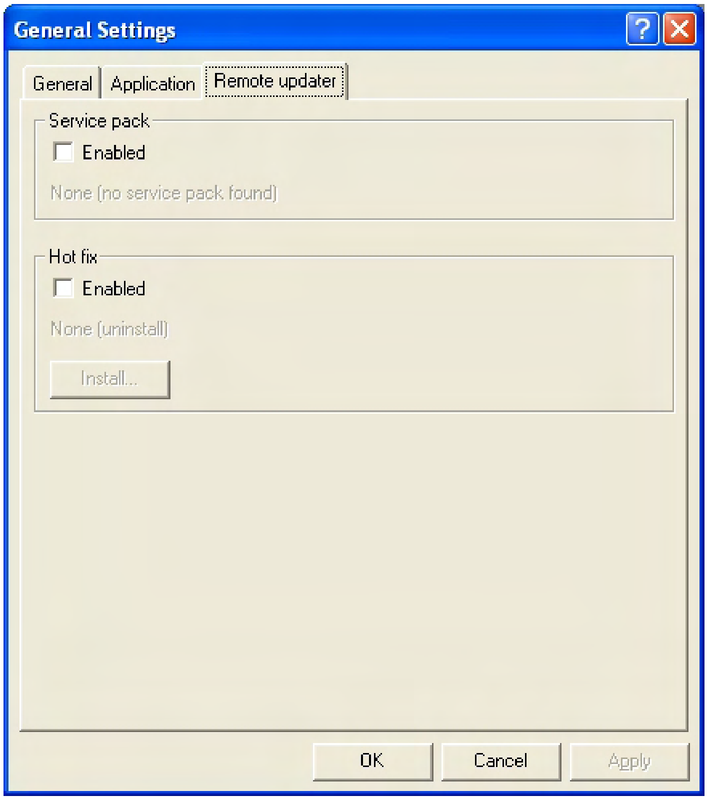
The remote updater, as the feature is called, is not a replacement for the current patch mechanism, but coexists besides it. Customers who do not want to use the feature, can still use the current patch mechanism available.

When the remote updater feature is activated, either for service packs, hot fixes or both, and a new update is available and has been downloaded to the clients local data folder, the Java virtual machine (Microsoft jview or wjview) will be "restarted". The current Java virtual machine is shutdown but just before that, a new Java virtual machine process (the client) is launched with the new update included in the class path. This is done to simulate dynamic class loading; after a Java virtual machine restart, all classes are reloaded with their newer versions if available. Dynamic class loading is a great feature of Java but unfortunately is part of the Sun Java 1.2 and higher versions. As Microsoft Java, which is still used for the client, is only Sun Java version 1.1.4 compliant, we cannot use the Sun Java 1.2 or higher features. Dynamic class loading was therefore not an option.

A hot fix does not have a version, and therefore an *MD5 checksum* is used to determine if the hot fix, if present, on the client is the same as the one installed on the application server. This way a hot fix can be overwritten with a different version, as long as the byte code of the file differs. To remove a hot fix, it is sufficient to uncheck the hot fix checkbox in **Tools** → **System** → **hp OpenView Service Desk** → **General Settings** → **Remote Updater**. This will not remove the existing hot fix file on the server, but will tell new clients that connect to the application server to remove an existing hot fix if present.

The first addition to Service Desk is the addition of an extra tab in **Tools** → **System** → **hp OpenView Service Desk** → **General Settings**:

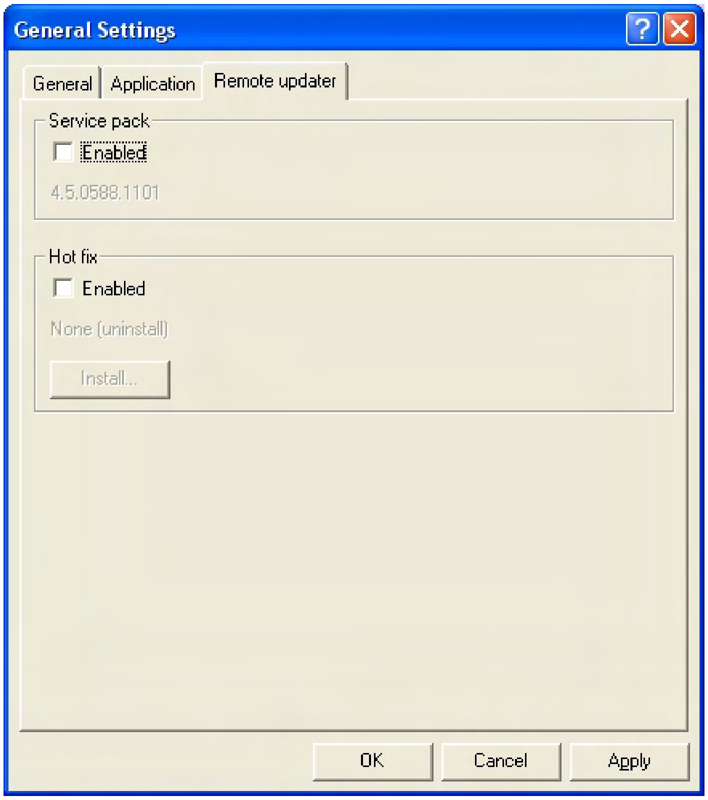
Figure 8



In this tab you will be able to activate or deactivate the remote updater functionality for a hot fix and / or service pack.

The first time you activate either the service pack or hot fix by checking the relevant check box, you will not get detailed information for the service pack or hot fix that is available, if any that is. When you activate the service pack functionality and close and reopen the **Tools** → **System** → **hp OpenView Service Desk** → **General Settings** screen you will see more detailed information about the hot fix (resolved defect numbers) or service pack (version information). This is shown below with an example for the service pack. Note that for the hot fix you will have to restart your client.

Figure 9



When the application server is updated with a service pack using the existing patch mechanism, an additional file, the service pack jar file for the client, is copied to the `<server home>/repo` folder. This file will be named `servicepack-client.jar` and is overwritten by a new client service pack jar file with every service pack that is released. When a new service pack is installed, the hot fix functionality is automatically disabled because in most cases, either the hot fix solution has been included in the service pack, or a new hot fix has to be provided by support because the hot fix needs to be recreated to work for that service pack.

Enabling the Service Pack

Enabling the service pack simply means that when a new service pack is released any client that connects to the application server and that has a lower version than the application server will receive the `servicepack.jar` for the client and restart. The application server service pack contains a `servicepack-client.jar` that is placed in the `<application server home>/repo` folder. In more detail this thus means that when clients connect the application server will send this `servicepack-client.jar` to the client if applicable, after which the client will save the file as `servicepack.jar` in the user data folder and finally the client will be restarted.

The new service pack needs to be installed using the existing patch mechanism on the application server(s) and other Service Desk components that either cannot use the remote updater functionality (e.g. agent, service pages, etc.) or require additional client updates because for example import and export features are used on that client.

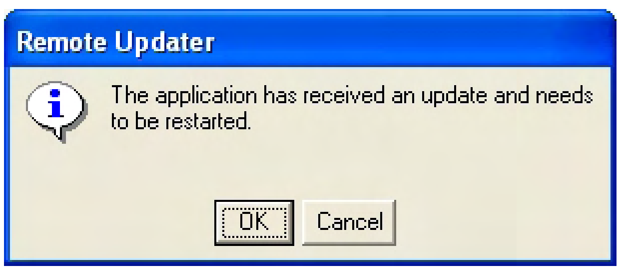
Enabling the Hot Fix

Enabling the hot fix will initially do nothing as no hot fix has been selected. This can be done be clicking the Install button and selecting the hot fix that needs to be distributed. The hot fix to be selected should be available on the client system that performs the remote updater administration. However, this is only required when initially installing or uploading the hot fix. This basically means that when the administrator selects a hot fix for distribution this hot fix is copied from the clients system to the application server that the client is connected to. The hot fix is stored on the application server in the `<application server home>/repo` folder so it is important for the application server process to be able to have write access for the `<application server home>/repo` folder.

As soon as a hot fix is uploaded to an application server this application server sends a message to other active application servers that a new hot fix is available. These application servers then retrieve that hot fix from the application server that send the message. Application server that are started after a hot fix has been distributed do not have the hot fix on startup, but as soon as a client connects to that application server, the application server requests the hot fix from other active application servers and sends that hot fix to the client as well as stores it on it's local file system.

When clients receive a new hot fix or a service pack, the following screen is shown:

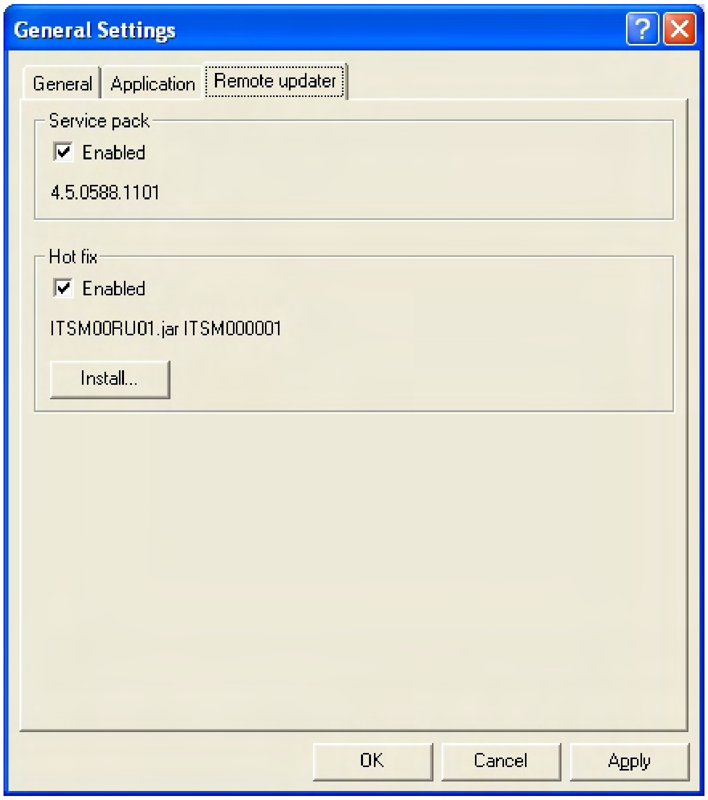
Figure 10



Clicking the **Cancel** button means that the client is closed but not restarted.

When both the service pack and hot fix remote update functionality have been enabled, the screen will look something like the one below:

Figure 11



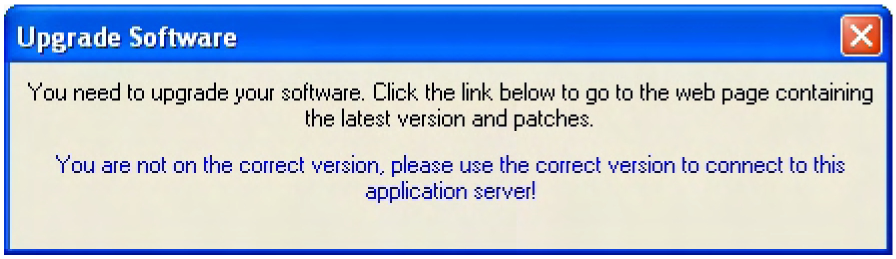
Enabling both the service pack and hot fix means that all clients that are not yet upgraded to that service pack, and that also do not have that hot fix, will receive first the hot fix after which the client is automatically restarted, and then after the version check with the application server, the service pack, after which obviously the client is again automatically restarted. This basically means that depending on which service pack the client is on, and which hot fix the client currently has, the application could be restarted automatically a couple times, because there is also the possibility of locked files. The overall time to startup a client however does not increase dramatically, except perhaps in the event of a service pack upgrade, because the version check is performed at a later stage during the logon sequence.

It will be possible only to install one hot fix at a time. A new hot fix will replace the current hot fix on a clients system.

Uninstalling a Service Pack

By clearing the service pack Enabled check box the application server no longer returns the `servicepack-client.jar` to client but as was in the past an incorrect version message. The client will then display the normal Upgrade Software message dialog as shown below:

Figure 12

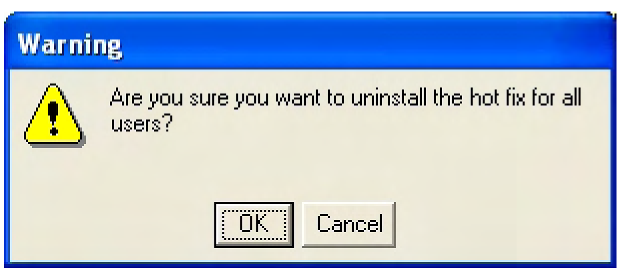


If the client is then upgraded using the normal msp patch mechanism and the client is started up, the `servicepack.jar` in the user data folder is automatically removed.

Uninstalling a Hot Fix

By clearing the hot fix Enabled check box the application server will notify all application servers that the hot fix needs to be removed from all newly connecting clients if applicable. A message dialog will pop up to indicate that the hot fix will be removed for all clients! See the screenshot below:

Figure 13



As soon as a client connects to an application server, a check is made to see if a hot fix is available. The application server responds that hot fixes should not be used after which the client removes an existing hot fix and restarts or simply continues the startup depending on whether a hot fix exists on the client.

The hot fix will not be removed from the `<application server home>/repo` folder. Be sure to check the `<application server home>/repo` folder on all application servers when applying numerous hot fixes because of potential disk space problems!

Known Errors and Limitations

Overall

As mentioned before only the actual client `servicepack.jar` is remote updatable. Other files are beyond the scope of this new feature. This means that end users that require all Service Desk functionality, like export and import for example, will have to install the normal client patch. A choice had to be made for the supported functionality this should be sufficient for a large (>95%) portion of the client install base. Client functionality that works with the remote updater is:

- client
- sd_dataform
- client accounts setting
- client support tool

Privilege Required

Write access for `<service desk user data folder>` is required to have the client updated via remote updater.

Existing Client Hotfixes

If a non-empty hot fix (a hot fix with additional files besides a manifest file) is already present in the `<client home>\lib` folder, the existing hot fix will be used instead of the hot fix that has been updated through the remote updater mechanism.

Windows 98

This feature is not supported for clients running on Windows 98 (or even lower). A short survey showed that most customers will have migrated to a newer Windows version. On top of that, technical difficulties where encountered with running command line commands on Windows 98.

Command line commands may not exceed 128 characters whereas commands in batch-files may not exceed 1024 characters. Starting the client console using `jview` or `wjview` requires the class path to be specified and even if the relative paths are used the command will still exceed 128 characters. Also, running `jview` or `wjview` from a batch-file will popup a DOS-box which will not disappear after the process has terminated.

Application Server

The application server process is required to have write access for the `<application server home>/repo` folder in order to be able to store hot fix files.

Hot fixes will not be removed from the `<application server home>/repo` folder. Be sure to check the `<application server home>/repo` folder on all application servers when applying numerous hot fixes because of potential disk space problems!

Troubleshooting

Database

The checks that the application server(s) perform during startup and runtime with regard to retrieving new service pack or hot fix information are actually queries to the database. The database contains new information about the status of the remote updater, whether it is enabled or not, and the names of the files that are available. To view this information in the database the following queries can be used:

- Retrieve the hot fix name

  ```
  select * from REP_GENERAL_SETTINGS
  where SET_OID = 281478295847001;
  ```

- Retrieve the hot fix checksum

  ```
  select * from REP_GENERAL_SETTINGS
  where SET_OID = 281478297026637;
  ```

- Retrieve the hot fix Enabled status

  ```
  select * from REP_GENERAL_SETTINGS
  where SET_OID = 281478295846999;
  ```

- Retrieve the service pack name

  ```
  select * from REP_GENERAL_SETTINGS
  where SET_OID = 281478295847002;
  ```

- Retrieve the service pack Enabled status

  ```
  select * from REP_GENERAL_SETTINGS
  where SET_OID = 281478295847000;
  ```

Client

If you have problems starting up the client it is possible to run the client with the -CLEARUPDATES option. This option will remove an existing hot fix and / or service pack from the user data folder and then proceed with the startup. Any updates available will then be retrieved from the application server. For example start the `sd_client.bat` from the `<Service Desk client home>/bin` folder using:
`sd_client.bat -CLEARUPDATES`

Application Server

In the event problems arise with either a hot fix, a service pack or both, and that cannot be resolved with starting up the client and disable the remote updater, the following database SQL statements can be used to disable the remote updater functionality:

- Disable service pack

  ```
  update REP_GENERAL_SETTINGS set SET_VALUE = 0 where SET_OID =
  281478295847000;
  ```

- Disable hot fix

  ```
  update REP_GENERAL_SETTINGS set SET_VALUE = 0 where SET_OID =
  281478295846999;
  ```

External Links

MD5: [The MD5 algorithm] takes as input a message of arbitrary length and produces as output a 128-bit "fingerprint" or "message digest" of the input. It is conjectured that it is computationally infeasible to produce two messages having the same message digest, or to produce any message having a given prespecified target message digest. The MD5 algorithm is intended for digital signature applications, where a large file must be "compressed" in a secure manner before being encrypted with a private (secret) key under a public-key cryptosystem such as RSA. Please refer to the *rfc1321.txt* specification for more details.

Test Cases

Table 1

Test case	**6537-6476-14237-1**
Setup	• Service Desk 4.5 service pack 11 • Remote updater service pack disabled • Remote updater hot fix disabled
Goal	Apply hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Check Hot fix > Enabled 4 Click Install, then browse to and select the `ITSM00RU01.jar` hot fix. 5 Close the Service Desk client 6 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000001.

Table 2

Test case	**6537-6476-14237-2**
Setup	• Situation test case 6537-6476-14237-1
Goal	Apply new hot fix over existing hot fix via the remote updater
Actions	1 Start a Service Desk client. 2 **Tools > System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater**. 3 Click Install, then browse to and select the `ITSM00RU02.jar` hot fix. 4 Close the Service Desk client. 5 Start a Service Desk client.

Test case	6537-6476-14237-2
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000002.

Table 3

Test case	6537-6476-14237-4
Setup	• Situation test case 6537-6476-14237-3
Goal	Result test case 6537-6476-14237-1 in order to check if hot fixes can be applied to an empty hot fix jar
Actions	1 Redo test case 6537-6476-14237-1
Result	See test case 6537-6476-14237-1

Table 4

Test case	6537-6476-14237-5
Setup	• Service Desk 4.5 service pack 11 build 1 • Remote updater service pack disabled • Remote updater hot fix disabled
Goal	Apply service pack via the remote updater
Actions	1 Start a Service Desk client 2 **Tools > System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Check Service Pack > Enabled 4 Close the Service Desk client 5 Stop the Service Desk application server 6 Do NOT upgrade the client to Service Desk 4.5 service pack 11 build 2 7 Upgrade the application server to Service Desk 4.5 service pack 11 build 2 8 Upgrade the database Service Desk 4.5 service pack 11 build 2 9 Start the Service Desk application server 10 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain 4.5.0588.1102 (SP11)

Table 5

Test case	6537-6476-14237-6
Setup	• Situation test case 6537-6476-14237-5

Test case	6537-6476-14237-6
Goal	Apply hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Check Hot fix > Enabled 4 Click Install, then browse to and select the `ITSM00RU01.jar` hot fix. 5 Close the Service Desk client 6 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000001.

Table 6

Test case	6537-6476-14237-7
Setup	• Situation test case 6537-6476-14237-6
Goal	Apply new hot fix over existing hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Click Install, then browse to and select the `ITSM00RU02.jar` hot fix. 4 Close the Service Desk client 5 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000002.

Table 7

Test case	6537-6476-14237-8
Setup	• Situation test case 6537-6476-14237-7
Goal	Remove existing hot fix that was installed via the remote updater

Test case	6537-6476-14237-8
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Uncheck Hot fix > Enabled 4 Click **Ok** when asked if you really want to uninstall the hot fix. 5 Close the Service Desk client 6 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should not contain ITSM000002.

Table 8

Test case	6537-6476-14237-9
Setup	• Situation test case 6537-6476-14237-8
Goal	Result test case 6537-6476-14237-6 in order to check if hot fixes can be applied to an empty hot fix jar
Actions	1 Redo test case 6537-6476-14237-6
Result	See test case 6537-6476-14237-6

Table 9

Test case	6537-6476-14237-10
Setup	• Situation test case 6537-6476-14237-9
Goal	Apply new service pack using normal patch mechanism whilst removing existing hot fix and service pack
Actions	1 Close the Service Desk client 2 Stop the Service Desk application server 3 Upgrade the client to Service Desk 4.5 service pack 11 build 3 4 Upgrade the application server to Service Desk 4.5 service pack 11 build 3 5 Upgrade the database Service Desk 4.5 service pack 11 build 3 6 Start the Service Desk application server 7 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain 4.5.0588.1103 (SP11)

Table 10

Test case	**6537-6476-14237-11**
Setup	• Situation test case 6537-6476-14237-9
Goal	Apply service pack via the remote updater whilst removing existing hot fix and replacing existing service pack
Actions	1 Close the Service Desk client 2 Stop the Service Desk application server 3 Do NOT upgrade the client to Service Desk 4.5 service pack 11 build 3 4 Upgrade the application server to Service Desk 4.5 service pack 11 build 3 5 Upgrade the database Service Desk 4.5 service pack 11 build 3 6 Start the Service Desk application server 7 Start a Service Desk client
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain 4.5.0588.1103 (SP11)

Table 11

Test case	**6537-6476-14237-12**
Setup	• Service Desk 4.5 service pack 11 • Remote updater service pack disabled • Remote updater hot fix disabled
Goal	Apply hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Check Hot fix > Enabled 4 Click Install, then browse to and select the `ITSM00RU01.jar` hot fix. 5 Close the Service Desk client 6 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000001. The dataform service call screen should appear.

Table 12

Test case	**6537-6476-14237-13**
Setup	• Situation test case 6537-6476-14237-12
Goal	Apply new hot fix over existing hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Click Install, then browse to and select the `ITSM00RU02.jar` hot fix. 4 Close the Service Desk client 5 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000002. The dataform service call screen should appear.

Table 13

Test case	**6537-6476-14237-14**
Setup	• Situation test case 6537-6476-14237-13
Goal	Remove existing hot fix that was installed via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Uncheck Hot fix > Enabled 4 Click Ok when asked if you really want to uninstall the hot fix. 5 Close the Service Desk client 6 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should not contain ITSM000002. The dataform service call screen should appear.

Table 14

Test case	**6537-6476-14237-15**
Setup	• Situation test case 6537-6476-14237-14
Goal	Result test case 6537-6476-14237-1 in order to check if hot fixes can be applied to an empty hot fix jar
Actions	1 Redo test case 6537-6476-14237-12
Result	See test case 6537-6476-14237-12

Table 15

Test case	**6537-6476-14237-16**
Setup	• Service Desk 4.5 service pack 11 build 1 • Remote updater service pack disabled • Remote updater hot fix disabled
Goal	Apply service pack via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Check Service Pack > Enabled 4 Close the Service Desk client 5 Stop the Service Desk application server 6 Do NOT upgrade the client to Service Desk 4.5 service pack 11 build 2 7 Upgrade the application server to Service Desk 4.5 service pack 11 build 2 8 Upgrade the database Service Desk 4.5 service pack 11 build 2 9 Start the Service Desk application server 10 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain 4.5.0588.1102 (SP11). The dataform service call screen should appear.

Table 16

Test case	**6537-6476-14237-17**
Setup	• Situation test case 6537-6476-14237-16
Goal	Apply hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Check Hot fix > Enabled 4 Click Install, then browse to and select the `ITSM00RU01.jar` hot fix. 5 Close the Service Desk client 6 Run sd_dataform "service call"

Test case	**6537-6476-14237-17**
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000001. The dataform service call screen should appear.

Table 17

Test case	**6537-6476-14237-18**
Setup	• Situation test case 6537-6476-14237-17
Goal	Apply new hot fix over existing hot fix via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Click Install, then browse to and select the `ITSM00RU02.jar` hot fix. 4 Close the Service Desk client 5 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain ITSM000002. The dataform service call screen should appear.

Table 18

Test case	**6537-6476-14237-19**
Setup	• Situation test case 6537-6476-14237-18
Goal	Remove existing hot fix that was installed via the remote updater
Actions	1 Start a Service Desk client 2 **Tools** → **System** → **hp OpenView Service Desk** → **System Panel** → **General Settings** → **Remote Updater** 3 Uncheck Hot fix > Enabled 4 Click Ok when asked if you really want to uninstall the hot fix. 5 Close the Service Desk client 6 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should not contain ITSM000002. The dataform service call screen should appear.

Table 19

Test case	6537-6476-14237-20
Setup	• Situation test case 6537-6476-14237-19
Goal	Result test case 6537-6476-14237-17 in order to check if hot fixes can be applied to an empty hot fix jar
Actions	1 Redo test case 6537-6476-14237-17
Result	See test case 6537-6476-14237-17

Table 20

Test case	6537-6476-14237-21
Setup	• Situation test case 6537-6476-14237-20
Goal	Apply new service pack using normal patch mechanism whilst removing existing hot fix and service pack
Actions	1 Close the Service Desk client 2 Stop the Service Desk application server 3 Upgrade the client to Service Desk 4.5 service pack 11 build 3 4 Upgrade the application server to Service Desk 4.5 service pack 11 build 3 5 Upgrade the database Service Desk 4.5 service pack 11 build 3 6 Start the Service Desk application server 7 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain 4.5.0588.1103 (SP11). The dataform service call screen should appear.

Table 21

Test case	6537-6476-14237-22
Setup	• Situation test case 6537-6476-14237-21
Goal	Apply service pack via the remote updater whilst removing existing hot fix and replacing existing service pack

Test case	6537-6476-14237-22
Actions	1 Close the Service Desk client 2 Stop the Service Desk application server 3 Do NOT upgrade the client to Service Desk 4.5 service pack 11 build 3 4 Upgrade the application server to Service Desk 4.5 service pack 11 build 3 5 Upgrade the database Service Desk 4.5 service pack 11 build 3 6 Start the Service Desk application server 7 Run sd_dataform "service call"
Result	During startup a message appears that a new update has been received and that the application needs to be restarted. The version in **Help** → **About** should contain 4.5.0588.1103 (SP11). The dataform service call screen should appear.

Table 22

Test case	6537-6476-14237-23
Setup	• See unit tests 1-11
Goal	Run unit tests 1-11 using the /settings parameter
Actions	1 Move(!) the existing /settings parameter located by default in: `C:\Documents and Settings\<user name>\Application Data\Hewlett-Packard\OpenView\Service Desk` 2 Instead of running the client without parameters, use the /settings parameter and point to the moved user_settings.xml location, for example: 3 `sd_client "/settings=c:\temp\user_settings.xml"`
Result	See unit tests 1-11

Table 23

Test case	6537-6476-14237-24
Setup	• See unit tests 12-22
Goal	Run unit tests 12-22 using the /settings parameter
Actions	1 Move(!) the existing /settings parameter located by default in: `C:\Documents and Settings\<user name>\Application Data\Hewlett-Packard\OpenView\Service Desk` 2 Instead of running sd_dataform without parameters, use the /settings parameter and point to the moved `user_settings.xml` location, for example: 3 `sd_dataform "/settings=c:\temp\user_settings.xml" "service call"`

Test case	**6537-6476-14237-24**
Result	See unit tests 12-22

Frequently Asked Questions

Q: What is the remote updater?
A: A new feature for clients which allows automatic updates of the client using a hot fix and /or a service pack without user interaction.

Q: Will the remote updater be a replacement for the current patch mechanism?
A: No, the remote updater will not be a replacement for the current patch mechanism. The remote updater can only be used in combination with the Service Desk client component, and has an additional limitation because it will only be a replacement for the `hotfix.jar` and / or `servicepack.jar`, if used that is. Please refer to *Known Errors and Limitations* for additional information on this.

Q: Is it possible to use the remote updater along side the current patch mechanism? So are you able to install the normal client msp patch on a set of clients, and use the remote update functionality to update a different set of clients?
A: Yes! First of all, it is not required to use the remote updater. Second, which is actually more important, you will have to upgrade certain clients using the msp patch because of the reasons described in answer 2.

Q: Will it be possible to upgrade only a selective range of clients using the remote updater?
A: No, but yes at the same. No because no option will be offered to select a selection of clients or users that can use the remote updater. And yes because you will always have the option to use the current patch mechanism. For example, clients that are on the same service pack version as the application servers will not receive the remote updater service pack update.

Q: Will it be possible to downgrade to a lower Service Desk version?
A: No, this will in almost all cases not be possible. Why? Because during a service pack upgrade, the database is also upgrade and in almost all cases there are updates done to the database that cannot be easily rolled back because they add or change a certain feature in such a way that a roll back is simply not possible. With the exception off course if you still have a database dump that was created prior to upgrading to a newer service pack. In the future, all database statements that will be applied to the database during an upgrade will be included in the service pack documentation, including a statement whether or not it will be possible to downgrade to the previous service pack version. Note. With previous service pack version the version that was released prior to the installed service pack version is meant, which is not necessarily the version that you have!

Q: Is Windows Terminal Services supported?
A: No, to support Windows Terminal Services the best and easiest way is to upgrade the client using the current patch mechanism, as all clients will then be upgraded in one go. The same goes for applying a hot fix.

Q: Is it possible to install more than one hot fix?
A: No, this is not possible. An existing hot fix will be replaced. Note that a hot fix that was copied in the client lib folder, as has been the case and is the case if you do not use remote updater, that hot fix will be used, and not the hot fix in the users data folder!

Q: How can you upgrade a new and fresh client installation?
A: First the new client should have to be installed with at least the official service pack 11 or higher regular msp patch. The easiest way to do this is create a new Client installation package that includes the latest official service pack patch. See the Windows Installation documentation for details. After that the client is remote updater prepared and can receive service packs and / or hot fixes using the remote updater feature.

Q: Is it possible to use this feature when user has limited access on their systems, e.g. a so-called locked down system?
A: Yes, that is if the users have normal write capabilities in their user data folder, e.g. `<documents and settings>\Application Data\Hewlett-Packard\OpenView\Service Desk`. This is where Service Desk client cache is stored, and is also the location for the hot fix and service pack that are retrieved from the application server.

4 Service Pack 12

This chapter contains information previously provided with Service Pack 12.

Importing Nodes and Services into Service Desk 4.5

Introduction

Since Microsoft Windows 2003 is introduced the integration between OpenView Operations for Windows (OVOW) and Service Desk is broken. The reason for this is that the WBEM ODBC Driver that is used in the integration between Service Desk and OVOW is not supported anymore by Microsoft on the Windows 2003 platform. So we need a new tool to export OVO for Windows CI and Services data, and import them into Service Desk. This tool should work on all Windows platforms.

Resolution

HP has provided this paper along with a new binary and several configuration files, to replace the WBEM ODBC Adapter on all Windows platforms and allow Service Desk to directly access the OVO for Windows CI and Services data via the Microsoft SQL Server ODBC Adapter.

The solution holds the following files:

- A readme file
- `OVSDSync.exe`
- Create tables in the OVO SQL Server database on the OVOW machine.
- `OVSDSync.exe.config`
- Configuration file for `OVSDSync.exe`.
- `ovowindowsci.ini`
- Configuration file that is used for the Service Desk Data Exchange Export functionality
- `ovowindowsservices.ini`
- Configuration file that is used for the Service Desk Data Exchange Export functionality

The OVSDSync.exe Tool

`OVSDSync` connects to the OVOW WMI Providers, extracts the required OVOW objects and places them in a set of newly created SQL tables.

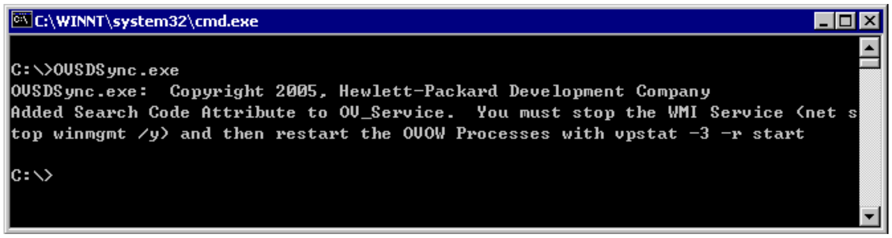
`OVSDSync.exe` uses the Microsoft .NET framework 1.1, and runs on the OVOW Management Server. You must run the `OVSDSync.exe` tool prior to export so any changes made to the OVOW Model carries over in the SQL Tables accessed by OVSD.

Windows 2003 comes with the .NET runtime; you do not need to install anything on Windows 2003. For Windows 2000, you can download the .NET runtime from:

http://www.microsoft.com/downloads/details.aspx?FamilyID=262d25e3-f589-4842-8157-034d1e7cf3a3&displaylang=en

`OVSDSync.exe` is a command-line tool. You should execute this tool from a command prompt. If it's the first time you run it, you will see the output as stated below:

Figure 1



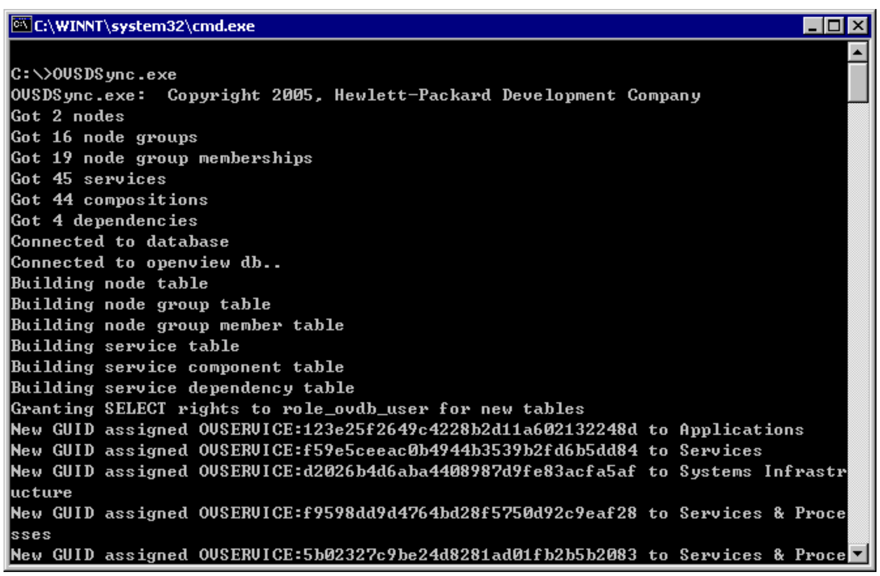
Please follow the below steps as prompt:

1 Stop the WMI Service using command: < `net stop winmgmt /y`>;

2 Restart the OVOW Processes using command: < `vpstat -3 –r start`>.

Then run `OVSDSync.exe` again, you will see the output as stated below:

Figure 2



If `OVSDSync.exe` keeps prompt you to stop WMI Service and restart OVOW Processes, please restart your computer.

When `OVSDSync.exe` is executed, it creates the following tables in the database and populates them for Service Desk Data Exchange to extract from;

- OV_SD_OW_SERVICE
  All information (copy) from the OV_Service table of OVOW.
  These are the OVOW Services.

- OV_SD_OW_SERVICE_COMPOSITION
  All information (copy) from the *OV_ServiceComposition* table of OVOW. These are the OVOW "Contains" relationships.
- OV_SD_OW_SERVICE_DEPENDENCIES
  All information (copy) from the *OV_ServiceDependency* table of OVOW. These are the OVOW "Contains" relationships.
- OV_SD_OW_NODE
  All information (copy) from the *OV_ManagedNode* table of OVOW. These are the OVOW Nodes.
- OV_SD_OW_NODEGROUP
  All of the OVOW node groups (not used by OVSD, but available for use)
- OV_SD_OWNODEGROUPNUMBER
  All of the OVOW node group membership relations (not used by OVSD, but available for use)

Once the tool has created the tables, the tool populates the tables with the OVOW Nodes and Services data; allowing you to perform a data exchange from the OVSD application.

Export: System DSN

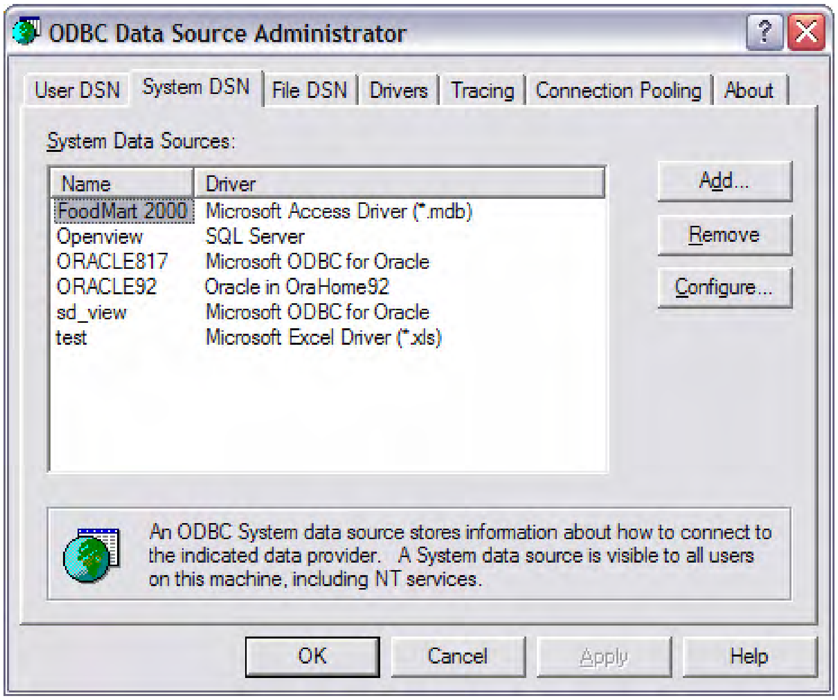
The Service Desk client needs to be configured to gather data from the newly created OVOW database tables. For this setup, the "Openview" System DSN has to be created on the machine where the Data Exchange export will be performed.

Only perform the System DSN steps if Service Desk is on another machine remote to OVOW. If OVOW and OVSD are located on the same machine and there is already an existing DSN called Openview, simply use that existing one and do not change the existing openview DSN.

When there is no System DSN called Openview, the following can be done to add a system DSN:

Start → Settings → Control Panel → Administrative Tools → Data sources (ODBC)

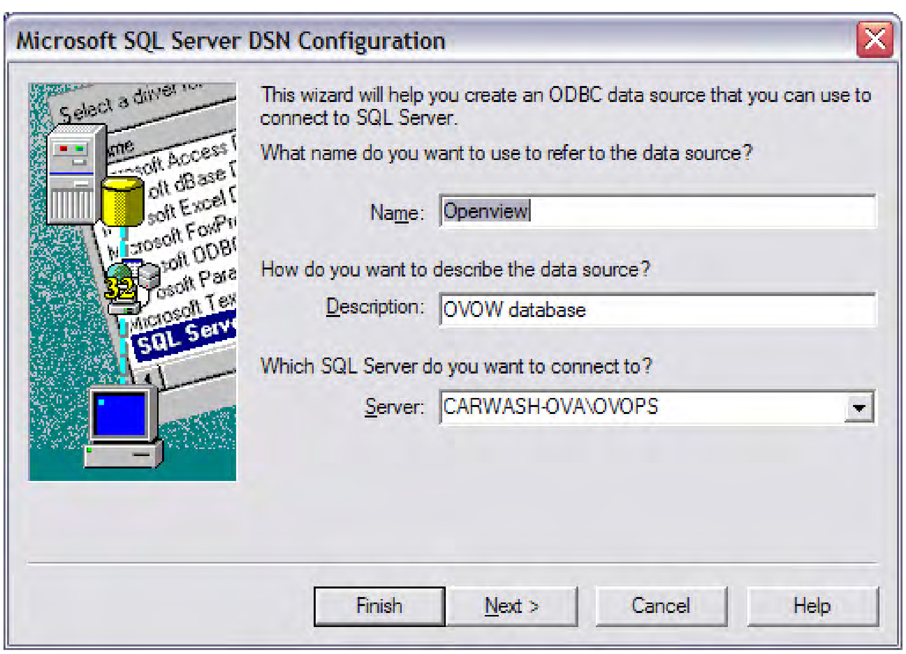
Figure 3



Click **Add** and select the SQL Server Driver followed by **Finish**.

The following screen will be visible:

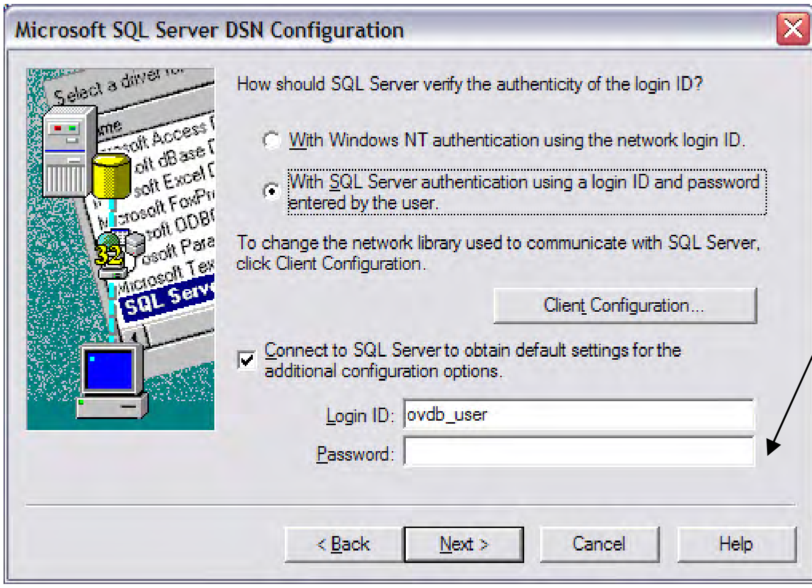
Figure 4



The server is the machine where OVOW is running.

Click **Next**.

Figure 5

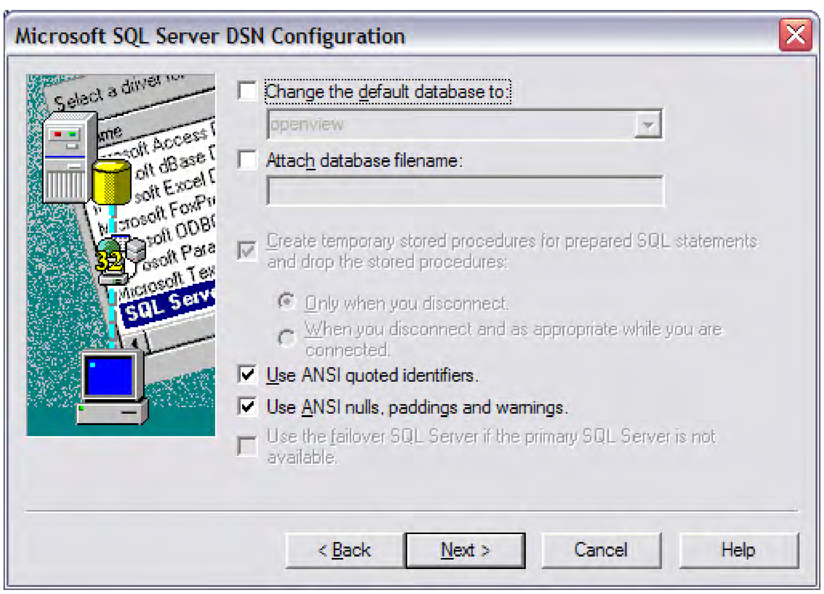


This is either: "openview" or "user"

Fill in the password (openview) followed by **Next**.

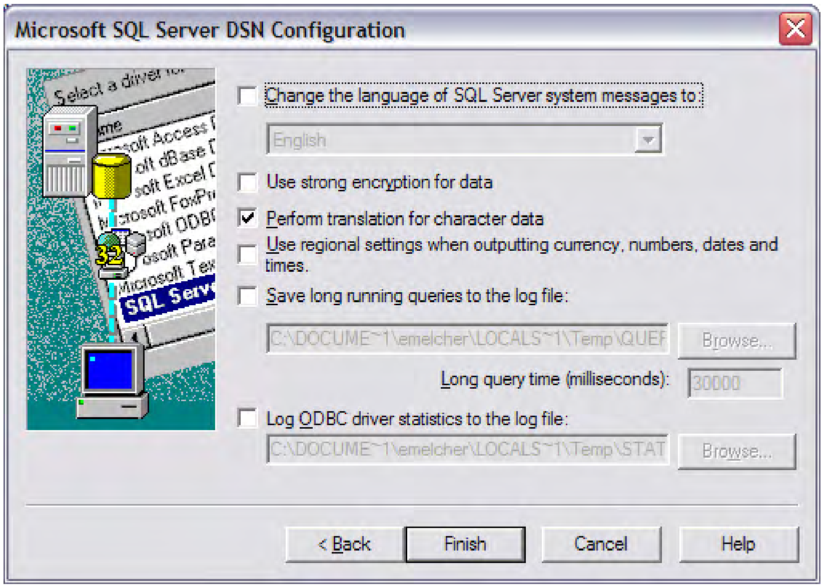
The following screen will be visible:

Figure 6



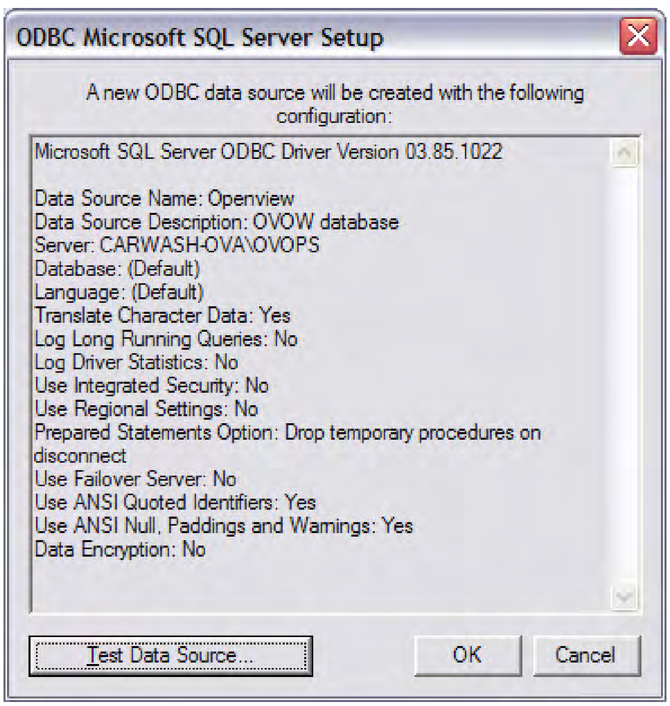
Click **Next**.

Figure 7



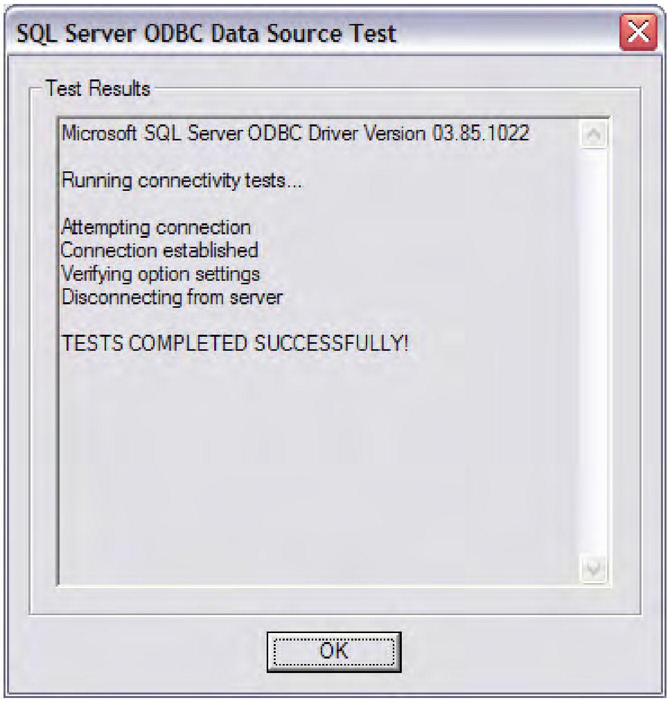
Click **Finish**.

Figure 8



Clicking **Test Data Source** gives the following screen:

Figure 9



This ODBC connection is going to be used by the Data Exchange Export functionality of Service Desk.

Export: Configuration Files

The original configuration files are completely based on the usage with WBEM ODBC Driver and therefore the configuration files are changed. The files are called “ovowindowsci.ini” and “ovowindowsservices.ini” and are written out in Appendix A.

The new files are to be copied to the locations:

- `<SD ROOT>\client\data_exchange\config`
- `<SD ROOT>\server\data_exchange\config`

When the files are copied they are directly available for use by the Service Desk Data Exchange Export functionality.

When defining your Data Exchange Tasks, you need to use the ini-files ovowindowsci.ini and ovowindowsservices.ini for the export of data.

The ini-files contain by default the password “openview” but when the password ”user” is filled in on page 8, please change the ini-files by changing the password to user:

```
[DSN]
NAME=Openview
USR=ovdb_user
PWD=openview => <change to user>
```

ACES: Import of Import Mappings

Due to the fact that a different ODBC Driver is used, the results given from this ODBC Driver is different. The WBEM ODBC Driver returns the values as they are, but the SQL Server Driver returns the results in uppercase. As a result of this the import mappings need to be changed.

This can be done by hand, but Service Desk has also functionality that makes it possible to export and import system panel attributes. This functionality is called ACES.

How the ACES functionality works can be found in the user guide of Service Desk 4.5 which is delivered with the product.

This fix also includes an ACES xml-file what makes it possible to import the import mappings that are changed because of this.

Before doing an ACES import, please be sure to make a copy op the existing import mappings.

To make use of this functionality the file `ACES_OVO_Mappings.xml` needs to be copied to the directory `<SD ROOT>\client\data_exchange\xml`.

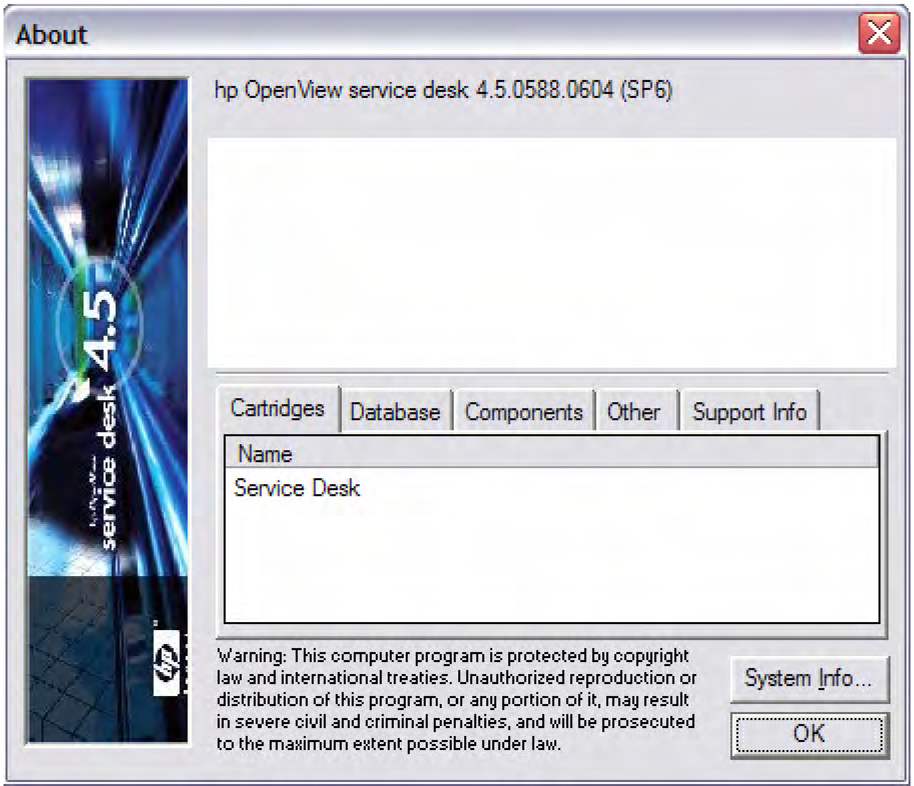
The working of the ACES functionality depends on the version of the Service Desk product. Therefore the ACES-file called `ACES_OVO_Mappings.xml` needs to be changed to the version of the client. In the xml-file the following line needs to be changed in order to make the import work for your version:

```
<ACES Count="1" GenerationDate="December 1, 2004 12:27"
SysVersion="4.5.0588">
```

to

```
<ACES Count="1" GenerationDate="December 1, 2004 12:27"
SysVersion="<SD VERSION>">
```

Change the <SD VERSION> to the version you are using. This can be found in the menu **Help** → **About** in your client application (use in this case 4.5.0588.0604 (SP6)):



The import mappings can then be imported though ACES in the following way:

Tools → **System** → **File menu** → **ACES** → **ACES Import wizard**

Import: Import Mappings

The import mappings delivered with Service Desk are created with the WBEM ODBC Driver in mind. In stead of the WBEM ODBC Driver we now use the SQL Server ODBC Driver. This driver returns the results in uppercase. As a result of this the field mappings of the import mappings also needs to be defined in uppercase.

The following import mappings are changed by changing the Field Mapping properties to uppercase:

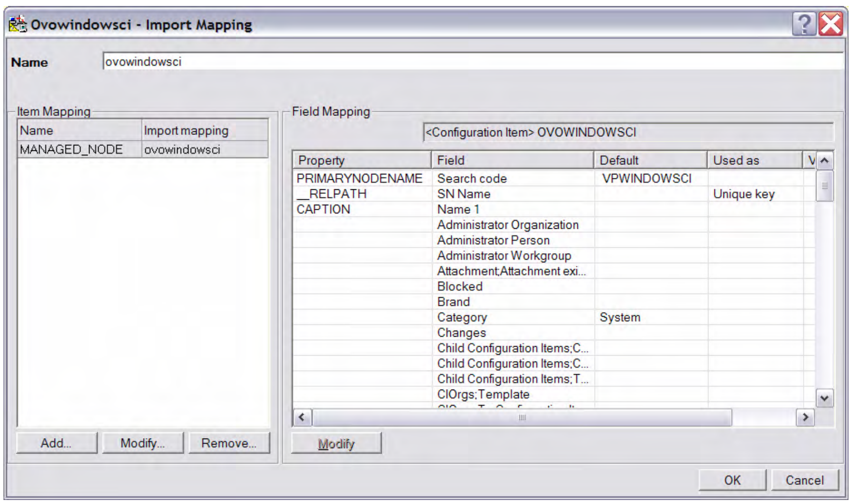
- ovowindowsci
- ovowindowsservices

There is a possibility that during the import some warnings are thrown regarding the location. This is because location can be set as a required field in Service Desk. This can be adjusted by setting the "location" field optional for CI's by removing it from the required fields list in the system panel: **Tools** → **System** → **Security** → **Prevention** → **Required Fields** → **Configuration Items**, for the CI Import to work.

Another way to work around this issue is to give them a default location by adding a value to the "Location" field in the template: **Tools** → **System** → **Data** → **Templates** → **Configuration Item** → **<The template that is used during the import>** → **field "Location"** should have a value.

Mapping: ovowindowsci

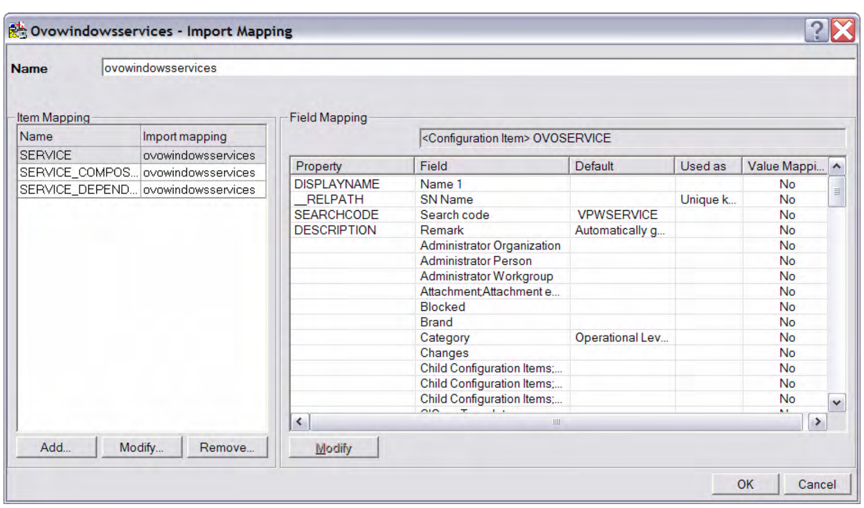
Figure 10



Mapping: ovowindowsservices

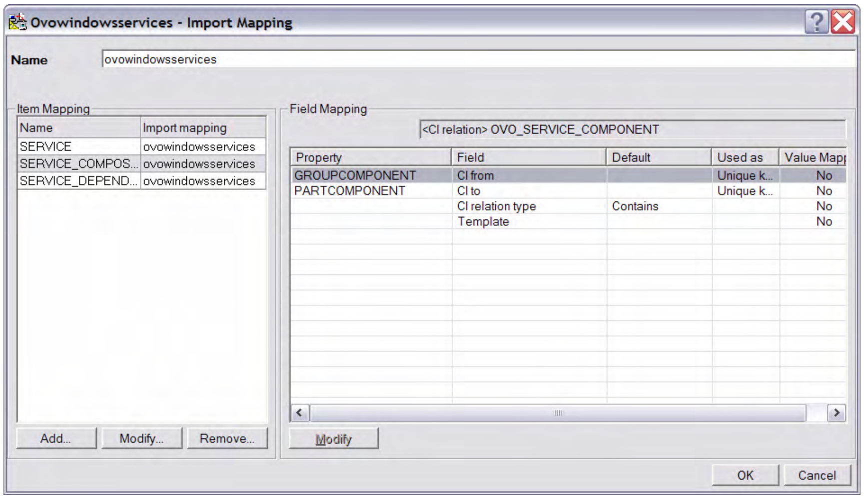
Ovowindowsservices: service

Figure 11



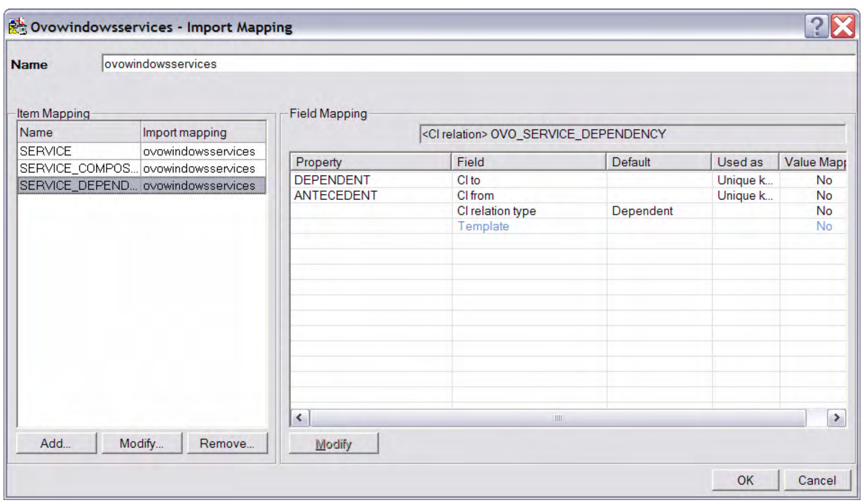
Ovowindowsservices: service_composition

Figure 12



Ovowindowsservices: service_dependency

Figure 13



Implementation Steps

Service Desk and OVOW are Running on Different Machines

On the machine where OpenView Operations for Windows runs:

1 Run `OVSDSync.exe` on the OVOW Server to create and populate the OV_SD_OW Tables

On is the machine where OpenView Service Desk Client runs (this tool needs to be run twice when if it's the first time you run it. Please following the steps illustrated in the above chapter).

2 Create a SQL Server ODBC connection.

3 Start a Data Exchange export process using the new configuration files `ovowindowsci.ini` and `ovowindowsservices.ini`.

4 Start a Data Exchange import process using the import mappings.

Service Desk and OVOW are Running on the Same Machine

1 Run `OVSDSync.exe` to create and populate the OV_SD_OW Tables (this tool needs to be run twice when if it's the first time you run it. Please following the steps illustrated in the above chapter).

2 Start a Data Exchange export process using the new configuration files `ovowindowsci.ini` and `ovowindowsservices.ini`.

3 Start a Data Exchange import process using the import mappings.

Appendix A

Service Desk Configuration File Contents

ovowindowsci.ini

```
--
-- @(#)$Workfile: ovowindowsci.ini $
--
-- Copyright (c) 1997-2004 Hewlett-Packard. All Rights Reserved.
--
-- This software is the confidential and proprietary information of
-- Hewlett-Packard. ("Confidential Information").  You shall not
-- disclose such Confidential Information and shall use it only in
-- accordance with the terms of the license agreement you entered into
-- with Hewlett-Packard.
--
-- Hewlett-Packard makes no representations or warranties about the
-- suitability of the software, either express or implied, including
-- but not limited to the implied warranties of merchantability,
-- fitness for a particular purpose, or non-infringement. Hewlett-
Packard
-- shall not be liable for any damages suffered by licensee as a result
-- of using, modifying or distributing this software or its
derivatives.
--
-- ovowindowsci.ini file for OpenView Operations for Windows and
Service Desk
-- integration.
-- Used for OVO Windows managed node -> SD CI data exchange.

[DSN]
NAME=Openview
USR=ovdb_user
PWD=openview
```

```
[SYSTEM]
LOG=              TRUE
XML=              TRUE
LOG_FILE=         OVO-WIN.log
XML_OUTPUT_FILE=  OVO-WIN.xml
APPLICATION_NAME= OVO

[CLASSES]
NAME=             MANAGED_NODE

[MANAGED_NODE]
SOURCE=           OV_SD_OW_NODE
ATT=              PRIMARYNODENAME, CAPTION, __RELPATH
COLUMNS=          OV_SD_OW_NODE.PRIMARYNODENAME, OV_SD_OW_NODE.CAPTION,
OV_SD_OW_NODE.Path AS __RELPATH
```

ovowindowsservices.ini

```
--
-- @(#)$Workfile: ovowindowsservices.ini $
--
-- Copyright (c) 1997-2004 Hewlett-Packard. All Rights Reserved.
--
-- This software is the confidential and proprietary information of
-- Hewlett-Packard. ("Confidential Information").  You shall not
-- disclose such Confidential Information and shall use it only in
-- accordance with the terms of the license agreement you entered into
-- with Hewlett-Packard.
--
-- Hewlett-Packard makes no representations or warranties about the
-- suitability of the software, either express or implied, including
-- but not limited to the implied warranties of merchantability,
-- fitness for a particular purpose, or non-infringement. Hewlett-
Packard
-- shall not be liable for any damages suffered by licensee as a result
-- of using, modifying or distributing this software or its
derivatives.
--
-- ovowindowsservices.ini file for Vantage Point for Windows and
Service Desk
-- integration.
-- Used for OVO service -> SD CI data exchange.

[DSN]
NAME=Openview
USR=ovdb_user
PWD=openview

[SYSTEM]
LOG=              TRUE
XML=              TRUE
LOG_FILE=         OVO-WIN.log
XML_OUTPUT_FILE=  OVO-WIN.xml
APPLICATION_NAME= OVO
```

```
[CLASSES]
NAME=              SERVICE, SERVICE_COMPOSITION, SERVICE_DEPENDENCY

[SERVICE]
SOURCE=            OV_SD_OW_SERVICE
ATT=               CAPTION, DESCRIPTION, __RELPATH
COLUMNS=           OV_SD_OW_SERVICE.SEARCHCODE as CAPTION,
OV_SD_OW_SERVICE.DESCRIPTION, OV_SD_OW_SERVICE.PATH AS __RELPATH

[SERVICE_COMPOSITION]
SOURCE=            OV_SD_OW_SERVICE_COMPOSITION
ATT=               GROUPCOMPONENT, PARTCOMPONENT
COLUMNS=           OV_SD_OW_SERVICE_COMPOSITION.GROUPCOMPONENT,
OV_SD_OW_SERVICE_COMPOSITION.PARTCOMPONENT

[SERVICE_DEPENDENCY]
SOURCE=            OV_SD_OW_SERVICE_DEPENDENCY
ATT=               ANTECEDENT, DEPENDENT
COLUMNS=           OV_SD_OW_SERVICE_DEPENDENCY.ANTECEDENT,
OV_SD_OW_SERVICE_DEPENDENCY.DEPENDENT
```

5 Service Pack 15

This chapter contains information previously provided with Service Pack 15.

Archiving Operators

This chapter provides a general overview of archiving operators.

Overview

The following operators have been added to the **filter** settings of any view:

- Is 2 years ago or more
- Is 1 year ago or more
- Is 6 months ago or more

The operators can only be applied to fields of type **Date**. The operators are available for every Service Desk item that can be archived, namely:

- Work order
- Change
- Project
- Problem
- Service Call
- Incident

Using the New Operators

The new operators can be used in the standard way that operators are used within a Service Desk view filter. Please note that for Archiving, the view filters must be set via the Administrator Console (**Presentation** → **Views**) to have any effect on the items archived.

The three new archive operators appear in the Operator field of the Filter Settings dialog under the Advanced tab. Note that the new operators will only appear for Fields selected which are of type Date, for example:

Figure 1

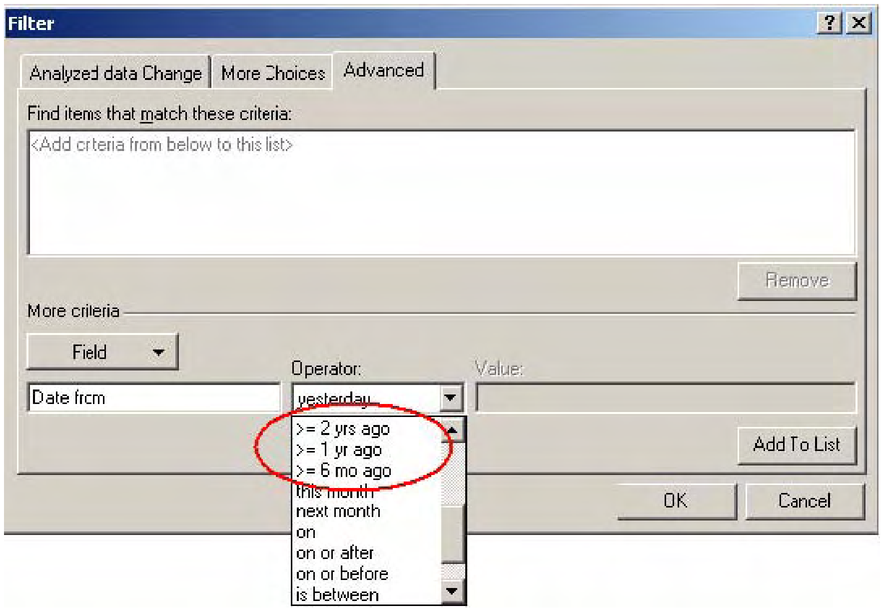


Table 1

Field	**Operator**
Date from	is >= 2 yrs ago

The archive will only contain incidents where the Date From date is 2 years ago or more.

Security

The filter functionality as described will be available to all who can define views. The usage of the views for Archiving will remain limited to System Administrator.

ACES Migration of Copy Fields

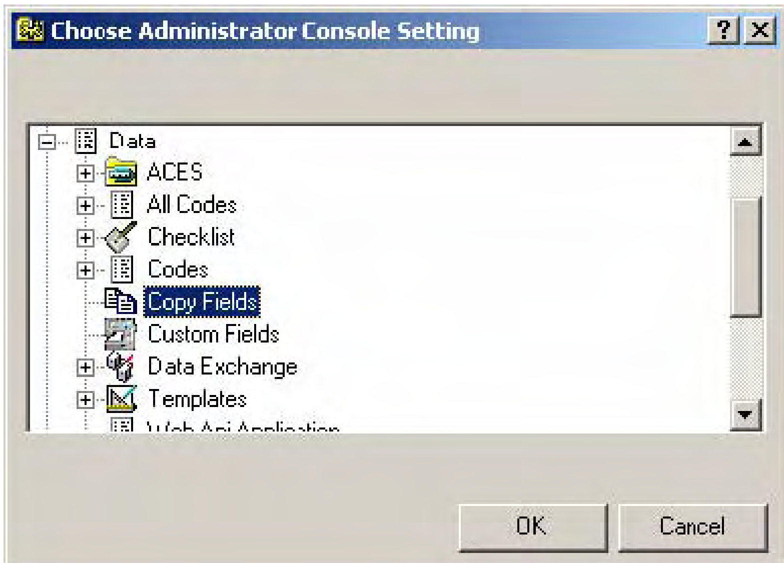
This chapter provides a general overview of how to export Copy Fields using ACES.

Overview

It is now possible to export Copy Fields settings using standard ACES functionality. The normal procedure should be followed to create an ACES Group from an ACES View and export the Copy Fields using the ACES Import/Export wizard.

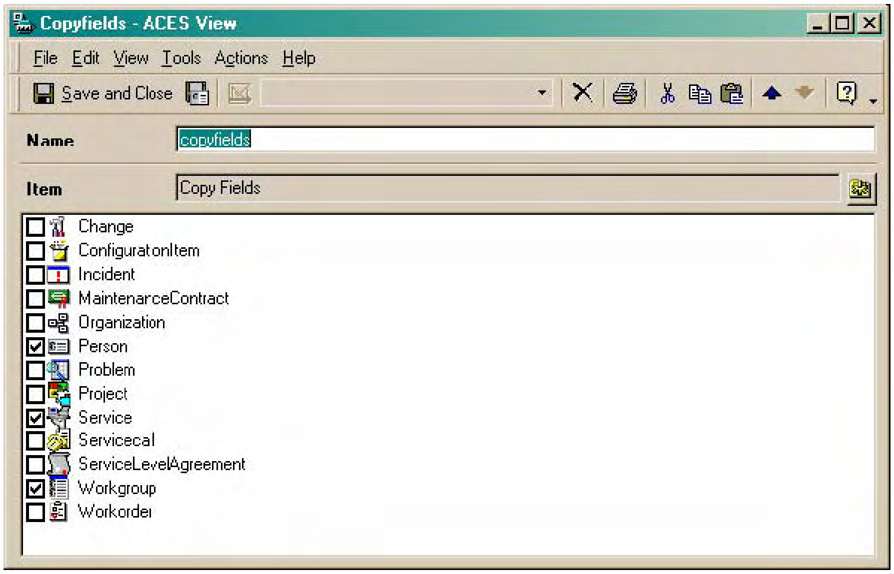
The "Choose Administrator Console Setting" dialog now contains the added *Copy Fields* item in the tree of data items that can be selected for inclusion in the view, for example:

Figure 2



Once the Copy Fields item is selected, a list of items for which the Copy Fields settings can be selected for export.

Figure 3



Then the ACES View can be saved as usual and import/export the selected Copy Fields using the standard ACES Import/Export wizard.

Security

All functionality described in this section is only available to System Administrator.

ACES Migration of Audit Rules

ACES has been adapted in order to export and import audit rules. This chapter provides a general overview of the modifications to ACES.

Overview

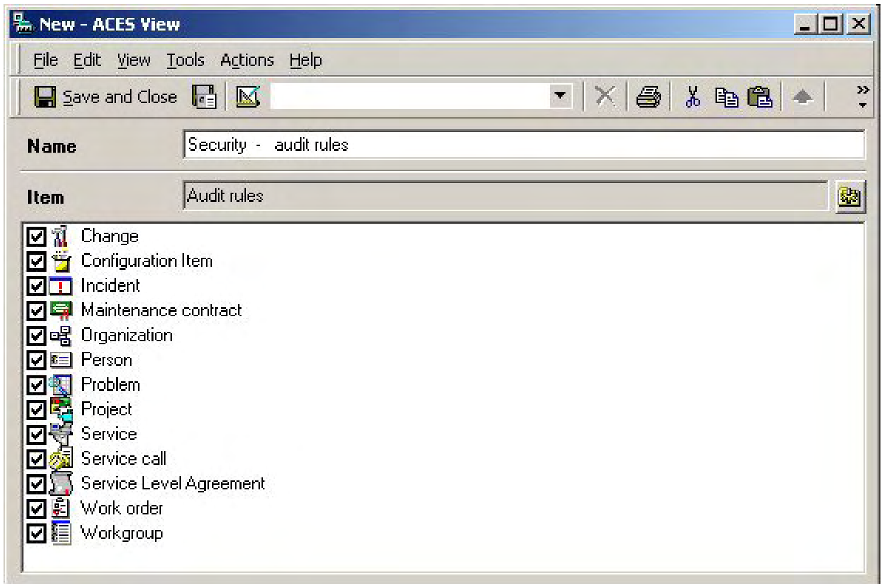
It is now possible to export Audit Rules settings using standard ACES functionality. The normal procedure should be followed to create an ACES Group from an ACES View and export the Audit Rules using the ACES Import/Export wizard.

The "Choose Administrator Console Setting" dialog now contains the added Audit Rules item in the tree of data items that can be selected for inclusion in the view, for example:

Figure 4

Once the Audit Rules item is selected, a list of items for which the Audit Rules settings can be selected for export.

Figure 5



Then the ACES View can be saved as usual and import/export the selected Audit Rules using the standard ACES Import/Export wizard.

Security

All functionality described in this section is only available to System Administrator.

Enhanced ACES Log File Readability

This chapter describes changes that enhance the ACES log file readability.

Overview

ACES (Administrator Console Export Settings) has to deal with many different objects and their settings. ACES keeps track of what is happening in a log file, but sometimes the log messages are too general or too technical.

General Improvements

The ACES log messages readability is improved using a better page layout. The indentation of the messages has been changed and the use of the newlines has also been changed.

> Newlines are operating system dependent. ACES logging uses the newlines of the operating system it runs on. When a log file is viewed on a different operating system than that on which it was created on, it still might have a bad layout (depending on the viewer).

Added Logging Features

Using Localized Names

The ACES export and import processes used to log the Service Desk internal names for items and fields. These internal names can be quite different from the localized names. Localized names are shown in the Service Desk forms and can be changed at **Tools→System→Presentation→Localized Texts→Labels**. The localized names are more readable than the internal names.

The ACES export and import process now tries to log the localized names of items and fields. At the same time, the internal name is logged between square brackets. The internal name still might be useful for the Service Desk Helpdesk. And when an item or field is not localized, at least the internal name is shown.

Log Level

The ACES export/import wizard now contains an option to decide on the log level. There are three possible log levels:

Table 2

Log Level	Description
Low	Show general log messages.
Medium	Show more detailed messages (maybe less readable).
High	Show all log messages (even stack traces).

Log level 'Low' and 'Medium' only show the first line of a stack trace. Log level 'High' shows the complete stack trace. The complete stack trace could provide useful information to the Service Desk Helpdesk when there really is something terribly wrong.

It might be that log level 'Low' or 'Medium' does not clarify the reported problems. In that case one could try to rerun the ACES export or import process at a higher log level.

At the moment there is no difference between log level 'Low' and 'Medium' in the ACES import process. However the ACES export process makes a difference between log level 'Low' and 'Medium'. It will show the less readable search conditions only at log level 'Medium' and 'High'.

Referenced Item Descriptions

ACES migrates items from a source system to a target system. The exported items might contain fields that refer to other items. If a referenced item does not exist on the target system, the import process has to deal with this in a special manner. Usually the item is imported without the reference. Sometimes the item is not imported at all (see the Service Desk Administrator Guide). In this last case the import process logs only the object id of the missing item. It is the only thing it knows about the item.

The ACES export process now provides some extra information about referenced items on the source system. The ACES import process logs this extra information when it fails to import an item due to a missing referenced item on the target system.

An example of the new log message is:

```
Failed to create relation to:
Item Account [Account] with Display name [DisplayName] 'TestAccount'
```

Serialized Item Descriptions

In Service Desk some items are stored as single, atomic objects instead of each attribute separately (serialized). These serialized items are also exported and imported serialized. But when something goes wrong the import process cannot deduct how to read these serialized items. In this case some additional information can be very useful.

The ACES export process now tries to provide some extra information for serialized items from the source system. This extra information is used by the ACES import process during logging.

At the moment this feature is implemented for the following serialized objects:

- Database Rules
- UI Rules
- Audit Rules
- Copy Fields
- Required Fields

An example of the new log message is:

```
Last read item from source file: DB rule with name 'Check approval
deadline'
```

Checking the ACES Log Files

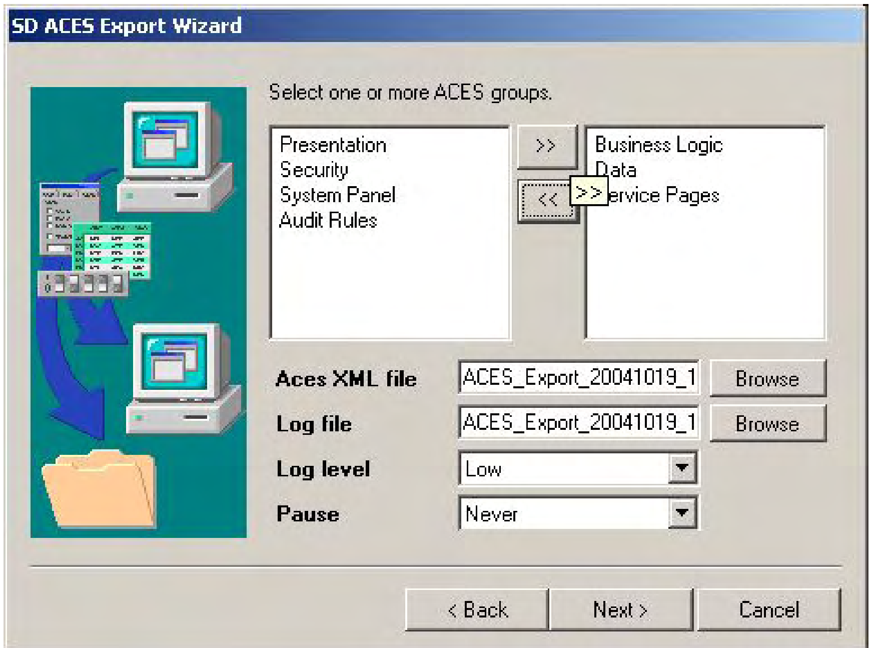
The ACES Export Log Files

The ACES export wizard can be started from the administrator console. Go to:

Tools → **System** → **File** → **Aces** → **Aces Export Wizard**.

Click **Next**. Choose a combination of ACES groups to be exported.

Figure 6

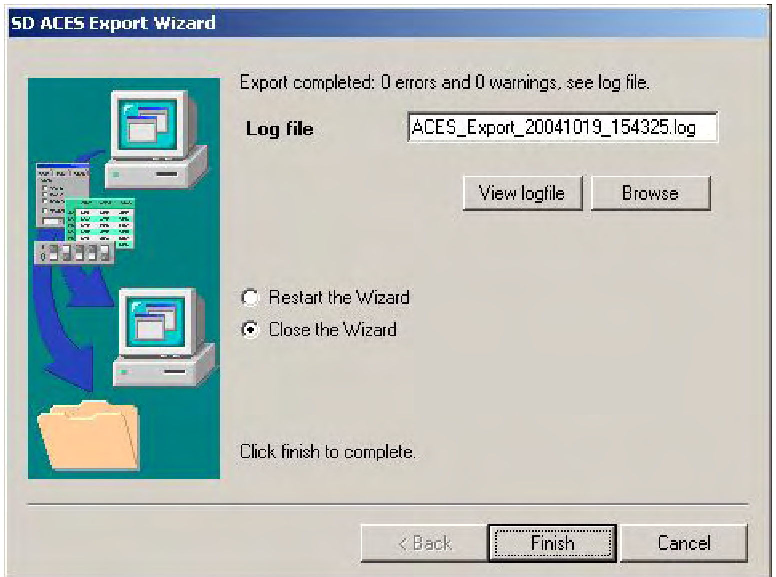


Click **Next** and then click **Start Export**.

Figure 7

When the export has finished, the wizard jumps to the last wizard page. Click **View logfile** to get a feeling for the log messages.

Figure 8



Close the log file and click **Finish**.

For people using this functionality for the first time it may be a good idea to repeat this process several times with different combinations of ACES groups. It is especially useful to provoke errors and see whether readable error messages are generated.

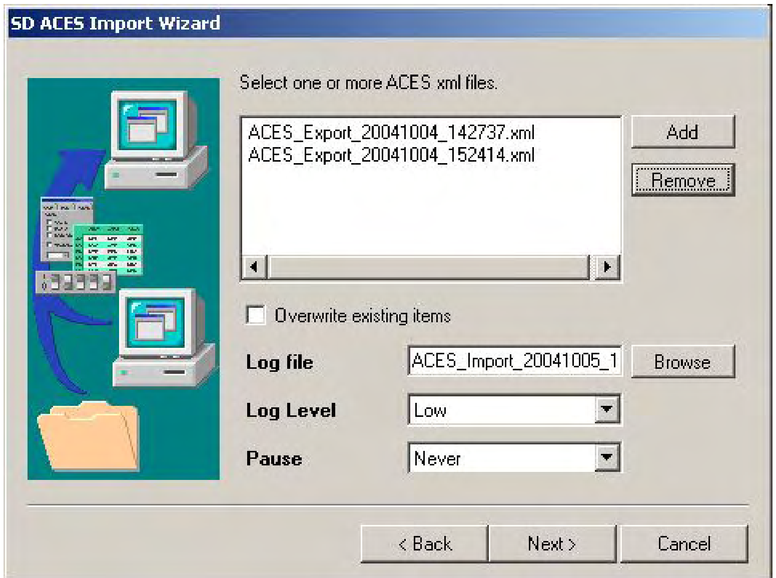
The ACES Import Log Files

The ACES import wizard can be started from the administrator console. Go to:

Tools → **System** → **File** → **Aces** → **Aces Import Wizard**.

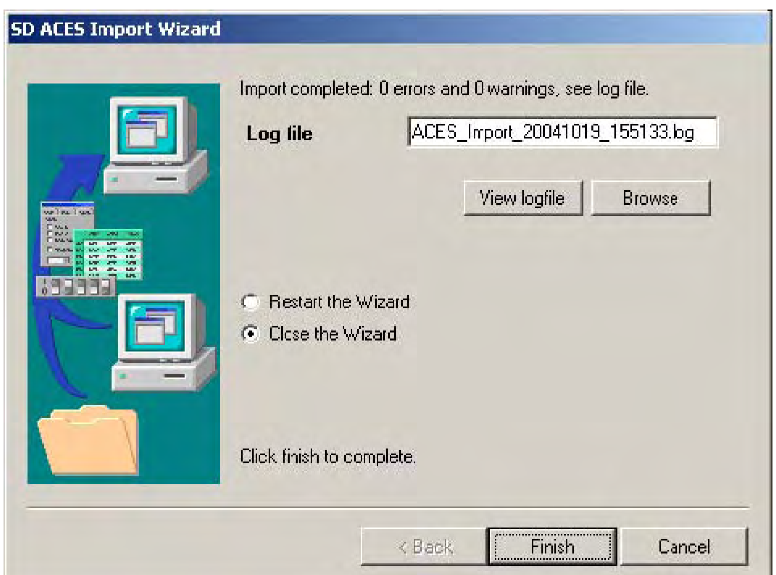
Click **Next**. Choose a combination of ACES XML files to be imported.

Figure 9



Click **Next** and then click **Start Import**. When the import has finished, the wizard jumps to the last wizard page. Click **View logfile** to see the log messages.

Figure 10



Close the log file and click **Finish**.

Again, for first time users it may be a good idea to repeat this process several times with different combinations of ACES xml files. It is especially useful to provoke errors and see whether readable error messages are generated.

Examples

An example of a test could be:

- Create an ACES view 'test' for exporting an object type of your choice.
- Create an ACES group 'test' which refers to the ACES view 'test'.
- Create an ACES view 'groups' which refers to all ACES groups.
- Create an ACES group 'groups' which refers to the ACES view 'groups'.
- Export ACES group 'groups' to '`groups.xml`'.
- Delete ACES view 'test'.
- Import `groups.xml`, check 'overwrite existing items' option.

ACES gives a notification about the missing view 'test'. Beside the object id the log file also contains the ACES view name 'test'.

Another example would be:

- Create a new application account 'test' with role 'help desk'.
- Create an ACES view 'role help desk' which exports role 'help desk'.
- Make sure to export the association 'roles per account'.
- Create an ACES group 'roles' which refers the ACES view 'role help desk'.
- Export ACES group 'roles' to `roles.xml`.
- Delete application account 'test'.
- Import `roles.xml`, check 'overwrite existing items' option.

ACES gives a notification about the missing account 'test'. Beside the object id the log file also contains the account name 'test'.

Enhanced ACES Run-Time Messaging Usability

This chapter provides a general overview of the modifications to ACES in order to enhance the ACES run time messaging.

Overview

ACES (Administrator Console Export Settings) can be used to copy the Service Desk settings from one Service Desk environment to another. First the settings are exported to an XML file, next this XML file is imported in the target environment. These export and import processes are guided by a wizard. When the actual export or import process is running, the wizard displays some messages about the progress of the process.

Changes have been made to improve the following areas:

- The wizard status field messages.
- View of the tail of the log file from the wizard.
- Let the wizard show real time feedback about the number of errors and warnings.
- Introduce the possibility to automatically pause the process on an error a warning.

Improvements

In order to provide space for the new options the layout of the existing wizard has been changed a bit. For example page two and three of the import wizard now looks like:

Figure 11 Page 2 of the import wizard

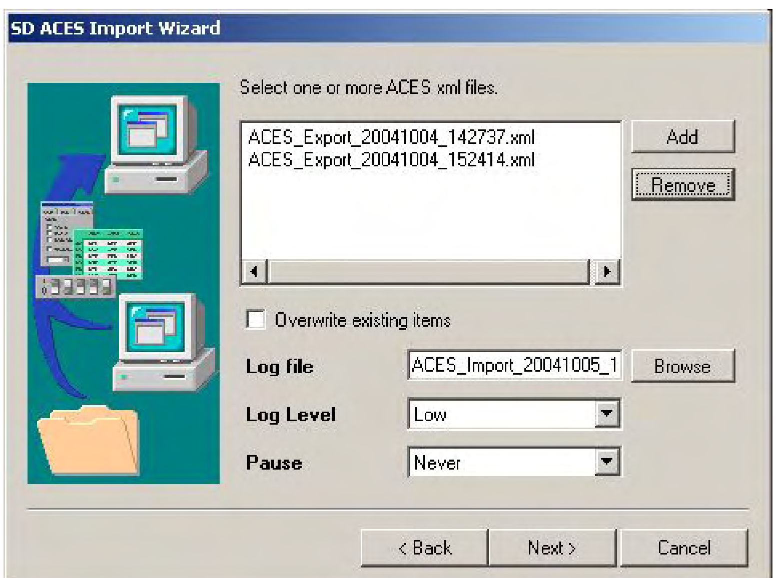
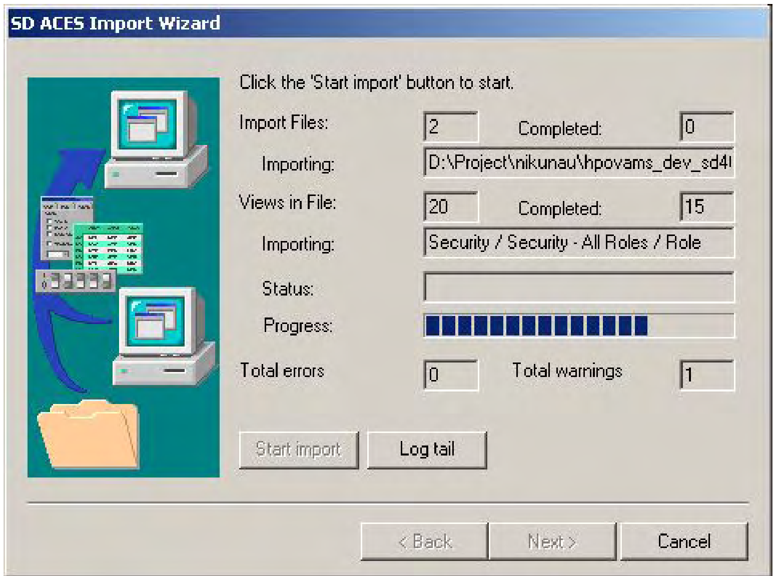


Figure 12 Page 3 of the import wizard



The layout of the export wizard has been changed in a similar way.

Run Time Status Field

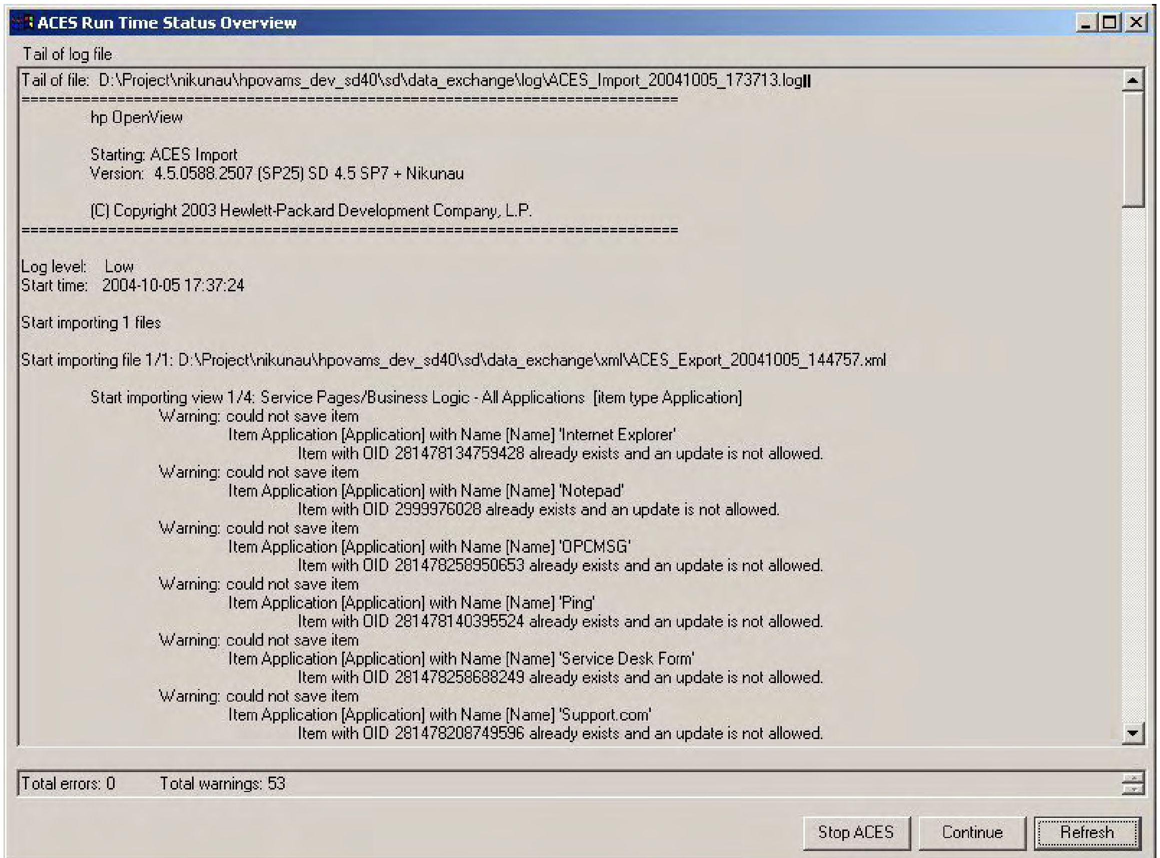
The third page of the ACES wizard has a status field which displays messages about the progress of the export or import process. Previous versions of ACES tried to fit multi-line error messages in this single line field. These error messages are now moved to the log file. The status field does not show these messages any more. It just informs about what the current export or import process is doing.

- When the exported or imported view is small the status field will be updated quickly. But it only contains progress information and no error information.
- The multi-line error messages can be viewed in the log file. For this the button **Log tail** can be used, which shows the tail of log file. The 'Log tail' functionality is described in the next paragraph.

View the Tail of the Log File

One way of getting feedback about the export or import process would be to have a look at the log file. The third page of the wizard now has a button 'Log tail' which starts a new screen called 'ACES run time status overview'.

Figure 13



It contains the tail of the ACES log file and the number of errors and warnings. In the background the ACES import or export process keeps running until 'Stop ACES' is pressed. Pressing 'refresh' will update the window and pressing 'Continue' will close the window.

The tail of log file can be 64 Kilo Bytes.

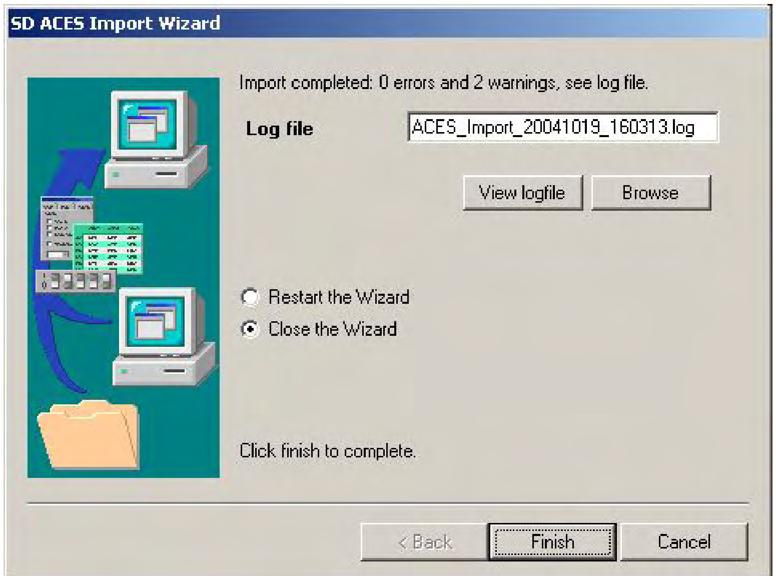
Run Time Feedback on Errors and Warnings

The third page of the wizard now provides feedback about the progress of the export or import process by showing total number of errors and the total number of warnings at run time. When an error or warning is reported, the process can be cancelled by clicking the Cancel button.

When the **Cancel** button is clicked, the dialog 'ACES run time status overview' pops up. It is the same dialog as described in the previous paragraph. In the background the ACES import or export process keeps running until 'Stop ACES' is pressed. Pressing 'refresh' will update the window and pressing 'Continue' will close the window.

The final page of the ACES wizard also shows the number of errors and warnings.

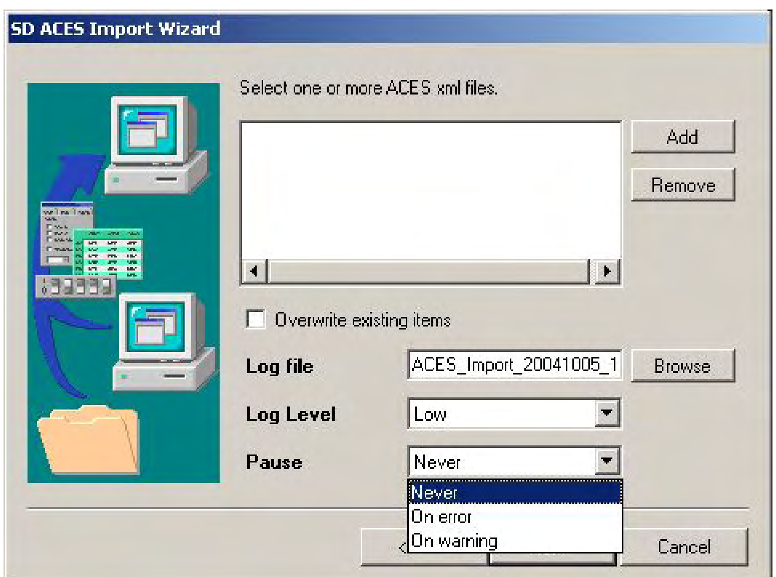
Figure 14



Pause on Error or Warning

The second page of the ACES export or import wizard now contains an option to pause the process. There are three choices: never, on error, on warning.

Figure 15



The option 'Pause never' is just the old behavior of ACES. ACES always tries to continue the export or import process. With the option 'pause on error' the process is paused on every error and with the option 'pause on warning' the process is paused on every warning. When the process is paused, the screen 'ACES run time status overview' pops up.

This is the same screen as described in the previous two paragraphs. But now the ACES export or import process does not keep running in the background. It is paused. Clicking 'Continue' will continue the export or import process and close the window. Clicking button 'Stop ACES' will really terminate the export or import process.

ACES differentiates between errors and warnings. Most of the ACES messages are warnings. Some of them are errors. ACES will normally terminate if it encounters an error.

Block/Unblock Selection

Overview

The Block/Unblock Selection Window provides a quick and convenient way for System Administrators to block/unblock the following types of Service Desk items:

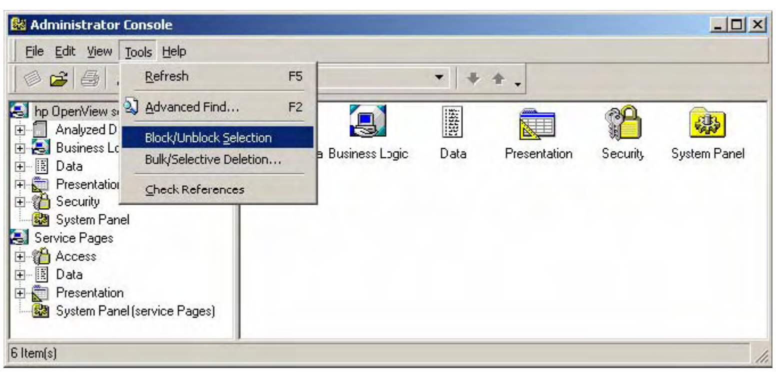
- Database Rules
- UI Rules
- Codes
- Actions
- Views
- Templates

Using Block/Unblock Selection

Starting Block/Unblock Selection

The Administrator Console contains a new menu option under the existing *Tools* option called *Block / Unblock Selection*.

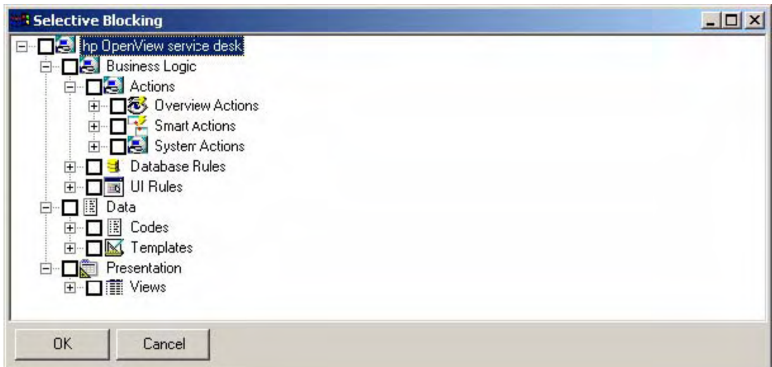
Figure 16 The Block/Unblock Selection Menu Item



Selecting the above option will invoke the Block/Unblock Selection Window (note that the title of the Window is "Selective Blocking").

The Block/Unblock Selection Window (see Figure 17) contains a tree view (similar to the Administrator Console) of all items which can be blocked.

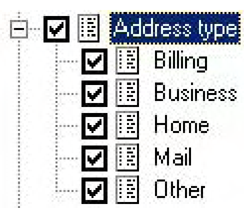
Figure 17 The Block/Unblock Selection Window



For each item in the tree a checkbox is provided. Clicking on the checkbox will "check" the item - this indicates that you would like this item to be **blocked**. Clicking the checkbox once again will remove the tick – this indicates that the item should be **unblocked**. If any item contains child items (obvious by the presence of a to the left of the checkbox) then all child items will reflect the checkbox status of the item that you selected, i.e. checked or unchecked.

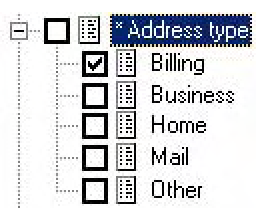
For example, checking the item "Address type" will check all the child items below it – marking everything as **blocked**:

Figure 18



If an item has an unchecked checkbox, i.e. and the name of the item starts with an asterisk (*) this indicates that at least one child item below it is checked. For example, the following shows at least one item below "Address type" is checked – in this case "Billing" :

Figure 19

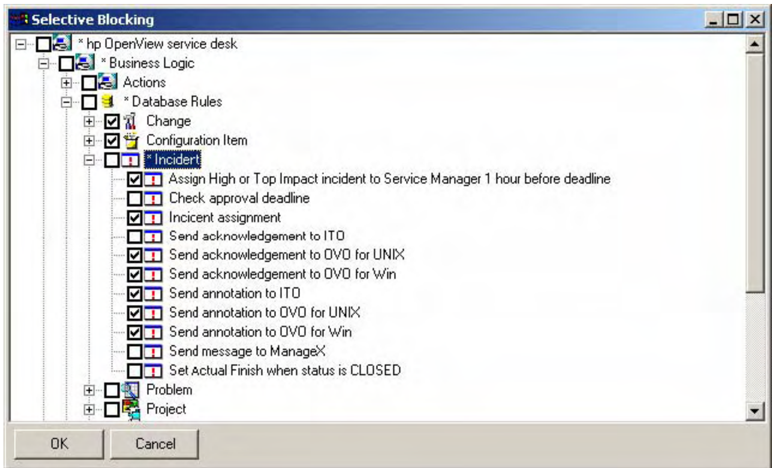


This feature allows you to quickly identify which items in the tree are marked as blocked by following the asterisks (*) down to the point in the tree where the item is checked.

On any level, a checked box indicates that all items below have also been checked. An entry in the tree marked with an asterisk (*) indicates that one or more items in the branch below have been checked.

Items with underlying structure can be recognized by the indicator. Expanded items will have in front. Items without either are (terminal) elements.

Figure 20 Block/Unblock Selection Window showing multi-selection of DB Rules



Loading the Block/Unblock Selection Window takes some time, mostly depending on the volume of data in the database.

Block/Unblock Items

Each item within the tree can be individually selected for blocking or unblocking.

Clicking on an unchecked checkbox will check that item and all items in the lower level structure below the item (if any). Similarly, unchecking an item will uncheck all items in the structure below (if any).

Example: Checking **Codes** at the top level will check all Codes.

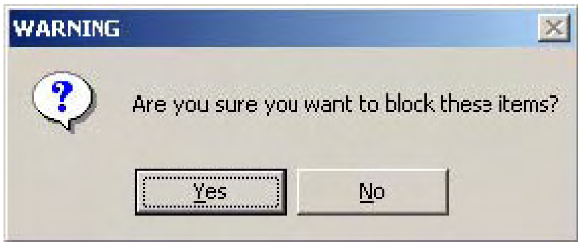
Clicking on the preceding a rolled-up item will expand the structure below that item one level. Clicking will roll up an expanded item completely.

If an item with a structure beneath has been checked and an item at a lower level is subsequently unchecked, all upper level items will be unchecked. In addition to the upper level items being unchecked they will also be marked with an asterisk (*) to indicate that there are item(s) below that are checked. This behavior allows a user to easily follow the asterisk path down a tree to the item(s) which are checked. This is especially useful when an item is checked at a low level in the tree and would otherwise be difficult to locate.

Example: if all Codes are checked, and only the Status Code 'Announced' for Service Calls is unchecked, the Status Code for Service Calls will be unchecked as well and marked with an asterisk (*). This is also true for Service Calls and Codes levels. All other items will remain checked.

Finally, when the user has finished selecting the items they wish to be blocked/unblocked and clicks OK, they are presented with a confirmation dialog as follows:

Figure 21



If the user clicks **Yes**, the changes will be applied and the user will be returned to the Administrator Console.

If the user clicks **No**, the changes will not be applied and the user will be returned to the Block/Unblock Selection Window.

Security

All functionality described in this section is only available to System Administrator.

Bulk/Selective Deletion

Overview

The Bulk/Selective Deletion window provides a quick and convenient way for System Administrators to delete the following Administrator Console items:

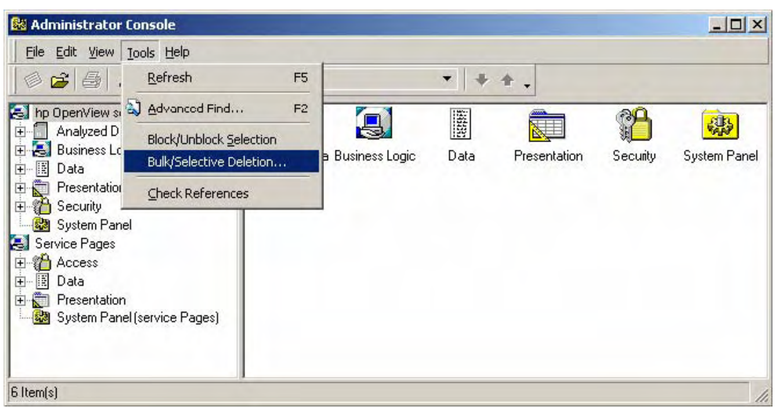
- UI Rules
- Database Rules
- Actions (Overview & Smart)
- Views
- Templates

Using Bulk/Selective Deletion

Starting Bulk/Selective Deletion

The Administrator Console contains an option under the existing *Tools* option called *Bulk/Selective Deletion*.

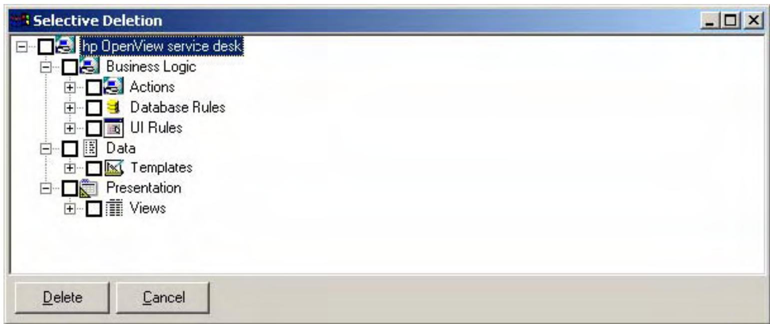
Figure 22 Bulk/Selective Deletion Command



Selecting the above option will invoke the Bulk/Selective Deletion Window.

The Bulk/Selective Deletion window (see Figure 23) contains a tree view (similar to the Administrator Console) containing Database Rules, UI Rules, Actions (Overview & Smart), Views and Templates which can be deleted.

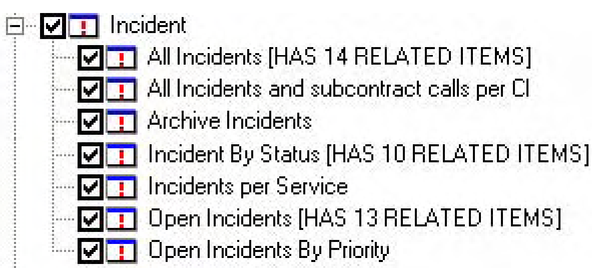
Figure 23 Bulk/Selective Deletion Window



For each item in the tree a checkbox is provided. Clicking on the checkbox will "check" the item - this indicates that you would like this item to be **deleted**. Clicking on the checkbox once again will remove the tick – this indicates that the item should NOT be deleted. If any item contains child items (obvious by the presence of a to the left of the checkbox) then all child items will reflect the checkbox status of the item that you selected, i.e. checked or unchecked.

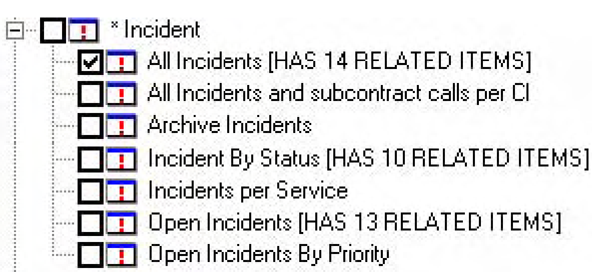
For example, checking the item "Incident" will check all the child items below it – marking everything for **deletion**:

Figure 24



If an item has an unchecked checkbox, i.e. and the name of the item starts with an asterisk (*) this indicates that at least one child item below it is checked. For example, the following shows at least one item below "Incident" is checked – in this case "All Incidents" :

Figure 25

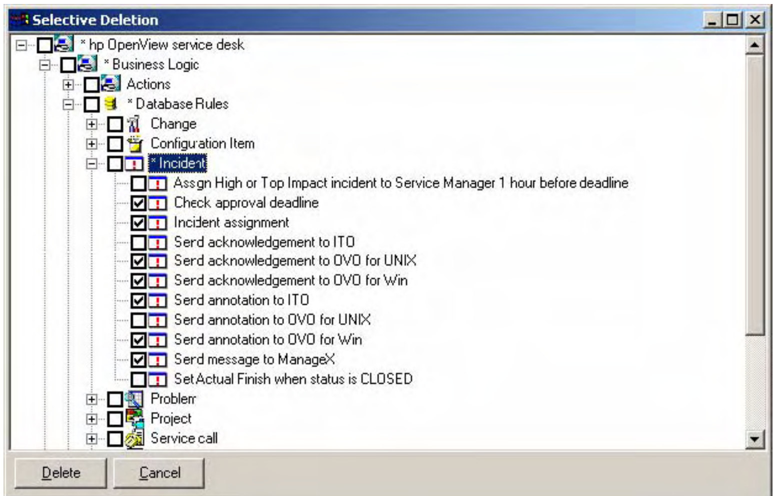


This feature allows you to quickly identify which items in the tree are marked for **deletion** by following the asterisks (*) down to the point in the tree where the item is checked.

On any level, a checked box indicates that all items below have also been checked. An entry in the tree in marked with an asterisk (*) indicates that an item in the branch below has been checked.

Items with underlying structure can be recognized by the indicator. Expanded items will have in front. Items without either are (terminal) elements.

Figure 26 Bulk/Selective Deletion Window – Example of multi-selection for DB Rules



Loading the Block/Unblock Selection Window takes some time, mostly depending on the volume of data in the database.

General Bulk/selective Deletion

Each item within the tree can be individually selected for **deletion**.

Clicking on an unchecked checkbox will check that item and all items in the lower level structure below the item (if any). Similarly, unchecking an item will uncheck all items in the structure below (if any).

Example: checking Views at the top level will check all views.

Clicking on the preceding a rolled-up item will expand the structure below that item one level. Clicking will roll up an expanded item completely.

If an item with a structure beneath has been checked and an item at a lower level is subsequently unchecked, all upper level items will be unchecked. In addition to the upper level items being unchecked they will also be marked with an asterisk (*) to indicate that there are item(s) below that are checked. This behavior allows a user to easily follow the asterisk path down a tree to the item(s) which are checked. This is especially useful when an item is checked at a low level in the tree and would otherwise be difficult to locate.

Finally, when the user has finished selecting the items they wish to be deleted and press *Delete*, they are presented with a confirmation dialog as follows:

Figure 27

Undo functionality does NOT exist for Bulk/Selective Deletion.

If the user clicks **Yes**, the selected items will be deleted and the user will be returned to the Administrator Console. If any of the selected items could not be deleted the following message will be displayed:

Figure 28

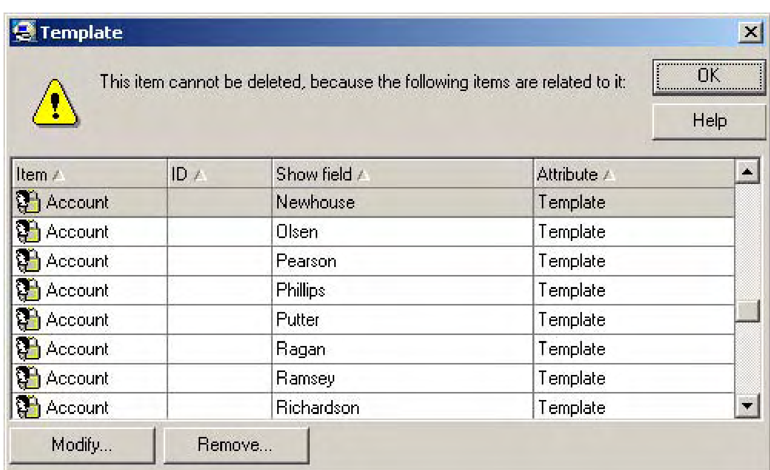
(see *Deletion Logs* for further details)

If the user clicks **No**, the changes will not be applied and the user will be returned to the Bulk/Selective Deletion Window.

Deleting Items with Relations

If an item is being referenced by another Service Desk item(s) and is selected for deletion, the following (standard Service Desk) window will appear:

Figure 29

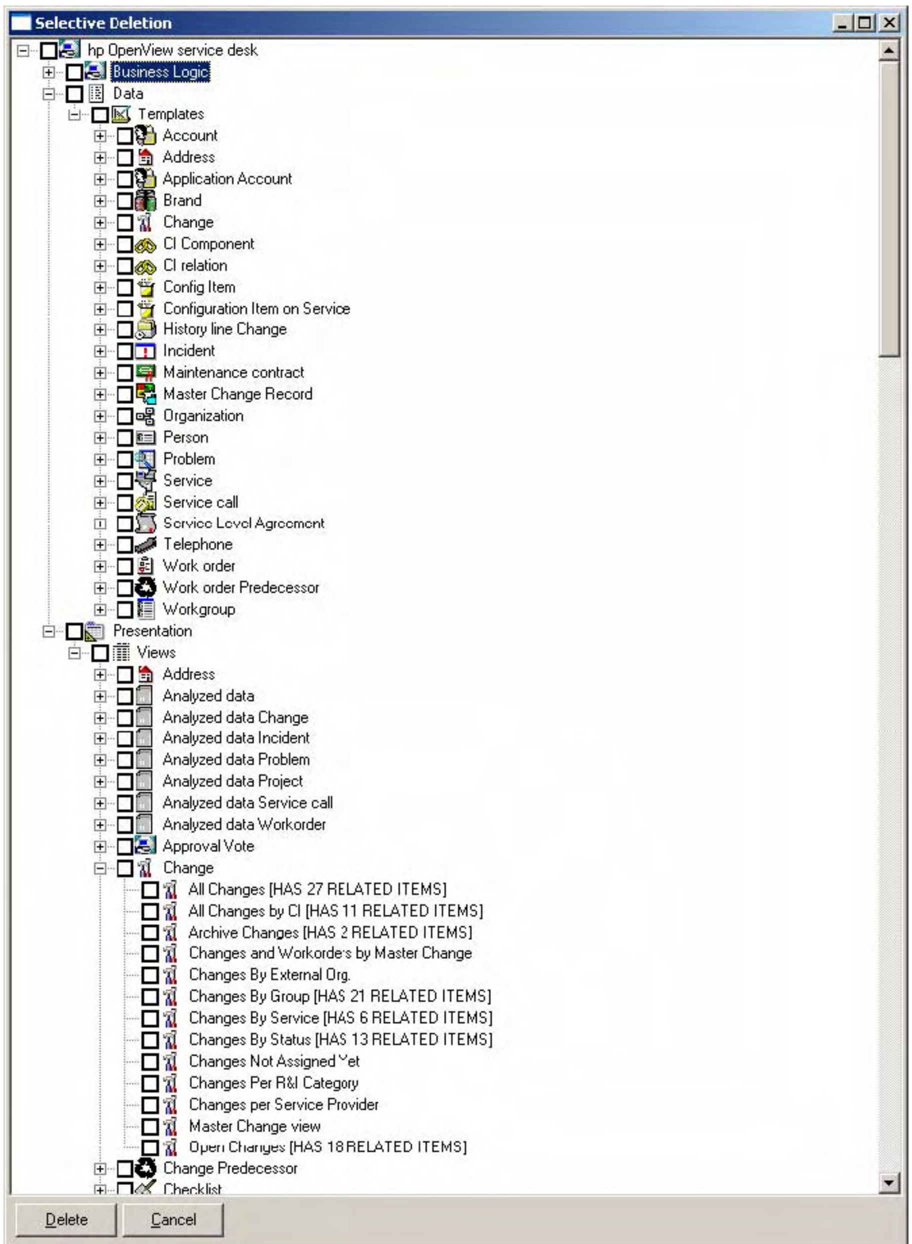


The normal procedure should be followed before the selected item will be deleted, i.e. delete or modify the items that are referencing the selected item.

Note that the above window will appear for each item that has been selected for deletion that has items related to it. If a number of items are selected which have related items, this could potentially take a long time to resolve. In order to minimize this and to allow System Administrator's to ignore these potentially time-consuming items (if they choose to) the Selective Deletion window will automatically pre-open all items that have related items. Additional information will appear in the tree next to the item name which details exactly how many items are related to the item, e.g. "All Changes **[HAS 27 RELATED ITEMS]**". This feature is also designed to prevent accidental selection of large numbers of items that have related items (remember that the above window will appear for each item selected that has related items). **Please note this information is NOT reported for Templates due to performance considerations.**

Below is an example of the pre-opening of items which have related items in the Bulk/Selective Deletion Window (note that **Templates** do not report the related item information):

Figure 30



Deletion Logs

The client log file (normally called LOGCLIENT.TXT) contains entries both for successful deletions and unsuccessful deletions and is useful as both an audit trail and to troubleshoot problems with deletion. For example, a successful deletion of an item will result in an entry similar to the following:

```
Wed, 06/10/2004 14:30:28 <System> Bulk/Selective Deletion:
successfully deleted View "Service Today For Workgroup by item"
(281480031895559)
```

and an unsuccessful deletion will result in an entry similar to the following:

```
Wed, 06/10/2004 14:29:50 <Trace> Bulk/Selective Deletion ERROR:
failed to delete Template: To: Network connection (281478278087301).
Please delete this item manually via the Administrator Console.
```

Security

All functionality described in this section is only available to System Administrator.

References Audit

Overview

The References Auditor provides the System Administrator with a report of Service Desk items that have been deleted / blocked but are still being referenced. With this information the System Administrator can take the appropriate action required, i.e. deleting items/unblocking items etc.

Additionally, the References Auditor reports on the following items which are missing/blocked, but are related to a **Role**:

- Actions (Overview & Smart)
- Forms (reports on *missing* Forms only as currently Forms cannot be blocked)
- Views
- Templates

The References Auditor will automatically select all current Roles defined within Service Desk and audit on each of the above items for each Role. Note that the Roles audit is always the first audit to run and therefore appears first in the generated report.

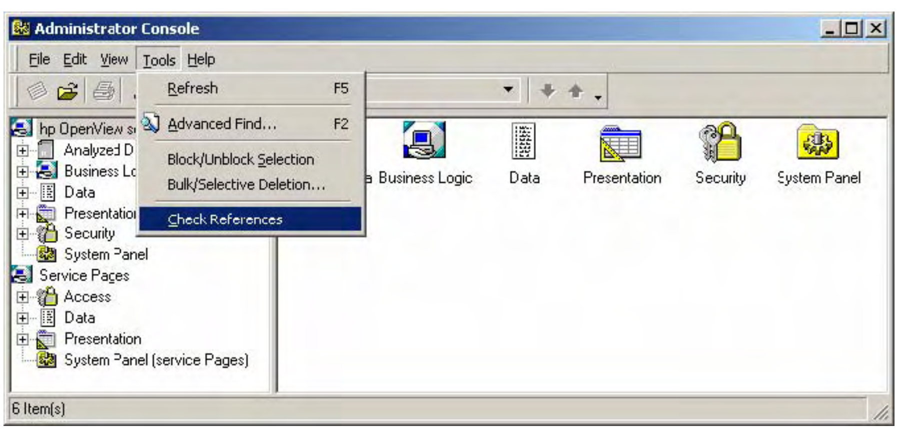
Using Check References

This section describes how to start and use Check References.

Starting Check References

The References Auditor can be started by using the **Check References** command from the **Tools** menu of the Administrator Console, for example:

Figure 31



Audit Results

After some time, mostly depending on the size of the database the system default browser (Internet Explorer, Netscape Navigator etc.) should open a window showing a page with the audit results. For readability, the example (overleaf) does not show a screenshot but a simulation of that page for descriptive purposes.

References Report generated at May 5, 2004 3:26:47 PM (Central European Time).

Affected Item Type	Affected Item	Field	Problem	Problem Item Type	Problem Item
Role	Change	Incident	Blocked	Overview action	Calls by category for this caller
Role	Helpdesk	Action Access	Missing	Smart action	3095983453
View info	Changes Not Assigned Yet (Table)	Assignment Status	Missing	Assignment Status	3095986180
View info	Open Work Orders (Table)	Status	Missing	Work order status	3095134440
View info	Work order Open By External Contact (Table)	Status	Missing	Work order status	3095134440
View info	Work order Open By External Org. (Table)	Status	Missing	Work order status	3095134440
View info	Work order Open By Group (Table)	Status	Missing	Work order status	3095134440
View info	Work order Open By Specialist (Table)	Status	Missing	Work order status	3095134440
Template	Main Incident template	Status	Missing	Incident Status	846331905
Action access	281478258950694	Action	Blocked	Action	Generate OpenView Operations Message manually
Smart Action Parameter	281478258950702	Smart action	Blocked	Smart action	Generate OpenView Operations Message manually
Smart Action Parameter	281478258950705	Smart action	Blocked	Smart action	Generate OpenView Operations Message manually
Smart Action Parameter	281478258950706	Smart action	Blocked	Smart action	Generate OpenView Operations Message manually
Smart Action Parameter	281478258950708	Smart action	Blocked	Smart action	Generate OpenView Operations Message manually

Checked all in 3.594 seconds.

It may take several hours to generate the report on database with scalable size. During the period, the client will not respond. Please DO NOT shut it down forcefully.

The previous page shows the audit results – you can see that two actions have been identified as a problem and are each related to a Role. The first item (an Overview Action) is Blocked and is related to the "Change" Role. The column "Field" indicates in this case that the item can be found under the Incident type in the Administrator Console or Selective Blocking. The second item (a Smart Action) is being referenced by the "HelpDesk" Role but is missing. The column "Field" in this case simply indicates the field which is referencing the missing item, "Action Access.". As the item is missing, the "Problem Item" column only contains the ID of the missing item as it is impossible to get more information for it.

You can also see that two status items have been deleted that were still referenced by Views. Also a blocked Smart Action is referenced by an Action Access object (security) and by a Smart Action Parameter.

A row like this:

Table 3

Affected item type	Affected item	Field	Problem	Problem item type	Problem item
Template	DTAVIDEO	Category	Missing	CI Category	3095396873

Affected item	Field

Reads: "A template called "DTAVIDEO" contains a reference in the field "Category" to the missing CI Category with ID 3095396873". The blue arrow shows the direction in which the reference points.

The first two columns show the item that contains the reference, the third and fourth column show the reference and what's wrong with it and the last two columns show the item that causes the problem. For items related to a **Role**, the "Field" column is the name of the type of item the problem item belongs to, i.e. "Service Call". This means that if two Problem Items appear in the report that have the same name, the particular item can be found and fixed by looking under the the type of item reported in the "Field" column. For example, the Bulk/Selective Blocking window could be invoked for a **blocked** item in the report found easily by navigating to "Service Call" or "Incident" etc. and unblocked because the type of the item is clear in the report.

In general there are three ways to resolve a problematic reference:

1 Remove the reference; this involves opening the affected item and modifying the field that contains the reference.

2 Remove the whole affected item, this effectively removes the reference as well.

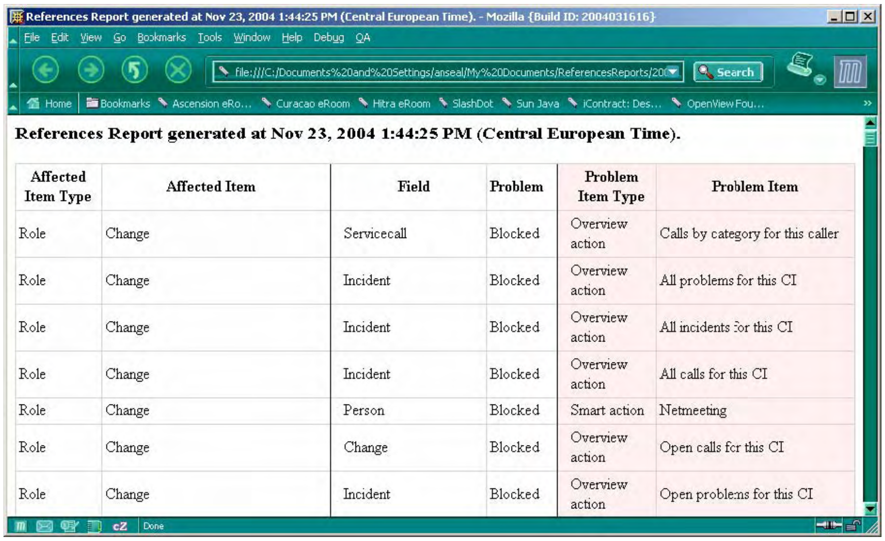
3 Unblock the problem item (only for blocked references)

For actually resolving the problems here are some tips:

- Templates are somewhat "self healing"; deleted references will not be shown when the template is opened. Just opening a template, changing something, changing it back and saving will remove any references to deleted targets.
- Sometimes the container cannot be opened directly, for example an Action Access item can only be changed by opening the role that contains it and click on "Advanced..." / "Actions"

Below is an actual example report containing items that are **Blocked** and are being referenced by a Role:

Figure 32



References Report generated at Nov 23, 2004 1:44:25 PM (Central European Time).

Affected Item Type	Affected Item	Field	Problem	Problem Item Type	Problem Item
Role	Change	Servicecall	Blocked	Overview action	Calls by category for this caller
Role	Change	Incident	Blocked	Overview action	All problems for this CI
Role	Change	Incident	Blocked	Overview action	All incidents for this CI
Role	Change	Incident	Blocked	Overview action	All calls for this CI
Role	Change	Person	Blocked	Smart action	Netmeeting
Role	Change	Change	Blocked	Overview action	Open calls for this CI
Role	Change	Incident	Blocked	Overview action	Open problems for this CI

References Report Folder

The Reports Folder which can be set in the General tab of the Options dialog, under the Tools menu, contains a sub-directory called `ReferencesReports` which is created by the References Auditor for the purpose of archiving all generated html report files. Each file contains the creation date and time as part of the filename so that individual reports based on the date/time can be easily retrieved.

Fixing Problem Types

This list will present possible solutions to problems found by the References Audit. It is organized per Affected Item Type. Missing Items are only possible for Rules, Views and Templates.

UI Rule / Database Rule

The rules are easy to find since the name is given. The symptom of a missing object is either a 'null' in the condition or a blank field where something should be. This can be anywhere in the condition or action part of the rule.

View Info

This is actually a View. A view contains a filter which is essentially the same kind of condition as in a Rule. Therefore similar rules apply.

Template

These are easy to find by name. The problems will most likely be in the template values. A template value that is missing will not be shown. Opening and closing the template is often enough to fix references to missing template values.

Impact Priority Setting

These are found in the Service Level which can be edited by opening an SLA, use Quick Find to get a list of Service Levels and then open one using right click. This Service Level might be blocked. It contains a list of the Impact -> Priority mappings for that Service Level. The Impact and Priority codes might be blocked too.

Person

This one is obvious. It can be found by name.

Roles per Account

These can be accessed either by going to the Role and looking at the list of accounts or the other way around. By un-checking the relevant checkbox this object is removed and the problem solved. Of course unblocking the Account works too (Roles cannot be blocked).

Action Access

The blocked action can be found by name and then deleted or unblocked.

Smart Action Parameter

The blocked action can be found by name and then deleted or unblocked.

Overview Condition Info

This is the part of an Overview Action that contains the condition that must hold to execute the action. It will usually be the Overview Action itself that is blocked. The resolution is to either unblock or delete the action.

Security

All functionality described in this section is only available to System Administrator.

Terms and Abbreviations

Table 4

Term	Explanation
ACES	Administrator Console Export Settings: a Service Desk feature to export and import administrator console objects.

6 Service Pack 16

This chapter contains information previously provided with Service Pack 16.

Migration of Attachment Folder Structure

Users upgrading from SP15 (or earlier) to SP16 (or higher) must migrate existing attachment folder structure.

Introduction

In Service Desk 4.5, attachments are stored as files on a FTP server. According to the current folder structure, attachments of same type item are stored in single folder. Since some file systems impose a physical limit on the number of entries within a single folder, Service Desk cannot store large number of attachments of same type item. In service pack 16 the folder structure is changed for storing more attachments. The existing attachments in service desk have to be migrated into this new structure.

This guide gives a step by step plan for migrating existing attachments.

Folder Structure Reconstruction

Service Desk 4.5 (pre service pack 16) used the following attachment folder structure:

```
\Attachmentroot\Entitytype-name\Entity-oid\Attachment-oid
```

Entity is the technical term for items in Service Desk. OID means object id which is internal representation of items in service desk.

For example, there are 3 service calls with 2 attachments each.

Table 1

Service call OID	Attachment OID
281479802001001	281479802002001, 281479802002002
281479802001002	281479802002003, 281479802002004
281479802001003	281479802002005, 281479802002006

The old attachment folder structure is listed as following:

```
Root folder in FTP server = /opt/OV/sd/attachments
```

Attachments are stored in:

```
/opt/OV/sd/attachments/servicecall/281479802001001/281479802002001
/opt/OV/sd/attachments/servicecall/281479802001001/281479802002002
/opt/OV/sd/attachments/servicecall/281479802001002/281479802002003
/opt/OV/sd/attachments/servicecall/281479802001002/281479802002004
/opt/OV/sd/attachments/servicecall/281479802001003/281479802002005
/opt/OV/sd/attachments/servicecall/281479802001003/281479802002006
```

Some file systems impose a physical limit on the number of entries that may be stored within a single folder. The following table lists some of the most common types in use by our installed base:

Table 2

OS	Version	Limit
HP-UX	Standard	32767
HP-UX	PHKL_28185	65534
Windows	FAT32	65534
Windows	NTFS	4,294,967,295 Files per volume

When the number of items of the same type reaches this limit, it is no longer possible to attach files.

The new folder structure is constructed as following:

```
\Attachmentroot\Entitytype-name\Entity-uuid-
fragements1...11\Attachment-uuid
```

UUID (Universally Unique Identifiers) is generated from object OID. It is 128 bits (16 bytes) long. Entity-uuid-fragements1...11 represents breaking entity uuid into 11 fragments. The first ten fragments are 3 bytes long each and the last fragment is 2 bytes long.

Entity-uuid is generated from entity OID and Attachment-uuid is generated from attachment OID. Attachment-uuid uses the following format (where H is a hexadecimal number):

```
HHHHHHHH-HHHH-HHHH-HHHH-HHHHHHHHHHHH
```

In new attachments folder structure, a Service call attachment path will be like:

```
Attachmentroot\Servicecall\000\000\000\000\000\000\010\000\c31\100\fa\0
0000000-0000-0000-0001-0000c6280140
```

Compared to the old attachments folder structure, the number of entries in any folder decreases to 16^3=4096 which is well below the limit. So it won't cause mentioned limitation.

The change requires the existing attachments in service desk to be migrated to new structure.

Getting Started

Install Perl 5.8

Before running the migration script, Perl 5.8 must be installed on FTP server.

Table 3

OS	Download Website
Windows	**http://www.activestate.com/Products/ActivePerl/**
HP-UX	**http://hpux.cs.utah.edu/hppd/hpux/Languages/perl-5.8.7/**
Solaris	**http://www.activestate.com/Products/ActivePerl/**

After installing perl on FTP server, please verify perl version by executing command line

```
perl -version
```

The version must be higher or equal 5.8.

How to Execute Migration Script

Preliminary: Execution privilege to Perl application and full access to attachment folder is required.

It is recommended to backup old attachments before executing the migration script.

Execute <**`service pack extraction folder`**>`\doc\itsm007474\mig_attachments.pl` on FTP server in shell or command line.

```
perl mig_attachments.pl [-c] <sourcedir> <targetdir>
```

Also please make sure to log the output of the script by redirecting the standard output and standard error to a log file. On Unix shells, this is usually done as "> outputfile.log 2>&1".

```
perl mig_attachments.pl olderfolder newfolder > migoutputfile.log 2 > & 1
```

On Windows shells (non-FAT file system), type following command:

```
perl mig_attachments.pl olderfolder newfolder > migoutputfile.log 2 > & 1
```

On Windows shells (FAT32 or FAT16 file system), type following command:

```
perl mig_attachments.pl -c olderfolder newfolder > migoutputfile.log 2 > &
1
```

After execution, please check the `migoutputfile.log` to find migration result.

Hard Link vs. Actual File Copy

By default, the script creates hard links from the current attachment folder structure to the new attachment folder structure. A hard link is an additional reference to an existing file or directory on a disk. By using hard links, the amount of space required for the migration is limited to the overhead of having a new directory structure and the additional references. The performance is high compared to actual copying the file data.

Many operating systems have limitations on the use of hard links. On many Unix systems, it is not allowed to create hard links across different partitions. On Windows, hard links are only allowed on NTFS file systems.

If for some reason you cannot use hard links on your FTP server, you can specify the -c option as first argument to mig_attachments.pl, which will instruct the migration program to copy the files to their new structure instead of hard linking them. **Using -c option, disk space reserved should be at least the same as that of existing attachments.** Also the time consumed will be greater.

Table 4

Item	Use Hard link	Use Actual File Copy
Performance	High (less than half hour)	Low (few hours)
Disk Space	Little	Same disk space used by existing files
Script execution	No additional parameter	-c option

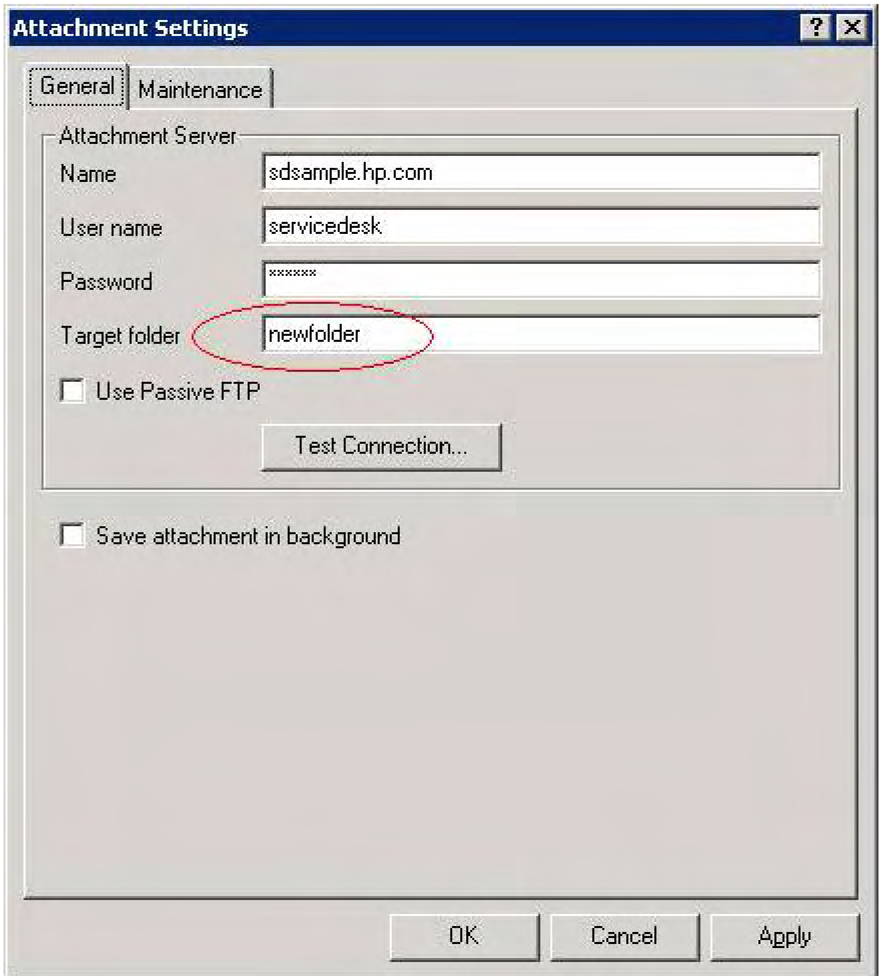
Migration Strategy I: Offline Migration

Offline strategy is applicable for file system which supports hard link (faster migration).

Detailed steps are:

1. Install perl 5.8 or higher.
2. Stop application server.
3. Back up existing attachments.
4. Create new attachment root folder and check disk space.
5. Execute the migration script.
6. Upgrade service desk.
7. Start application server and client, change folder of attachment setting. **Tools** → **System** → **System Panel** → **Attachments Settings**→ Change **Target folder** to `newfolder`.

Figure 1



Migration Strategy II: Online Migration

Online strategy is applicable for file system which DOES NOT support hard link (slower migration).

Detailed steps are:

1. Install perl 5.8 or higher.
2. Back up existing attachments.
3. Create new attachment root folder and check disk space.
4. Execute the migration script using -c option while the service desk of old version is still running (This won't impact the attachment function as the script only copies the attachments).
5. After first round of migration is finished, stop the application server.
6. Upgrade service desk.
7. Execute the migration script using -c option again to migrate the files changed after first round migration started. The second round will take less time because no effort will be made to link or copy files that already have been migrated.
8. Start application server and client, change folder of attachment setting.

Frequently Asked Questions

Q: Why doesn't my perl version change after installing new one?
A: Sometimes perl command is a link to real perl execution file. The perl command keeps the original link even when you have installed a newer version. You should change the link to new version perl execution file. Another possible reason is that in environment variable **path** the original perl execution file is prior to new one.

Q: Does migration destroy original attachment folder structure?
A: The migration just copies original attachment files. So it won't cause any corruption on the original folder structure and files.

Q: How about the migration performance?
A: There is no performance issue in case that file system supports hard link. For file system not supporting hard link it will take more time to migrate. Usually it will be few hours based on the number of attachments and hard disk speed. Please refer to Hard Link vs. Actual File Copy.

Q: How much disk space do I need to reserve?
A: Only little (additional) disk space is required in case that file system supports hard link. For file system not supporting hard link at least the same amount disk space should be reserved as consumed by the original attachments. Please refer to Hard Link vs. Actual File Copy.

Q: Could you recommend steps regarding migration?
A: For faster migration we recommend to use Migration Strategy I: Offline Migration. For slower migration we recommend to use Migration Strategy II: Online Migration

Q: Does the execution of migration script depend on application server?
A: No.

Q: Can user access attachments when migration is being executing?
A: Users can access those attachments when migration is going because the migration just copies the original attachments.

Q: Can I delete old attachment folder after migration?
A: Yes. You can delete old attachments safely no matter you are using hard link or actual file copy.

Q: Could you explain the new attachment folder structure?
A: Please refer to Folder Structure Reconstruction.

Q: How about the compatibility with SD 5.0?
A: The new attachments folder is compatible with SD 5.0.

Service Desk on Citrix

Publishing the Service Desk 4.5 Client

This document explains how you enable the Service Desk 4.5 client to work as a Citrix Metaframe Presentation Server 4.0 published application.

Requirements

Before you are able to publish the Service Desk 4.5 client several requirements have to be fulfilled. This chapter describes the steps to take.

1. Ensure the Citrix Metaframe Presentation Server 4.0 has been succesfully installed and is working correctly. Use the Citrix documentation for guidance to install and configure the application.

   > Although the Citrix Metaframe Presentation Server can be used on Unix based operating systems, this is NOT supported for using the Service Desk 4.5 client because there is no possibility to run the Service Desk 4.5 client on Unix based operating system because the Microsoft Java Virtual Machine is required.

2. Ensure the Citrix environment has enough licenses available for the users to start published applications. For example start the "License management console" and make sure a license is working and available.
3. Install the Service Desk 4.5 client on the Citrix Metaframe Presentation Server 4.0 machine. Please refer to the Service Desk 4.5 installation documentation for instructions.

   > By default, the installation of the Service Desk 4.5 client on Windows 2003 is not supported. However, for use with the Citrix Metaframe Presentation Server 4.0 there is an exception but you do need to install the Microsoft Java Virtual Machine first because by default this is not installed on Windows 2003. See the chapter "Installing *Microsoft Java Virtual Machine on Windows 2003"* for details.

4. A small modification has to be made to the "sd_client.bat" if you do not want to see a DOS box that remains open every time you start a Service Desk 4.5 client using Citrix. Steps to take:
   a. Navigate to the Service Desk 4.5 installed location and open the bin folder.
   b. Edit the "sd_client.bat" file
   c. Replace the beginning of the line "jview /cp ..." with "start /B wjview /cp ..." so that it looks like:

      start /B wjview /cp lib\hotfix.jar;lib\...

   d. remove the line that contains "pause" at the end of the file, the complete file should now look something like:

      ```
      @ECHO OFF
      cd ..
      start /B wjview /cp lib\hotfix.jar;lib\servicepack.jar;lib\...
      cd bin
      ```
5. Apply the service pack that your current Service Desk environment is using.

6 Ensure that all users that you want to be able to use the Service Desk 4.5 client on Citrix are able to access the Citrix machine. For example, add them to the Citrix machine Windows "Users" group.

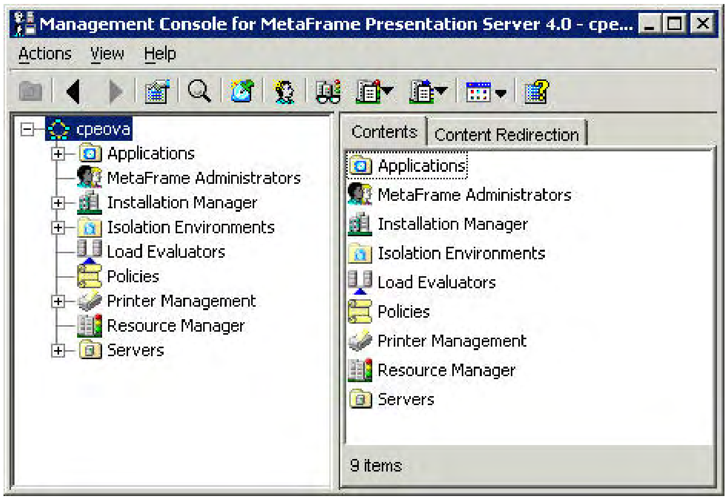
7 Now it's time to publish the application within Citrix, see Publishing *Service Desk 4.5 Client Example.*

Publishing Service Desk 4.5 Client Example

Open the Presentation Server Console:

- **Start** → **Programs** → **Citrix** → **Management Consoles** → **Presentation Server Console**

Figure 2



Publish a new application

- **Actions** → **New** → **Published Application**

Figure 3

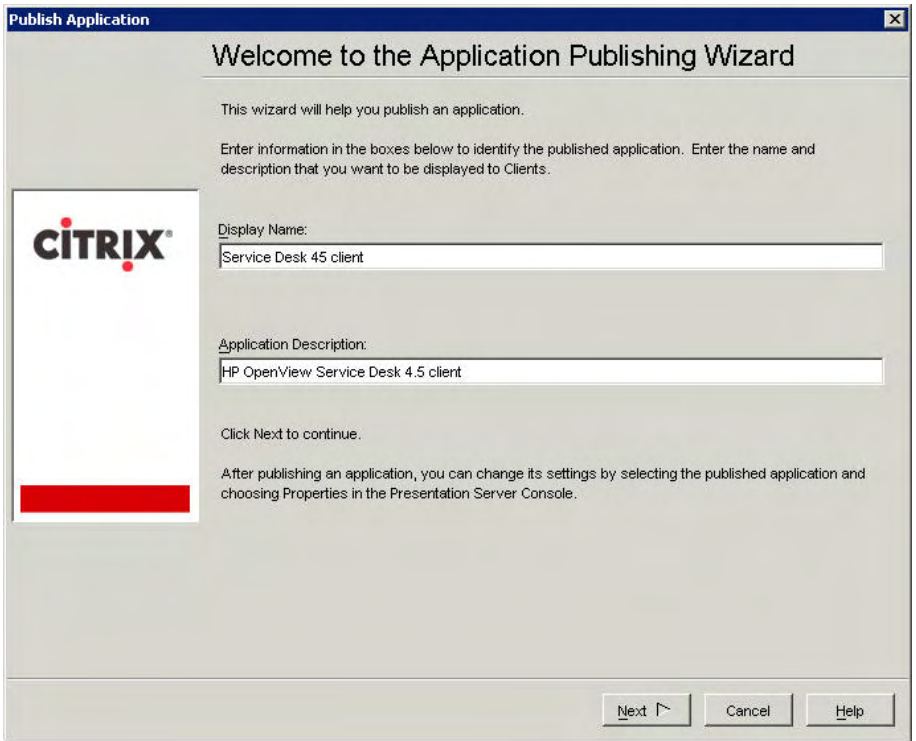


Figure 4

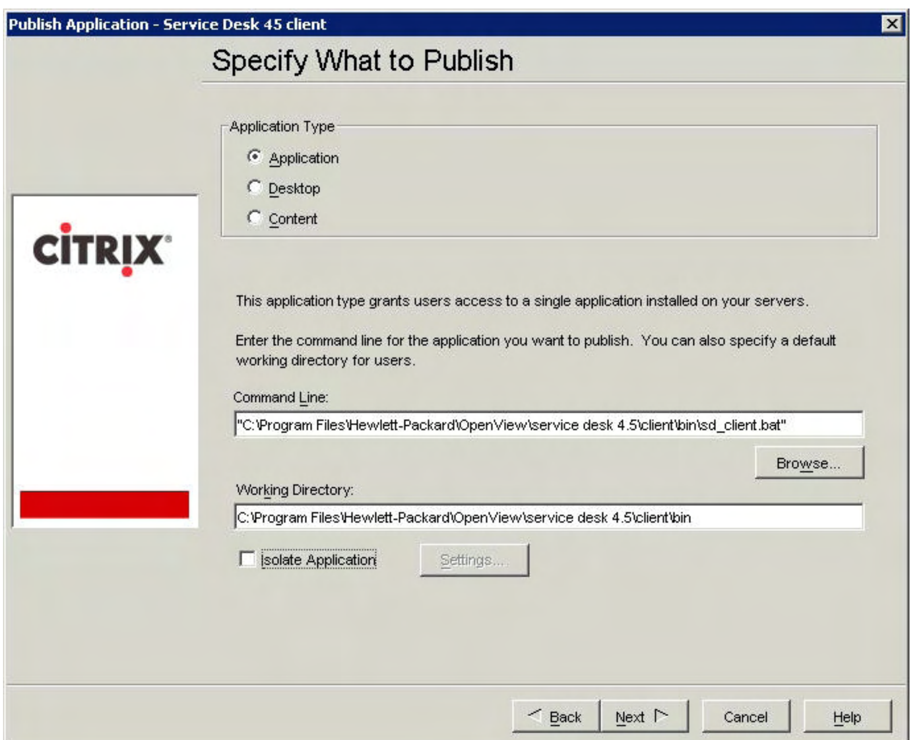


Figure 5

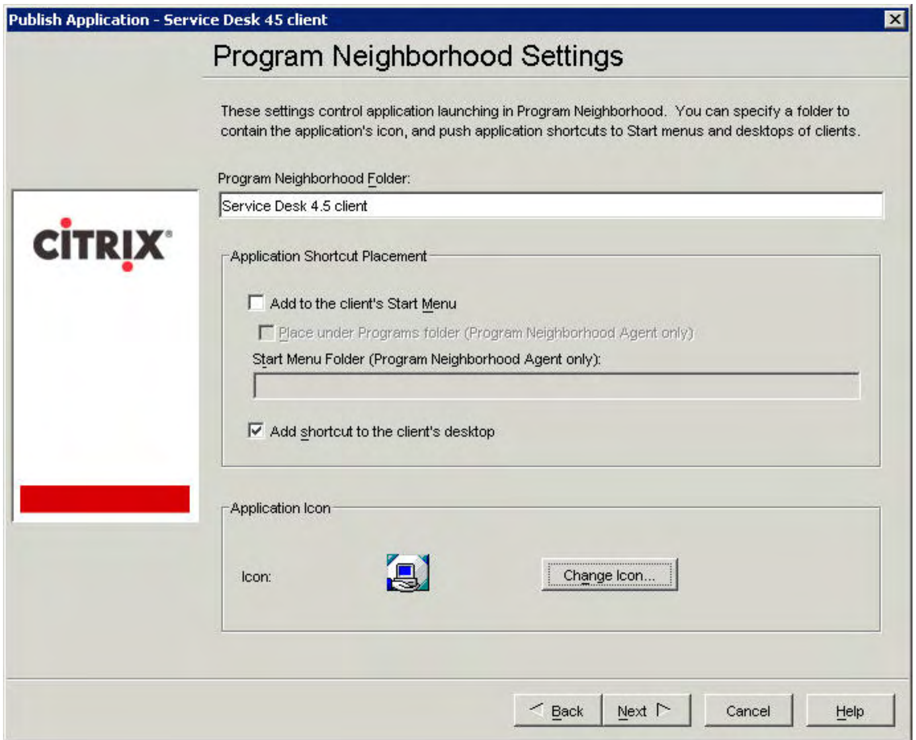


Figure 6

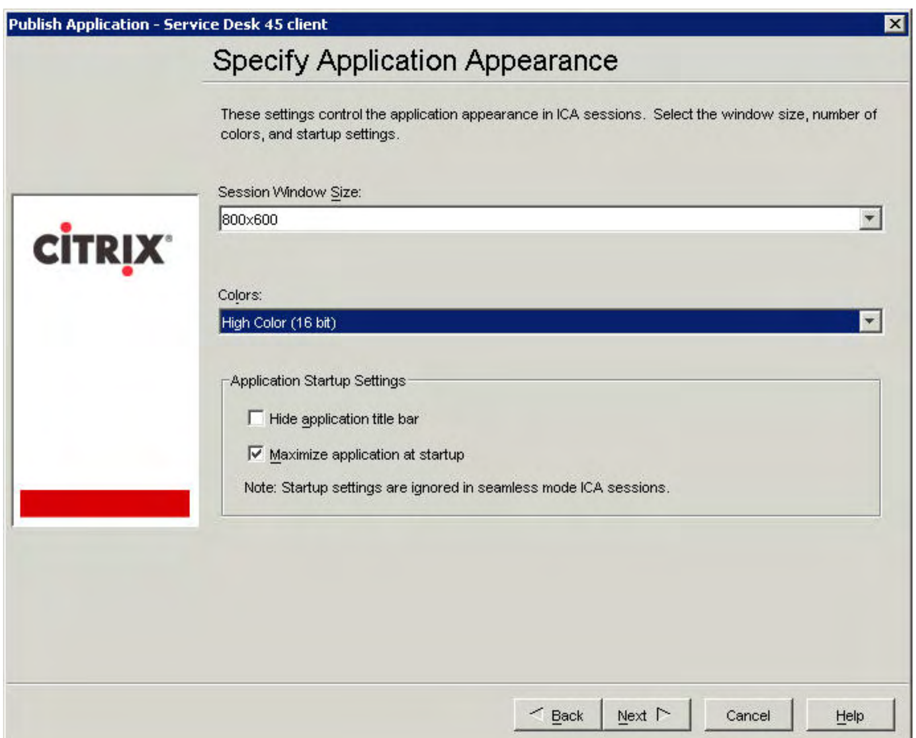


Figure 7

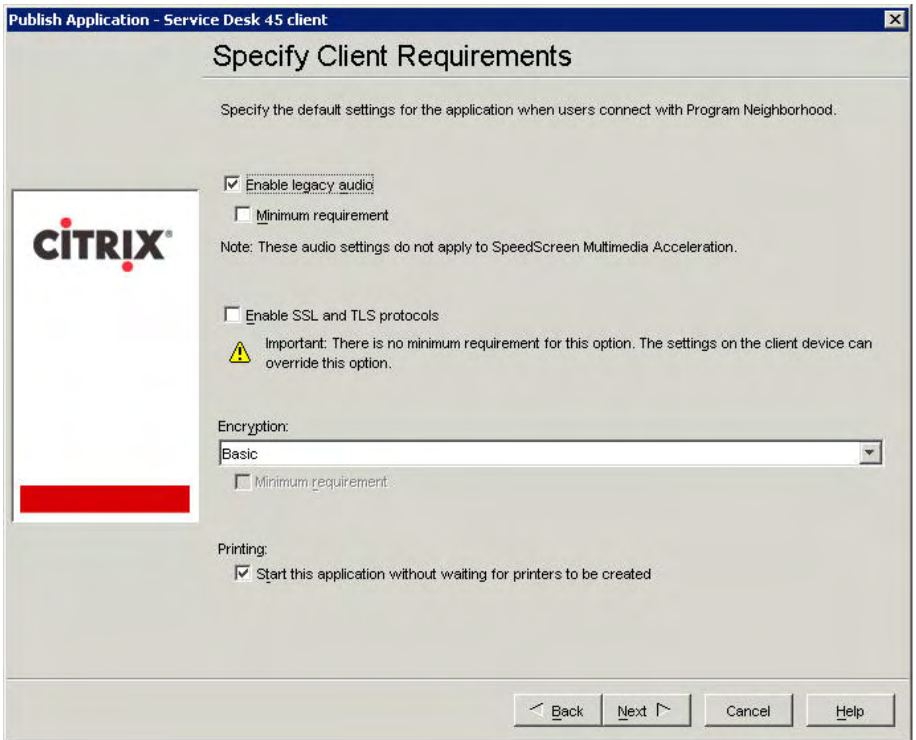


Figure 8

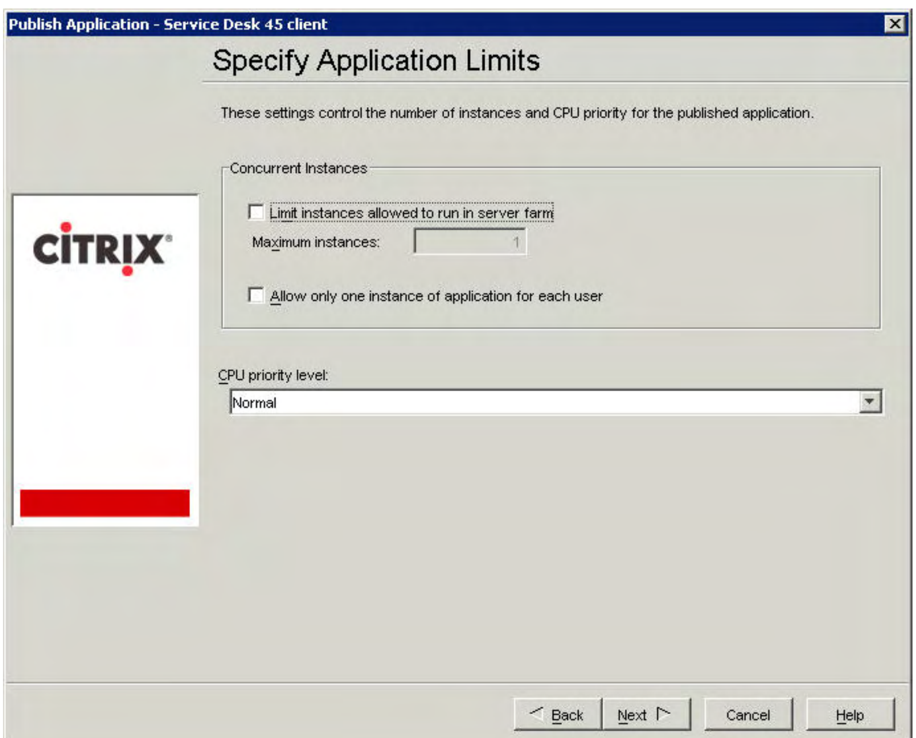


Figure 9

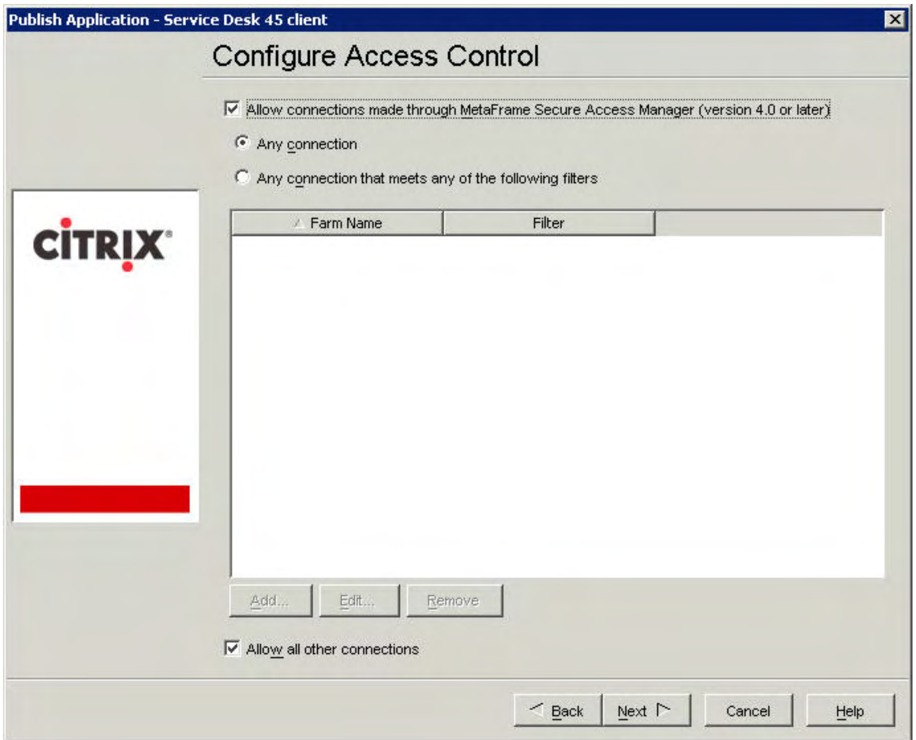


Figure 10

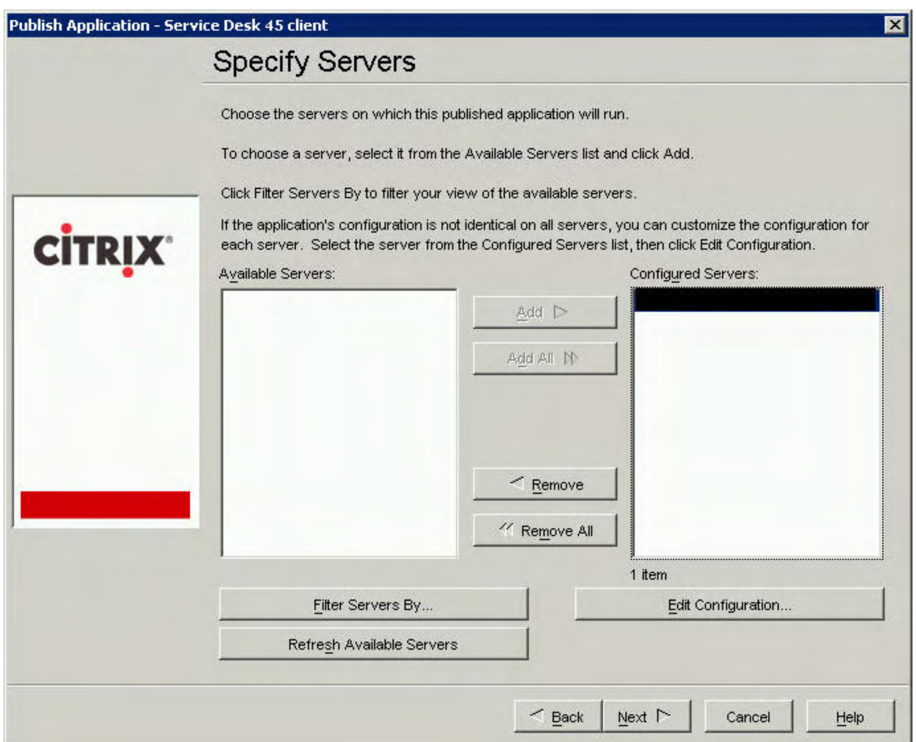
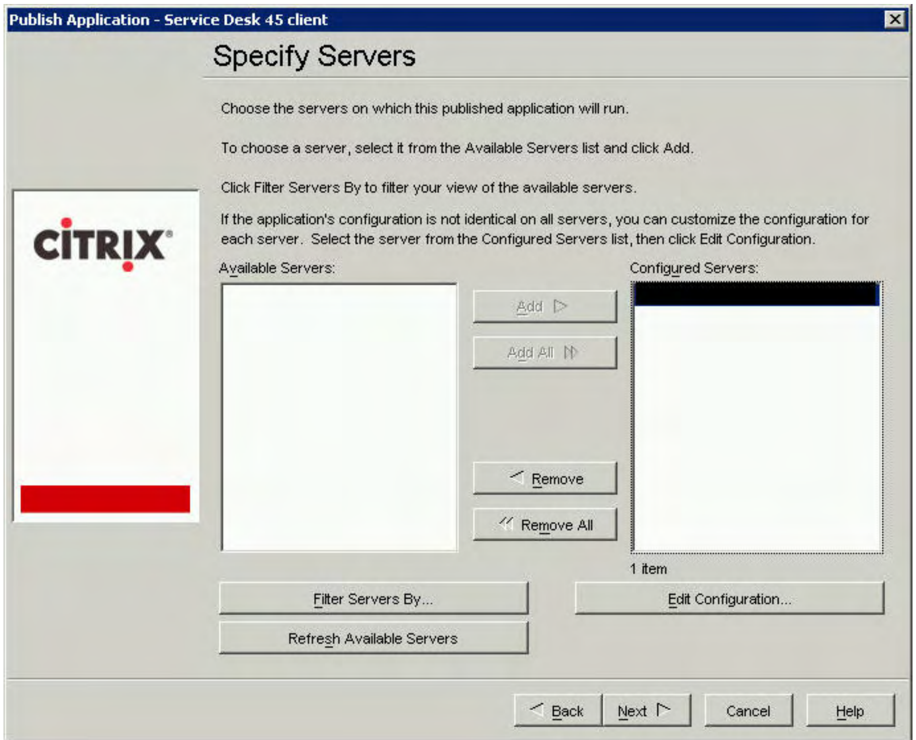
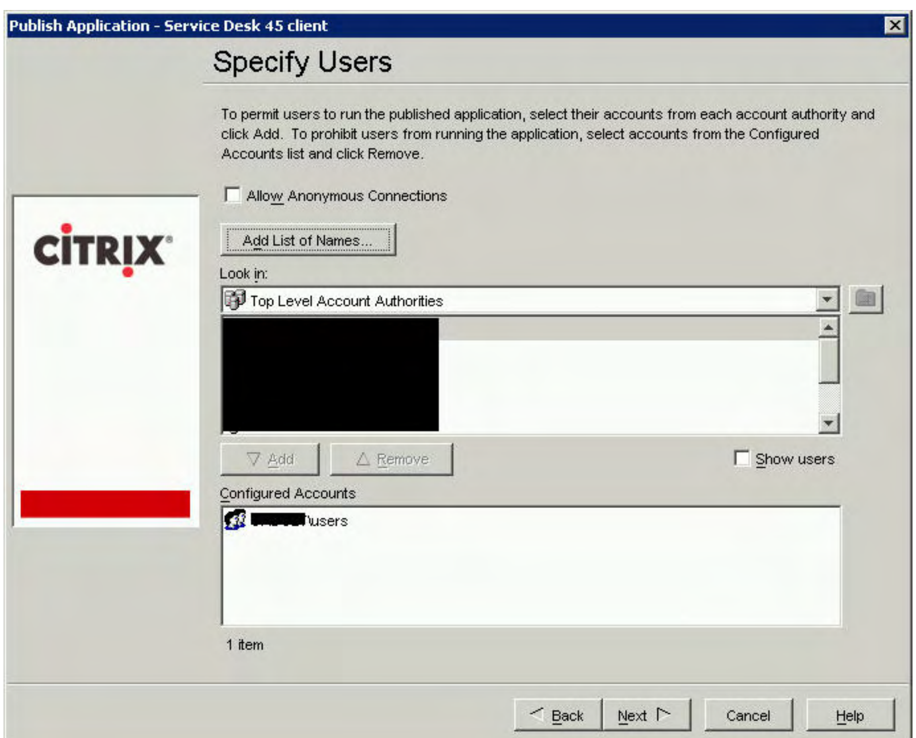


Figure 11



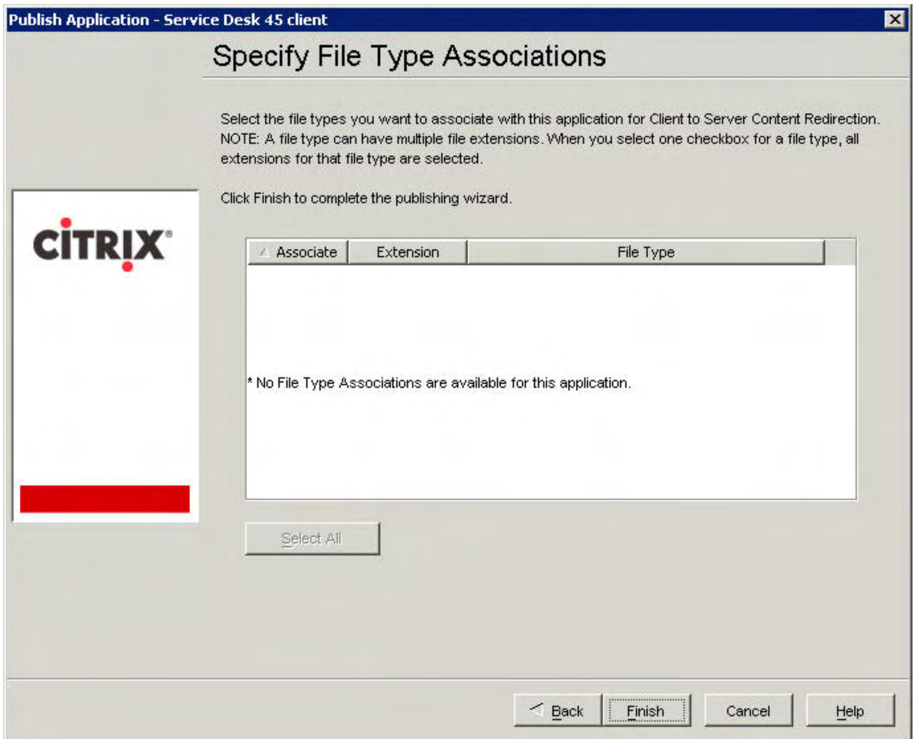
Make sure your Citrix server is available in the Configured Servers list.

Figure 12



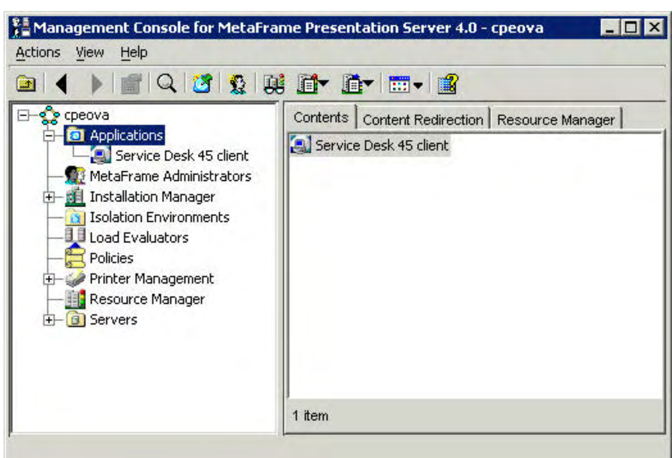
Add users to the published application that will be allowed to see and start the published application. In the above example the <machine name>\Users group was added. You can obviously also add domain groups and / or users.

Figure 13



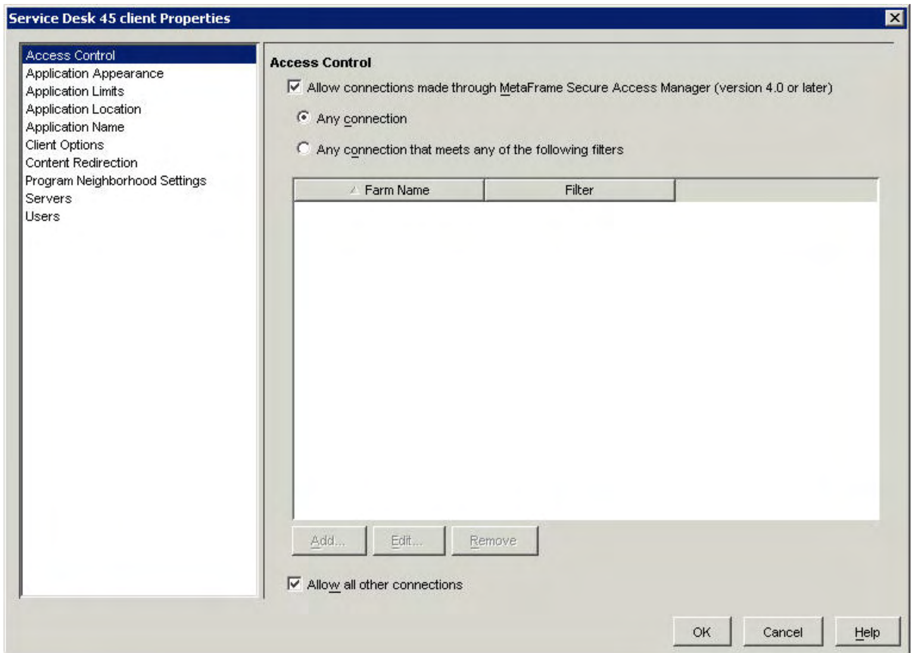
Checkout the published applications:

Figure 14



If you want to modify properties after finishing the wizard you right click the published application and select **Properties.**

Figure 15



Installing Microsoft Java Virtual Machine on Windows 2003

By default, there is no installation of the Microsoft Java Virtual Machine (MS JVM) on Windows 2003 versions. Thus, the MS JVM will have to be installed before the Service Desk 4.5 client will be able to function. To do so, to installation packages are required. The original MS JVM, and an updated version of the MS JVM. Perform the following steps to ensure the MS JVM is installed succesfully:

1 Install the msjavx86.exe package found in the <unpacked service pack location>\windows\tools\msjvm folder.

> Do NOT reboot yet after installing the package. This should be done later to avoid multiple reboots.

2 Install the MSJavWU_8073687b82d41db93f4c2a04af2b34d.exe package found in the <unpacked service pack location>\windows\tools\msjvm\win2003 folder.

3 Reboot the machine.

4 After the machine is rebooted login and open a command prompt, then type the following:

type jview and press enter.

```
Microsoft Windows [Version 5.2.3790]
(C) Copyright 1985-2003 Microsoft Corp.

C:\Documents and Settings\<user name>>jview
Microsoft (R) Command-line Loader for Java Version 5.00.3810
Copyright (C) Microsoft Corp 1996-2000. All rights reserved.

Usage: JView [options] <classname> [arguments]

Options:
```

```
/? displays usage text
/cp <classpath> set class path
/cp:p <path> prepend path to class path
/cp:a <path> append path to class path
/n <namespace> namespace in which to run
/p pauses before terminating if an error occurs
/v verify all classes
/d:<name>=<value> define system property
/a execute AppletViewer
/vst print verbose stack traces (requires debug classes)
/prof[:options] enable profiling (/prof:? for help)

Classname:
.CLASS file to be executed.

Arguments:
command-line arguments to be passed on to the class file
```

Limitations

Although the Service Desk 4.5 client is supported for Citrix Metaframe Presentation Server 4.0 there are limitations. These limitations will prevent certain client functionality to work and also mean that that functionality will not be supported. This document will explain what limitations there are and describe possible workarounds.

- Although the Citrix Metaframe Presentation Server can be used on Unix based operating systems, this is NOT supported for using the Service Desk 4.5 client because there is no possibility to run the Service Desk 4.5 client on Unix based operating system because the Microsoft Java Virtual Machine is required.
- Flashing tray icon: will not work in a Citrix environment because the application runs on the Citrix server, and not on your Citrix client machine.
- Flashing tray icon: a flashing icon appears in your Windows tray bar with a sound alerting you that you have new unopened items in Service Desk that need your attention.
- Attachments: it is impossible to drag and drop attachments in for example service calls, workorders, etc. The "add attachment" functionality should be used for this functionality. This is a Citrix limitation for all published applications.
- Spelling checker: will only work if Microsoft Office, in specific Microsoft Word, is installed on the Citrix Metaframe Presentation Server 4.0. It does not have to be published in order to work!

How To

Change the default C drive from the local Citrix server drive to the local Citrix client C drive: use the Remap Drive Utility. This Utility is included within the Citrix Metaframe Presentation Server 4.0 CD. For example, the user would see the local client drive on the C drive when adding an attachment using Service Desk 4.5.

Install and Configure the Citrix Metaframe Access Client on Windows

This document explains how you can install and configure the Citrix Metaframe Access Client on Windows.

Requirements

Changes made to the client package and or installation could potentially cause differences to occur in the screenshots shown. Please refer to the official Citrix documentation on how to properly install and configure the Citrix Metaframe Access Client and to learn about additional requirements or options.

Make sure you have the correct Citrix Metaframe Access Client for your Citrix Metaframe Presentation Server. For example download the latest version from http://www.citrix.com**.** At the time of writing the location to download this package is: http://www.citrix.com/English/SS/downloads/details.asp?dID=2755&downloadID=15015&pID=12752

Installing Citrix Metaframe Access Client

Install the Citrix Metaframe Access Client: `ACM-en.msi.`

Figure 16

Figure 17

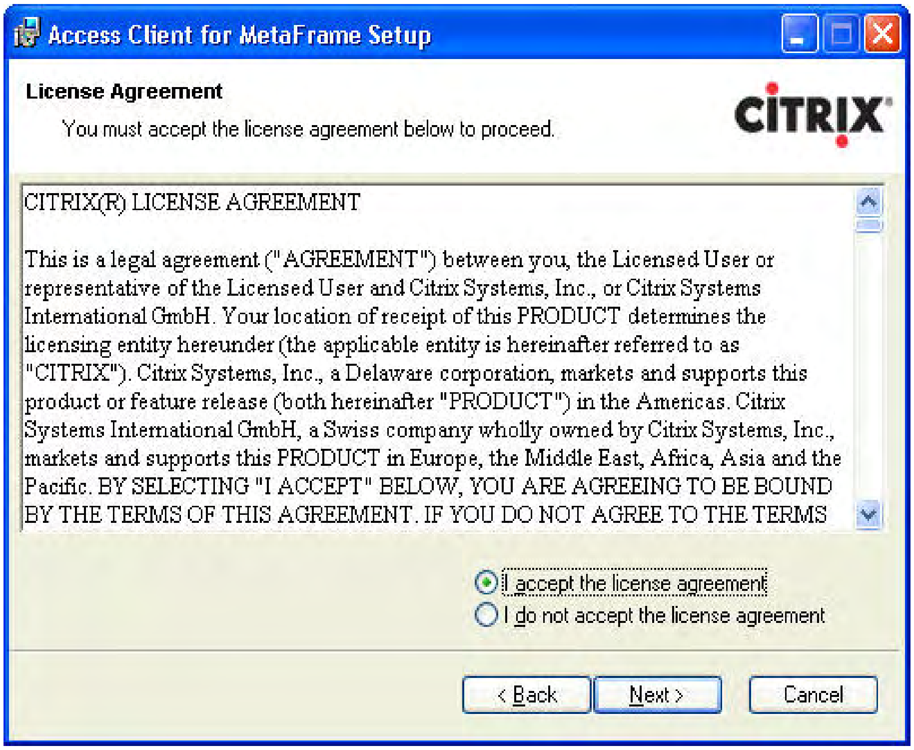
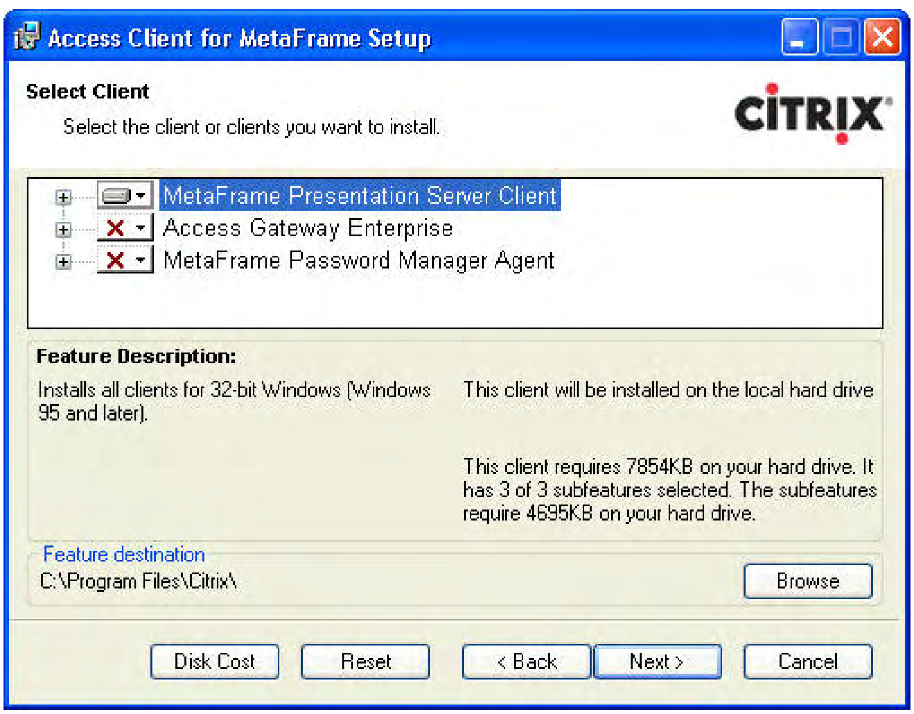


Figure 18



➤ Only install the "MetaFrame Presentation Server Client" part.

Figure 19

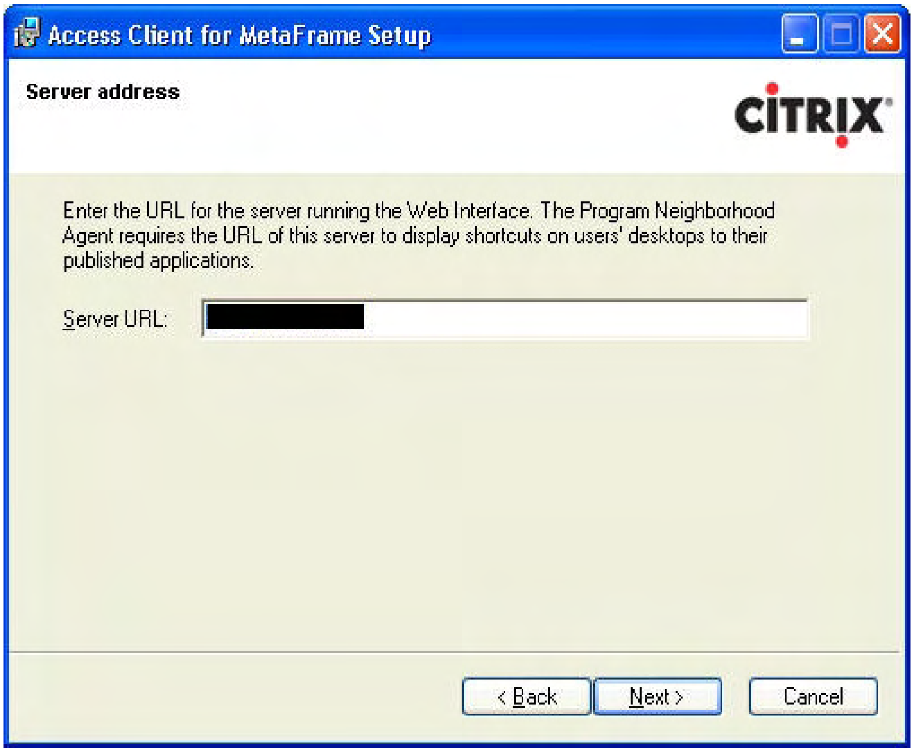


Figure 20

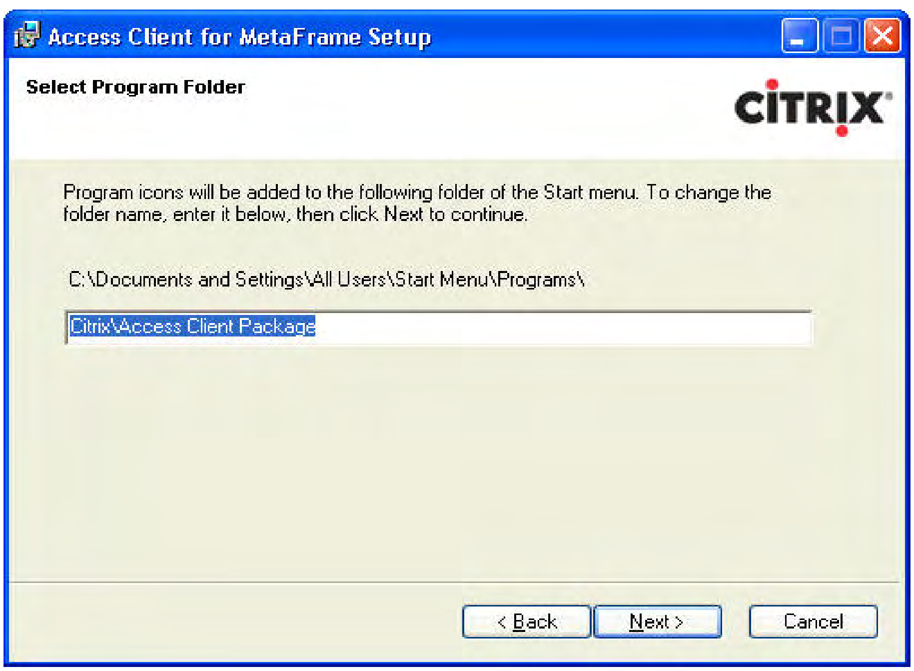


Figure 21

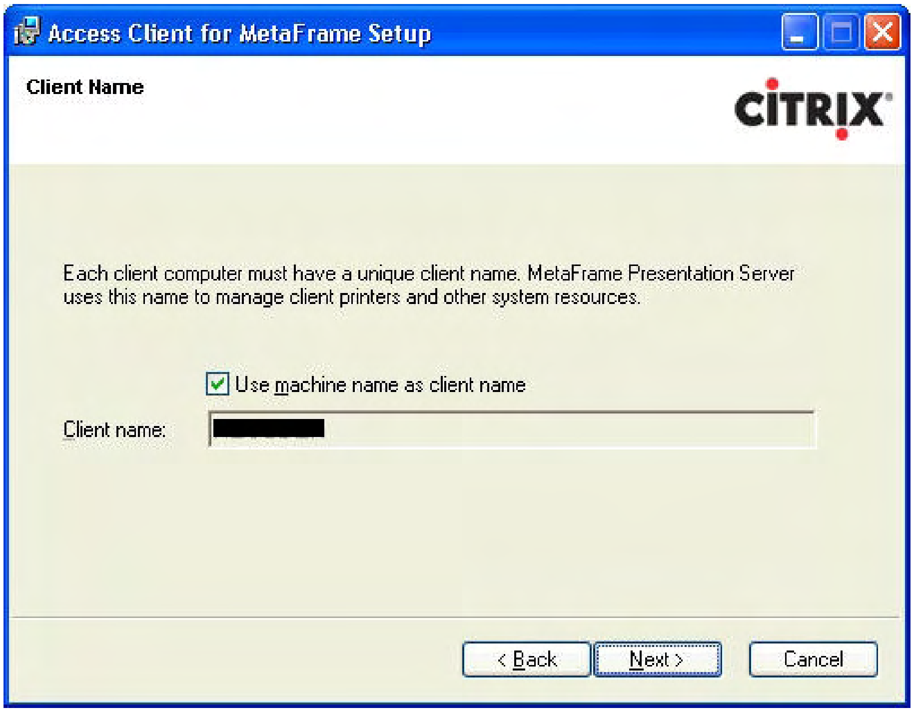


Figure 22

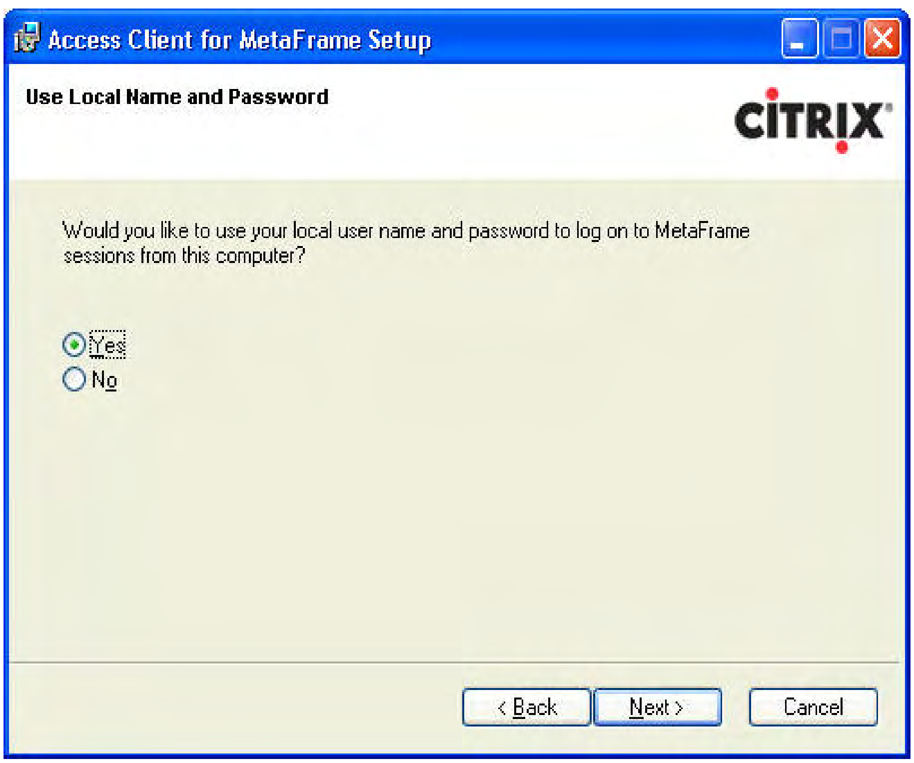


Figure 23

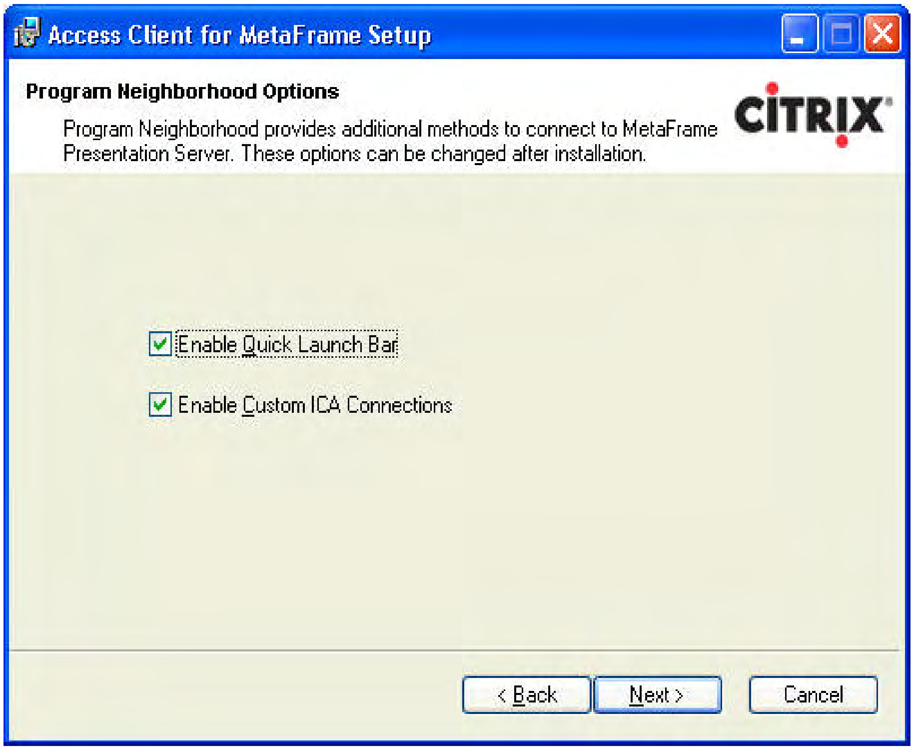
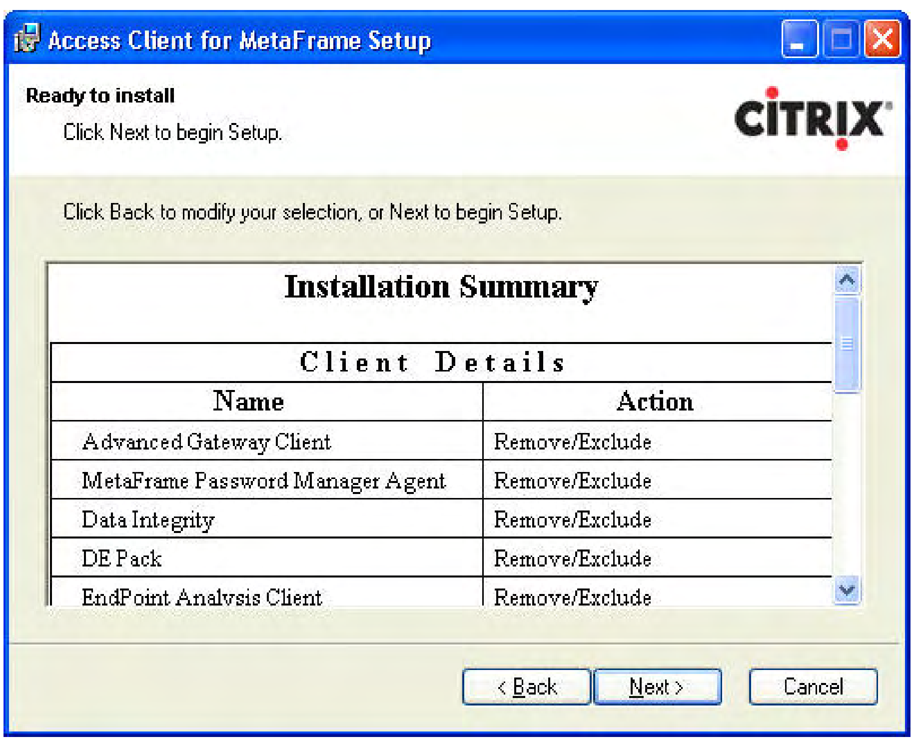


Figure 24

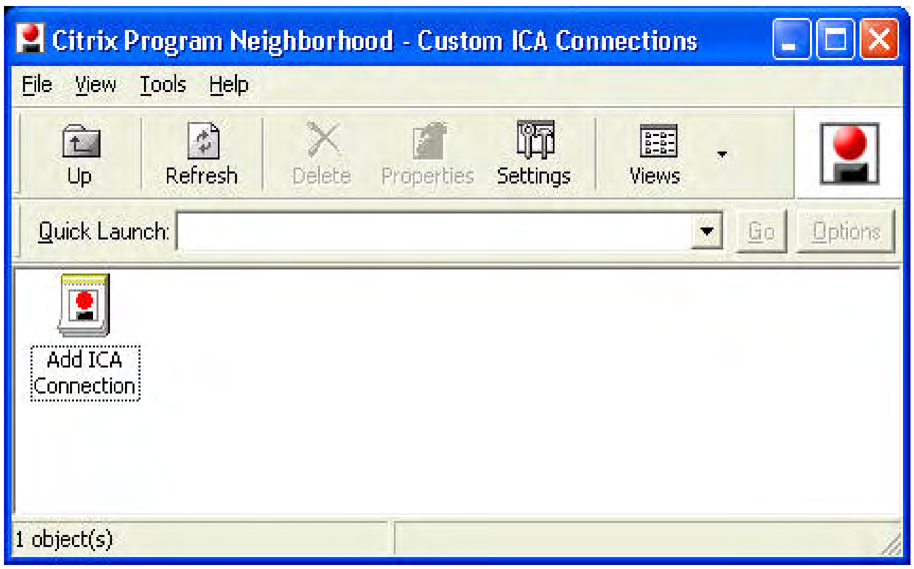


➤ REBOOT the machine after installing this package!

Configure the Citrix Metaframe Access Client and Start Service Desk

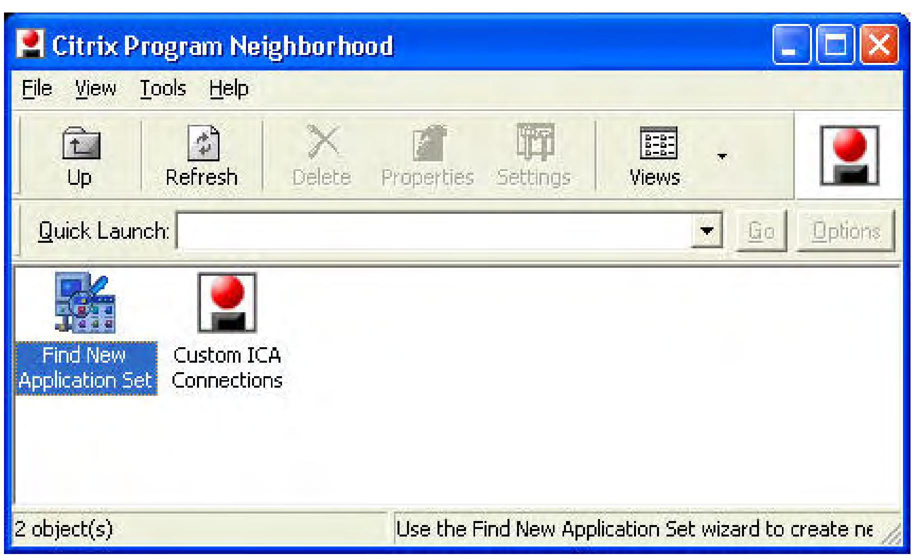
From the desktop, start the "Citrix Program Neighborhood".

Figure 25



Click **Up.**

Figure 26



Start **Find New Application Set.**

Figure 27

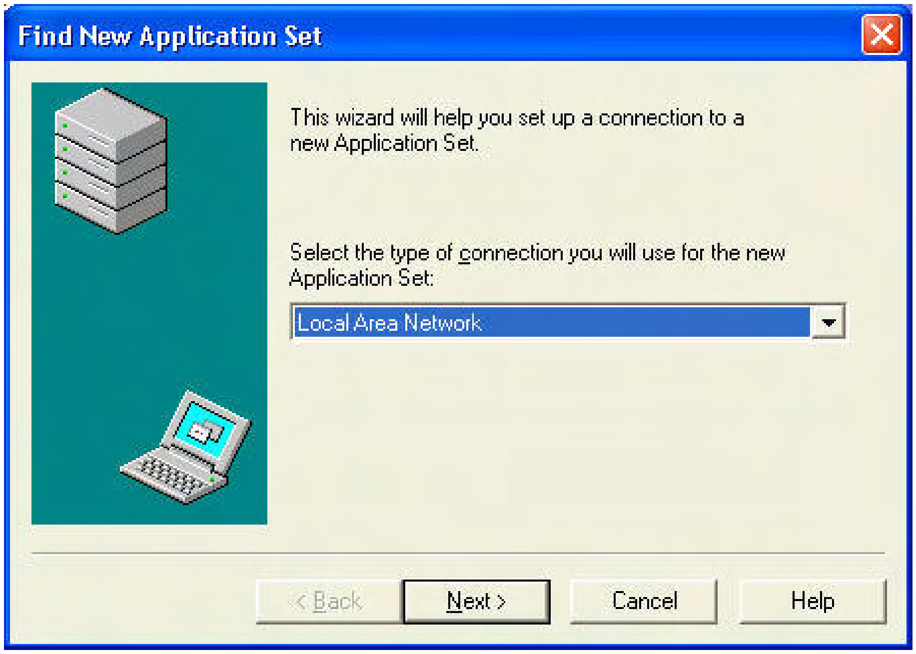
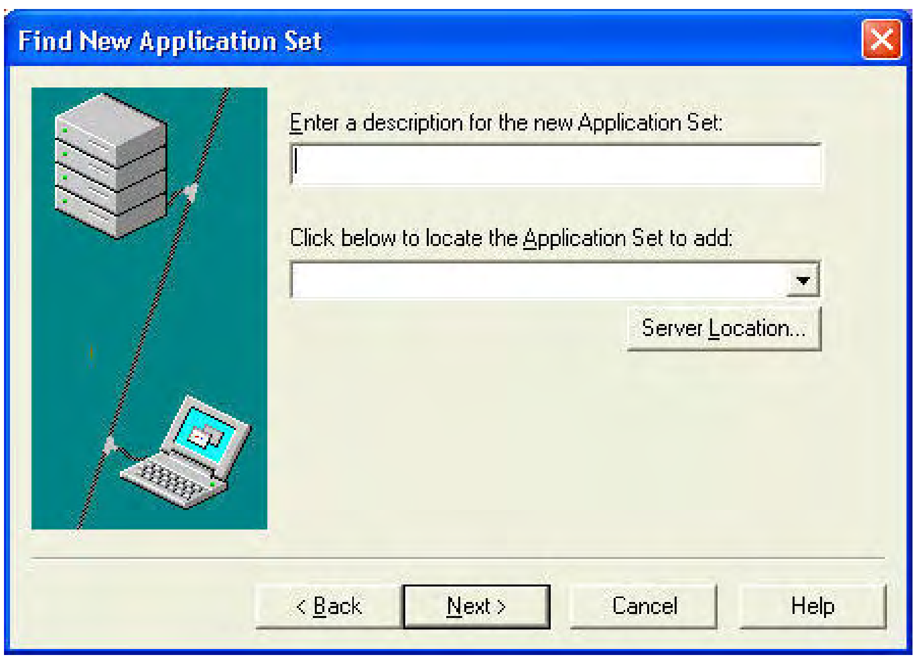
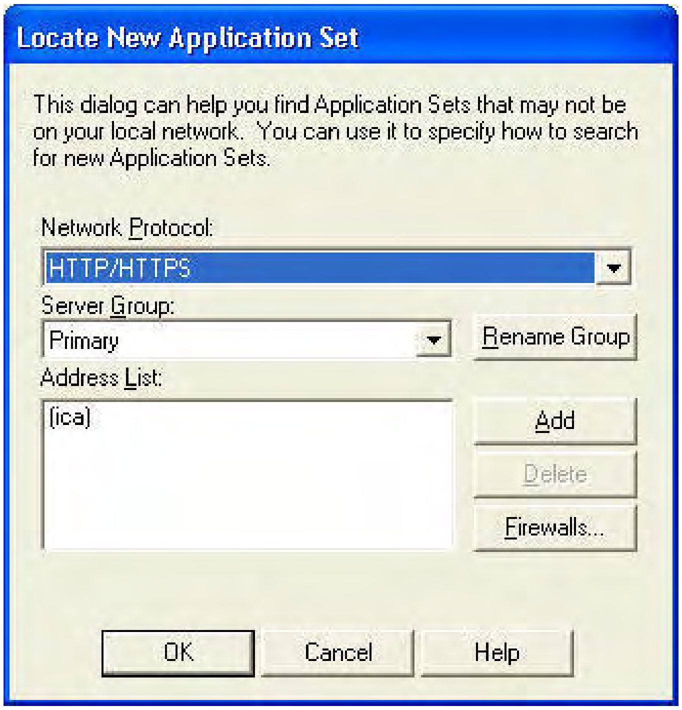


Figure 28



Click **Server Location.**

Figure 29



Click **Add.**

Figure 30

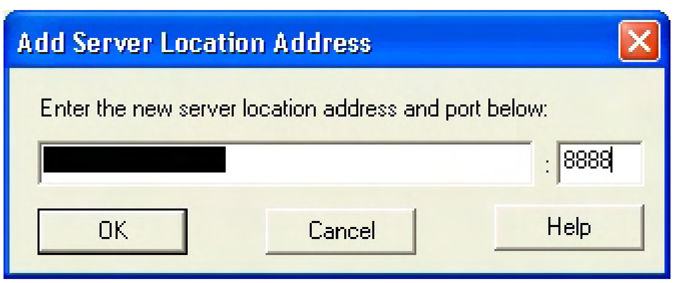
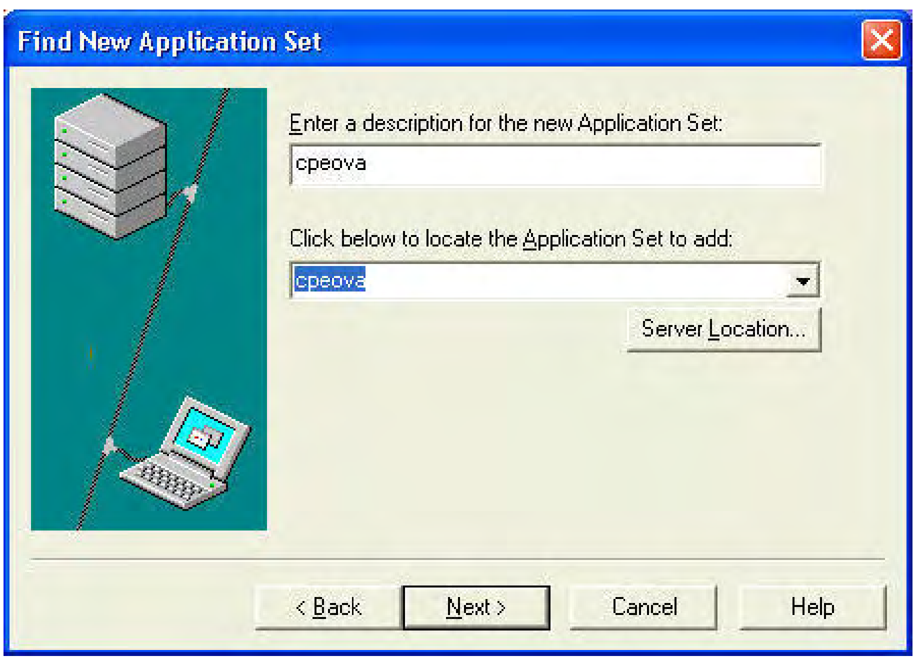


Figure 31

Figure 32



Click on the dropdown list button next to the **Click below to locate the...** field.

Select the Citrix server name from the list. In this example, it is "cpeova".

Figure 33

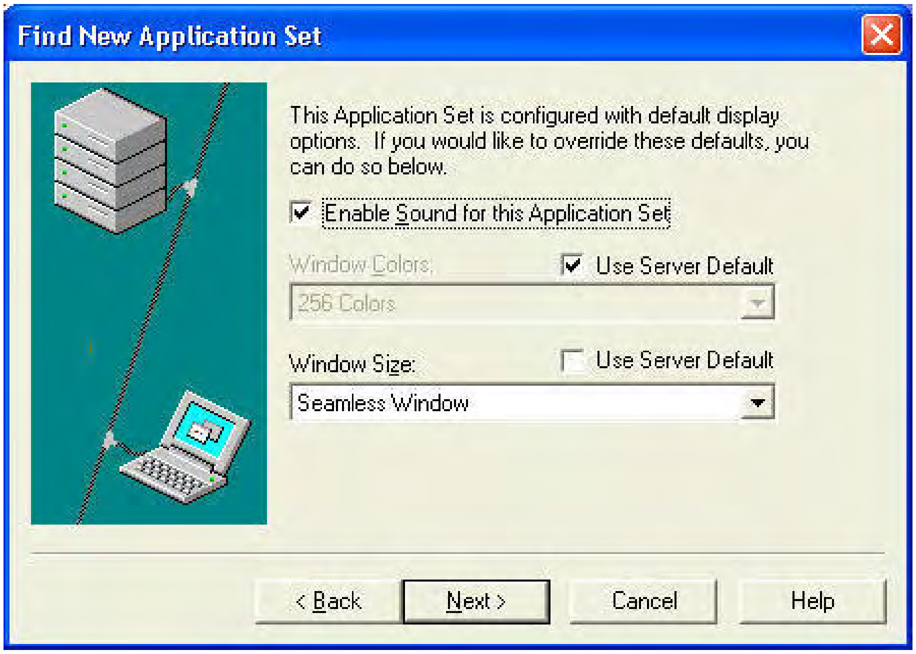
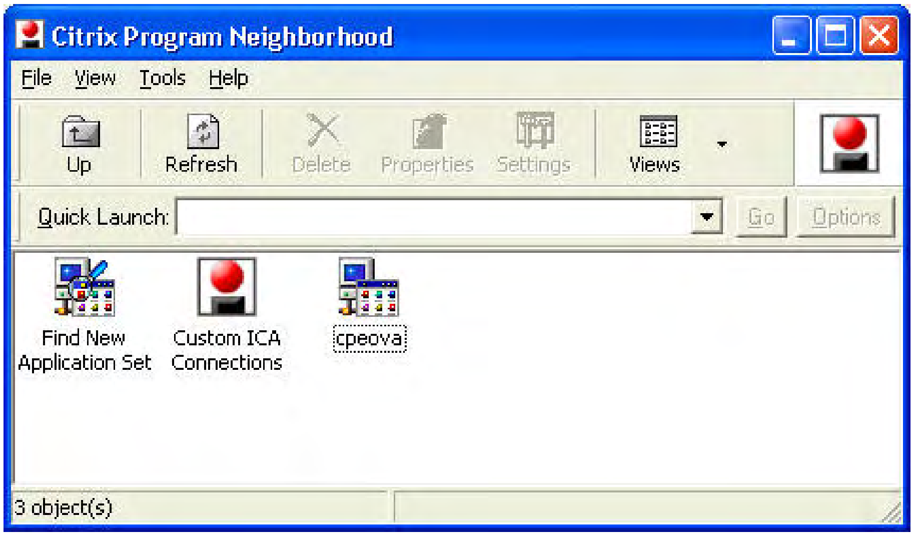


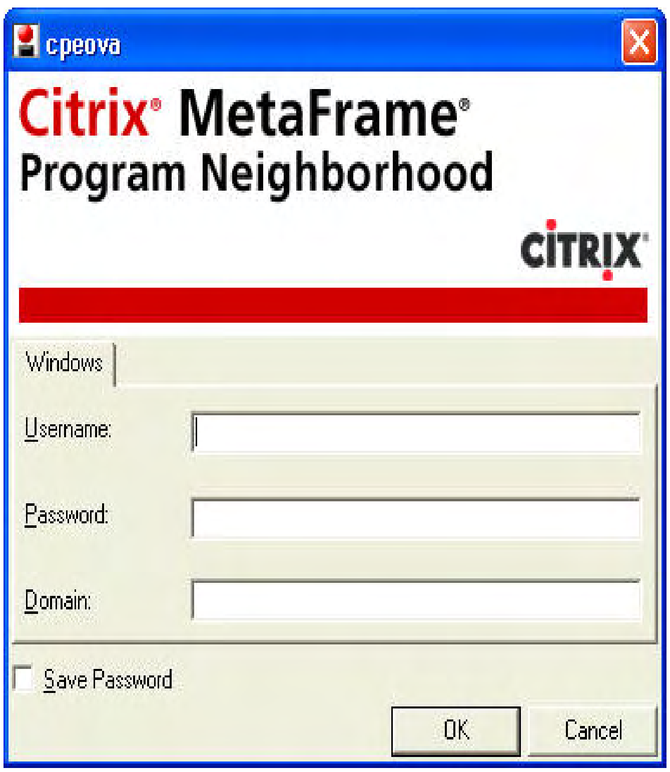
Figure 34

Figure 35



Open the application set. In this example it is "cpeova".

Figure 36



Fill in your Windows login details. See example below:

Figure 37

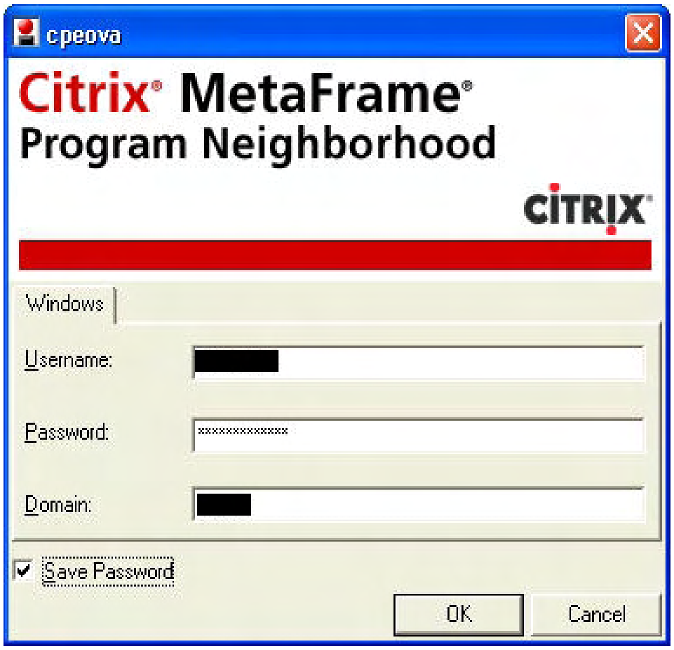
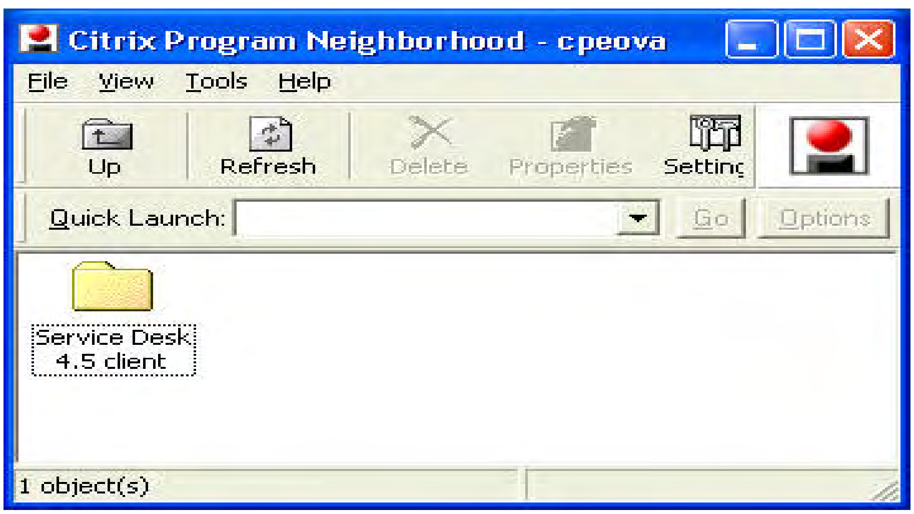
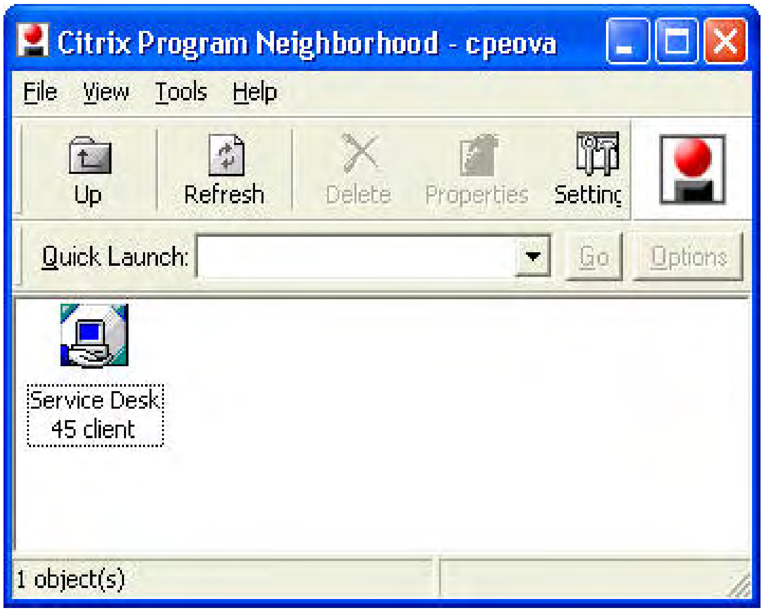


Figure 38



Open the **Service Desk 4.5 client** folder.

Figure 39



Double-click the **Service Desk** shortcut to start the application or close the application and double click the **Service Desk 45 client** shortcut on your desktop of you choose to publish the application with the "Add shortcut to the clients desktop" option enabled.

Please refer to the official Citrix documentation on how to properly install and configure the Citrix Metaframe Access Client and to learn about additional requirements or options.

OpenView Operations for Windows 7.5 Cluster Integration

Introduction

OVOW 7.5 is the first release of OVO to support installation and operation of the Management Server in a failover configuration running a Microsoft Windows 2003 Cluster.

From service pack 16 Service Desk starts to support integration of OVOW (OpenView Operations for Windows) 7.5 cluster and OVSD (OpenView Service Desk) 4.5. The document is an addendum to *OpenView Operations Integration Administrator's Guide* which provides additional information about how to setup the integration.

The document assumes that you are an experienced OVOW administrator and you understand OVOW cluster concepts and terminology.

Getting Started

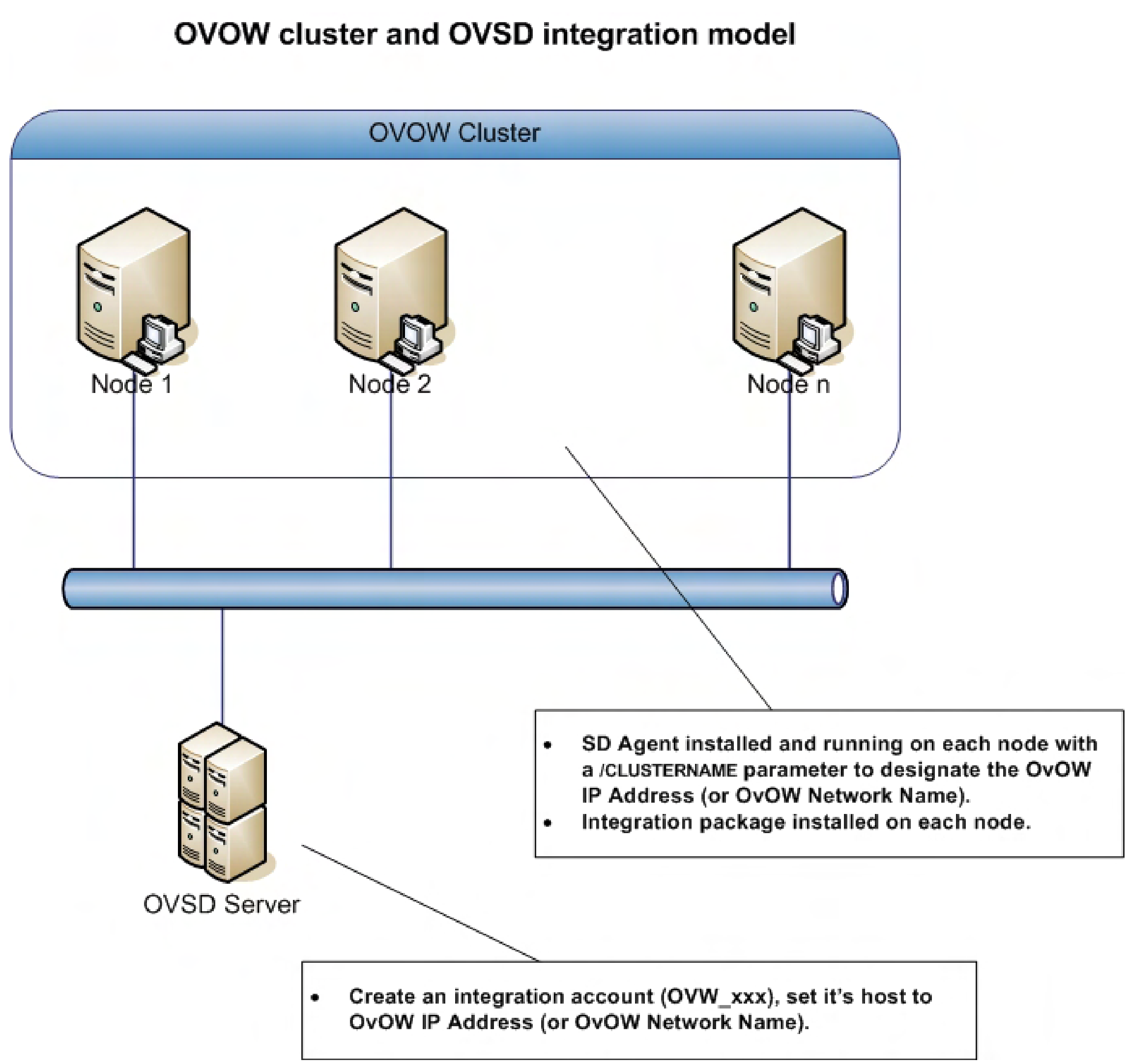
Integration Model

OVOW forwards messages to OVSD as Incident by using an integration account (OVW_xxx) of OVSD, then OVSD sends back Annotation/Acknowledge actions to OVOW. Since OVSD can not execute these actions on OVOW directly, it dispatches these actions to OVSD Agent (hosted on OVOW side) to exeucte them. OVSD Server locates the OVSD Agent according the Host field of OVW_xxx.

OVOW cluster acts as a virtual server and has a unique IP Address/Network Name (say **ovow_ip_address**). The virtual server runs on one cluster node at any time, on which the **ovow_ip_address** is bound. If failover occurs, the virtual server switchs to another node, **ovow_ip_address** transfers as well. Therefore, by running an OVSD Agent on each cluster node and setting OVW_xxx's Host to **ovow_ip_address**, OVSD Server can always find the "active" OVSD Agent through **ovow_ip_address**. In that way, actions can be sent to OVOW no matter when or how failover occurs.

Addtionally, OVSD Agents (running on each node) broadcast a kind of "alive" message (contains **ovow_ip_address**) periodically. Every time OVSD Server gets the message, it resends failed action(s) (if any) belong to that address.

Figure 40

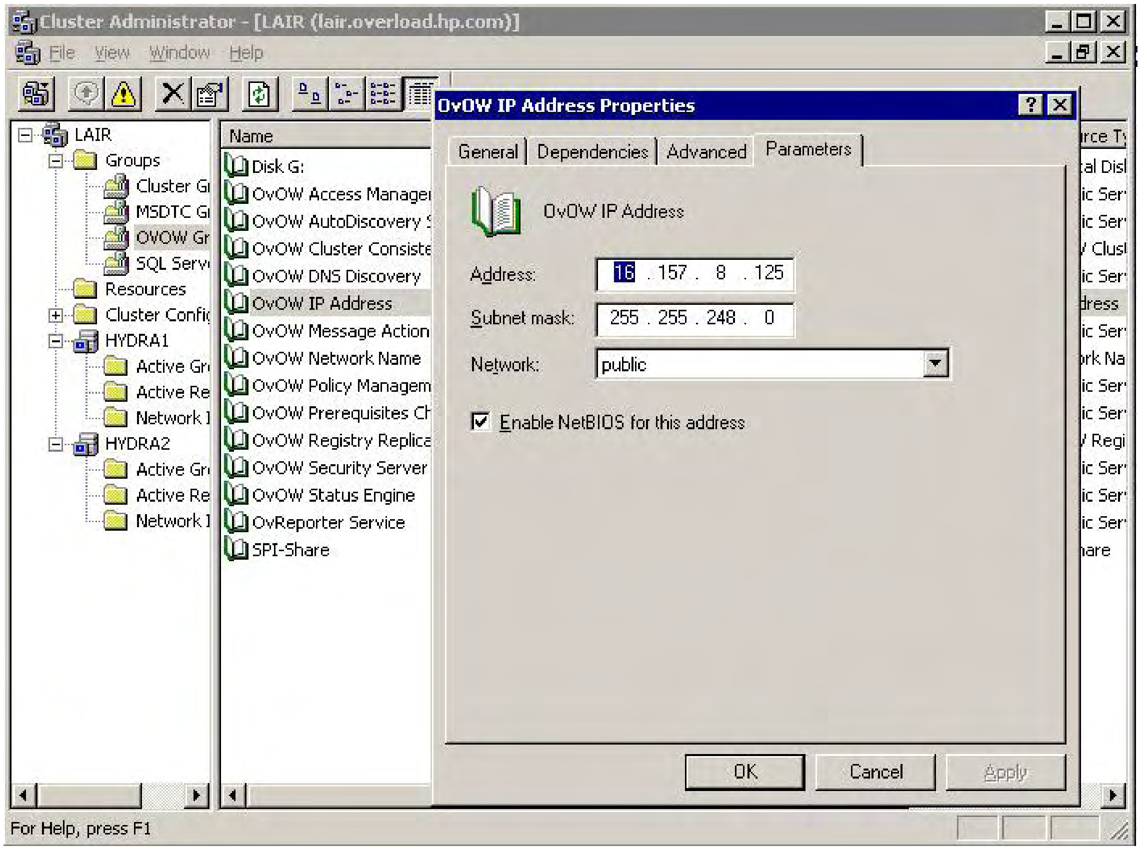


Setup Steps

Basic steps can be found in *OpenView Operations Integration Administrator's Guide*, here are **additional** steps you must perform when integrating OVSD with an OVOW cluster.

1 Get **ovow_ip_address**, you can find it on the Windows Cluster Administrator console, it resident in the Parameters tab page of OvOW IP Address or OvOW Network Name entry.

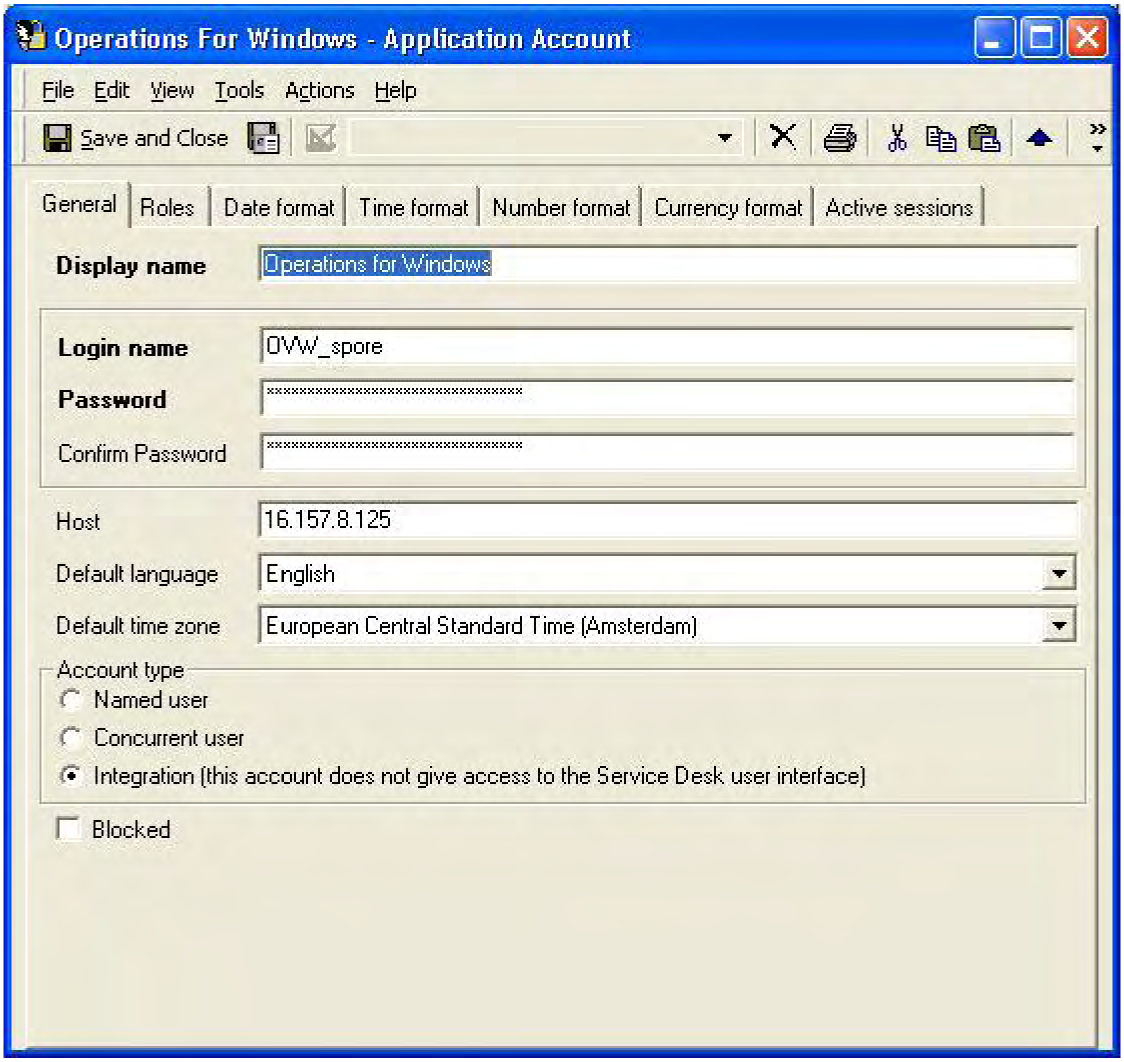
Figure 41



2 After installing OVSD Agent, modify `%SD_AGENTHOME%\bin\installagentservice.bat` and place "`/CLUSTERNAME=ovow_ip_address`" after "`-start com.hp.startup.Bootstrap -params com.hp.ifc.ev.ag.sdalocal.AppAgent`", then install/reinstall it as a service. If you start OVSD Agent from command line, just append "`/CLUSTERNAME= ovow_ip_address`" to `%SD_AGENTHOME%\bin\sd_agent.bat` in command line (here, the double quotation marks are mandatory).

3 For the integration account OVW_xxx in OVSD, specify its Host field as `ovow_ip_address`.

Figure 42



4 An optional step is to specify a "`/INTERFACE=xxx`" parameter for OVSD Agent. Since there are usually multi network interfaces on a node in a cluster environment, you can add a "`/INTERFACE=one_interface_ipaddress`" for Agent to specify which interface is used to send the "alive" messages. Otherwise, the messages will be sent out through every interface. Usually, you can specify `one_interface_ipaddress` as the node's real address (which must be accessible from OVSD Server).

Known Errors and Limitations

Messages from SD to OVOW May Delay During OVOW Failover

During OVOW failover, messages (which need to be sent from SD to OVOW) pile up on OVSD side and they will be sent after OVOW failover is finished. A worse but very rare scenario is, if the OVOW failover happens between a message handling process (from OVSD to OVOW), it may cause subsequent messages delay for about 5 minutes even when the OVOW failover finishes in a short time.

Message Loss if Active Node Crash

If an active OVOW node crashes while massive messages are being sent from OVSD to OVOW, OVOW can not guarantee all messages will get executed though it can be proved that those messages have arrived.

Troubleshooting

Can't Start OVSD Agent Service

Check that environment variable SD_AGENTHOME is set correctly. If not, most probably you have not restarted after installing OVSD Agent as it requires.

If that environment variable is correct, please double check whether there is any error in `installagentservice.bat`.

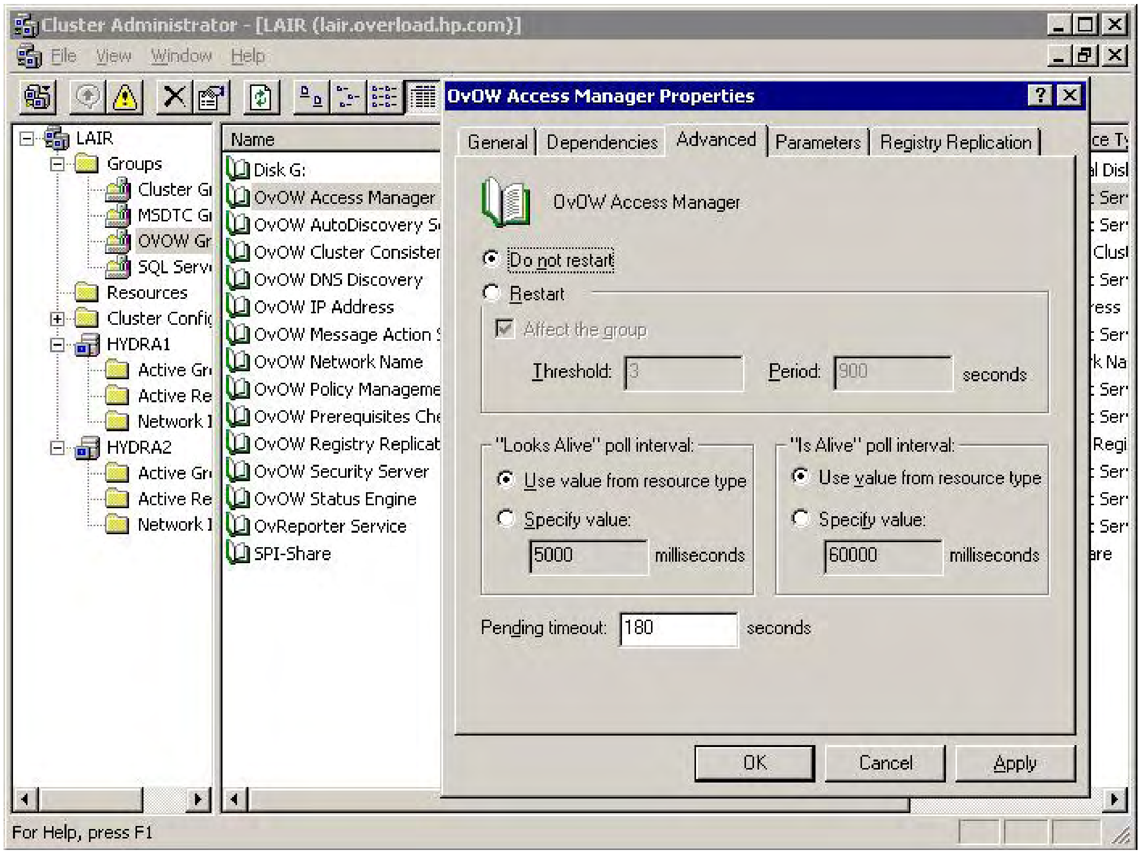
SD Agent Can't Parse Parameter Well From Command Line

If you start OVSD Agent from command line, you must enclose its parameter with double quotation marks if the parameter contains equality sign (=).

Can't Run OVSDSync.exe

The document of `OVSDSync.exe` tool says you must stop the `wmimgmt` service, but in a cluster environment you will encounter problem while you doing that, it complains that you could not stop some dependent services. To overcome this problem, you need set all those dependent OVOW services (in Windows Cluster Administrator) to "Do not restart" temporarily. Here is a screenshot:

Figure 43



Those dependent services are:

- OvOW Access Manager
- OvOW AutoDiscovery Server
- OvOW Message Action Server
- OvOW Policy Management and Deployment
- OvOW Prerequisites Check Server
- OvOW Status Engine

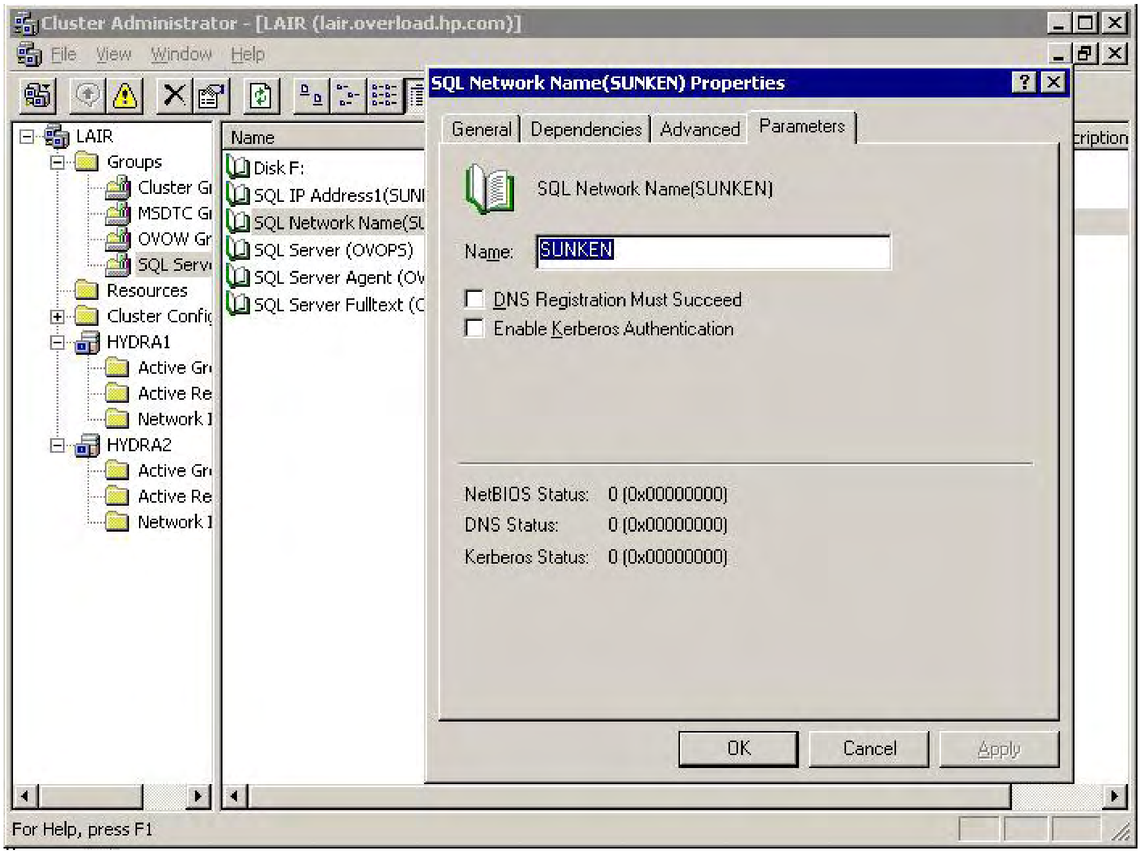
(Don't forget to restore the settings after you run OVSDSync successfully.)

Another problem is, by default, OVSDSync tries to connect to a local SQL Server instance, while in a typical cluster environment, the SQL Server usually resident in a virtual address. You need specify this address in OVSDSync.exe.config file:

```
<add key="OVSDSYNC.ConnectionString" value="integrated
security='true';initial catalog=openview;data source=Address of
Virtual SQL Server\OVOPS" />
```

The **`Address of Virtual SQL Server`** can be found on the Windows Cluster Administrator console:

Figure 44



Can't Send Message from OVSD to OVOW

Check that environment variable `SD_OVWHOME` is set correctly on OVOW nodes. If not, restart the OS or set it manually to let it point to `C:\OVOSD\`.

If that environment variable is correct, try to start OVSD Agent with the "`/DEBUG`" parameter and check the Agent log to see whether the job has been sent to OVSD Agent when the problem is reproduced.

Can't Install Service Desk Policy to OVOW

When installing Service Desk policy to OVOW, you may get an error like:

```
Ovpmutil cannot be started on inactive cluster node.
please review the output above to ensure success.
```

This is because you are trying to install policy on an inactive node. Installing the policy from an active node can solve the problem.

7 Service Pack 17

This chapter contains information previously provided with Service Pack 17.

Attachments with Database Rule Send Email Action

Introduction

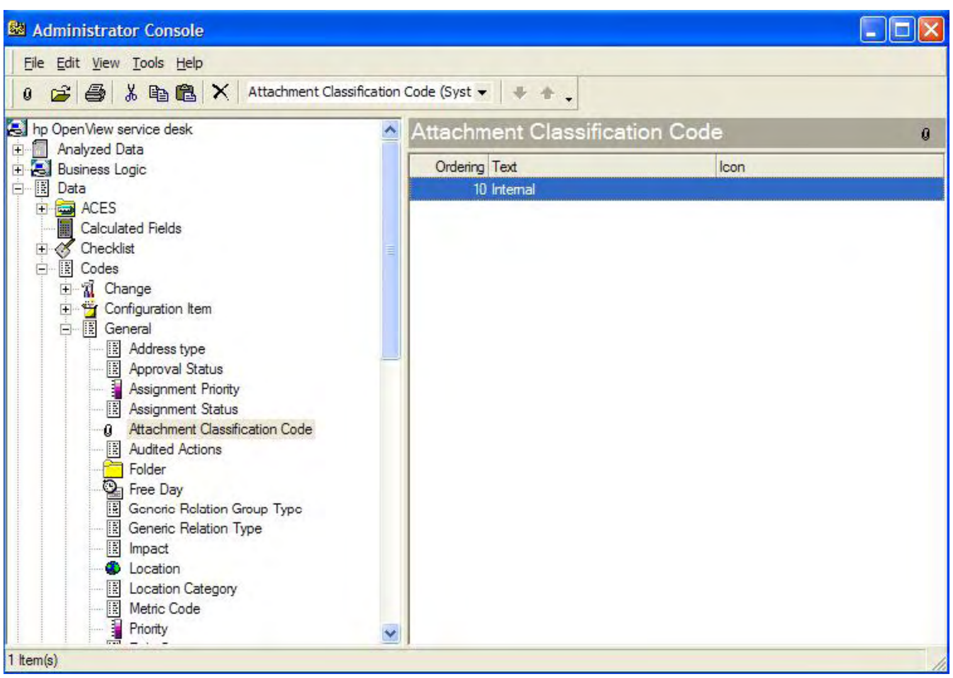
Attachment Classification Code is introduced to classify attachments. "Send e-mail message" action in Database Rule can send email with attachments of selected classifications.

Getting Started

Define attachment classification code

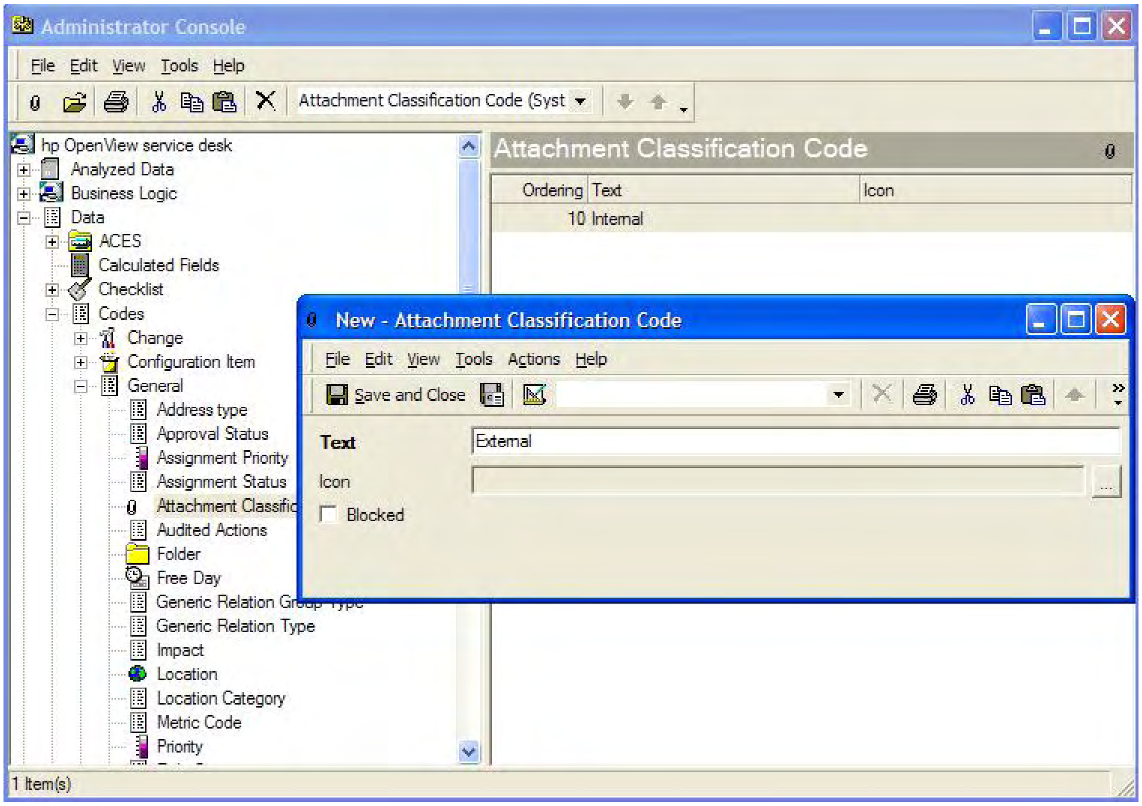
Logon with system administrator, open administrator console, go to **Data** → **Codes** → **General** → **Attachment Classification Code**.

Figure 1



Create an Attachment Classification Code, for example, "Internal" and "External"

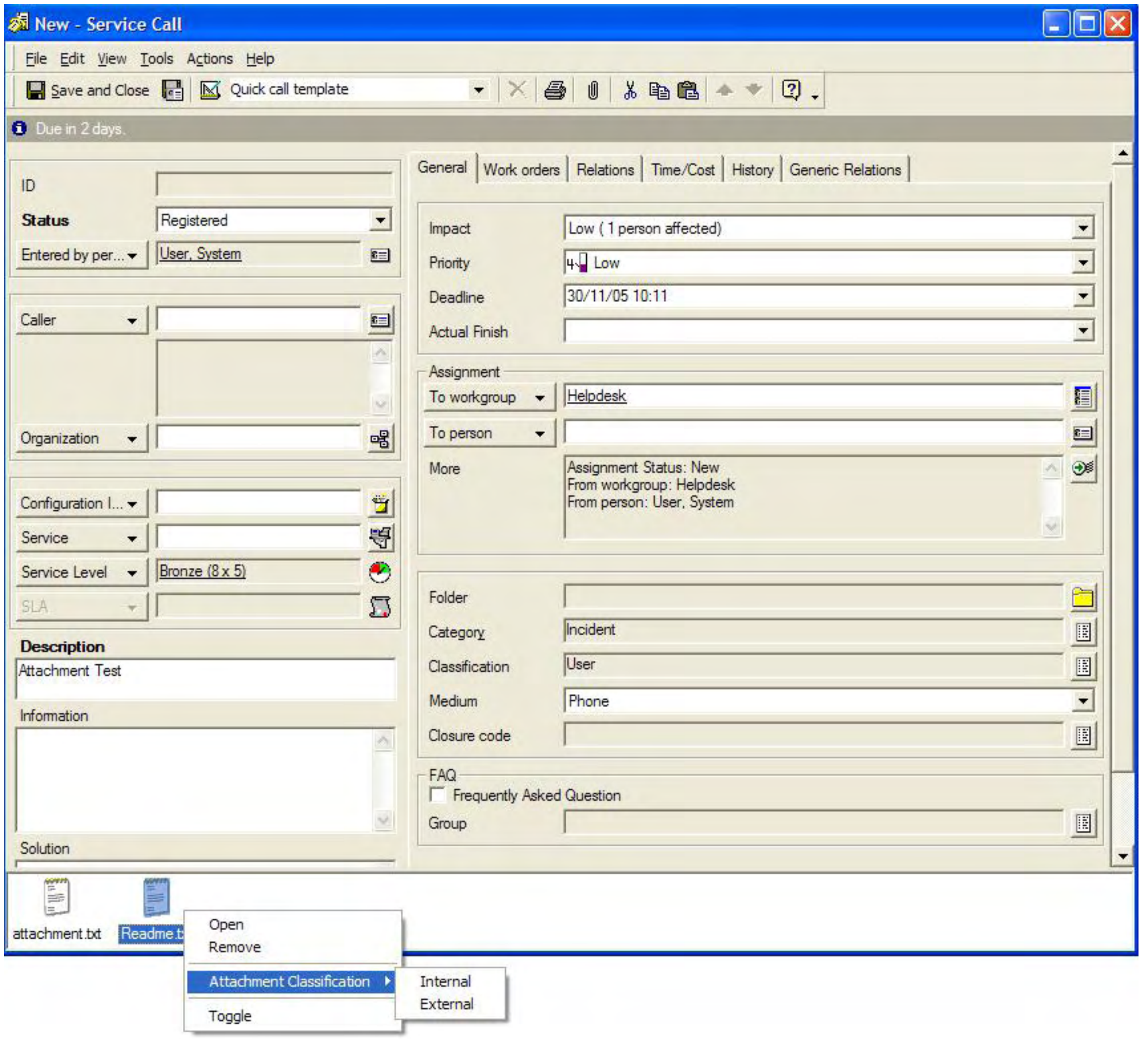
Figure 2



Set the Classification for Attachments

User can set attachment classification code by right clicking attachment icon, for example, in service call:

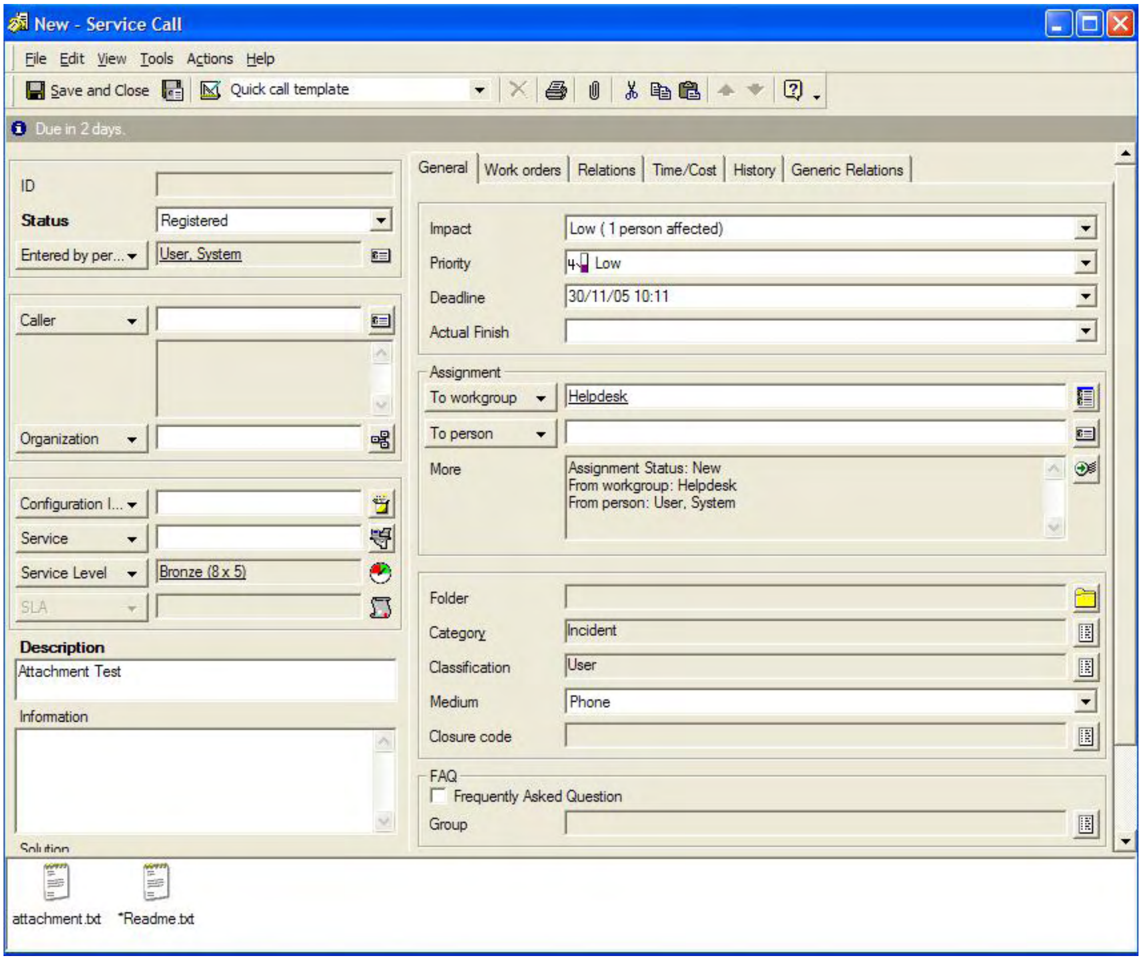
Figure 3



One attachment can be selected with multiple attachment classification codes.

The name of attachment which has been set classification is prefixed with an asterisk ('*'):

Figure 4



To get an overview of entire Attachment Classifications setting, user can click "Toggle" in Attachment Field:

Figure 5

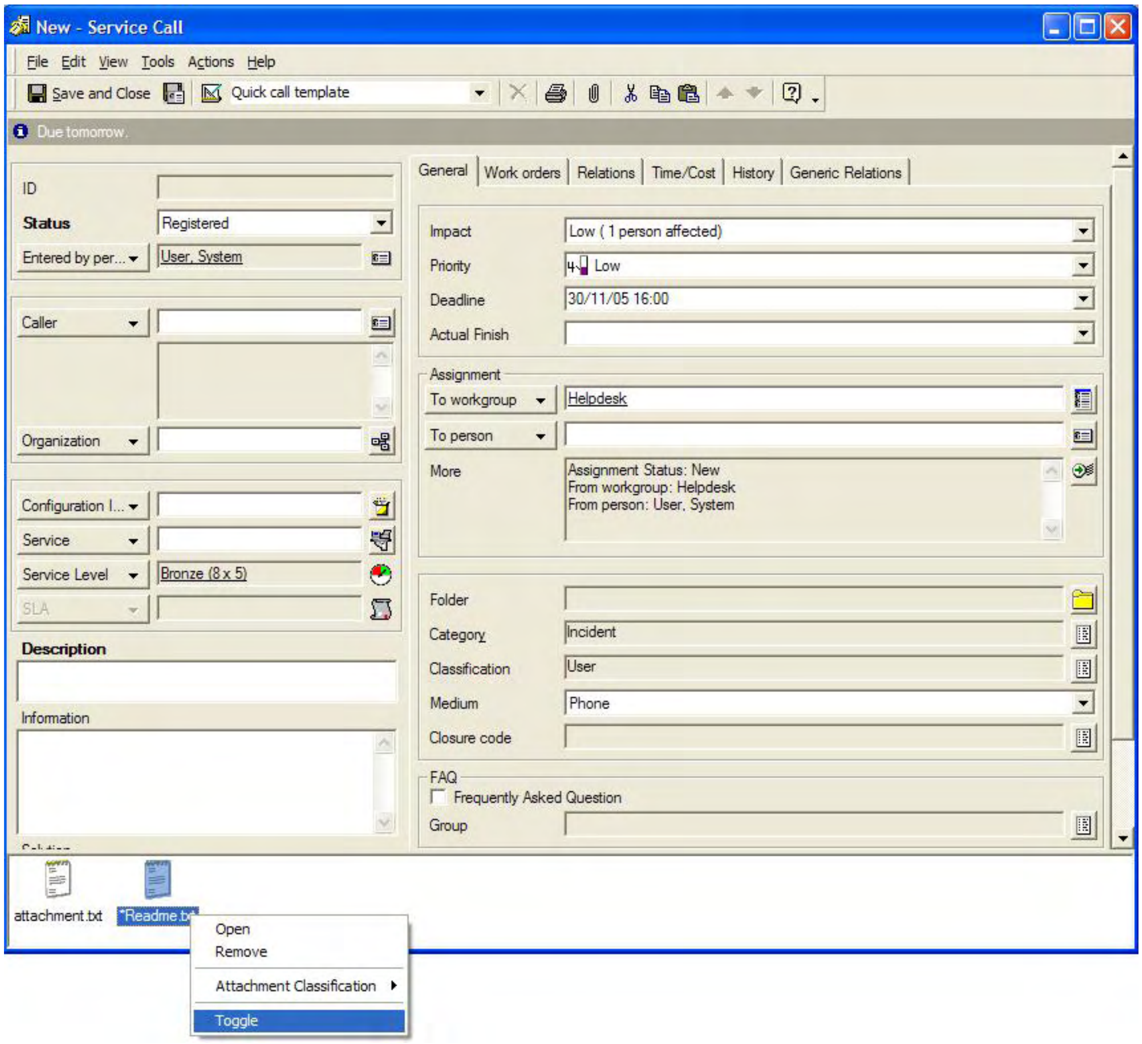
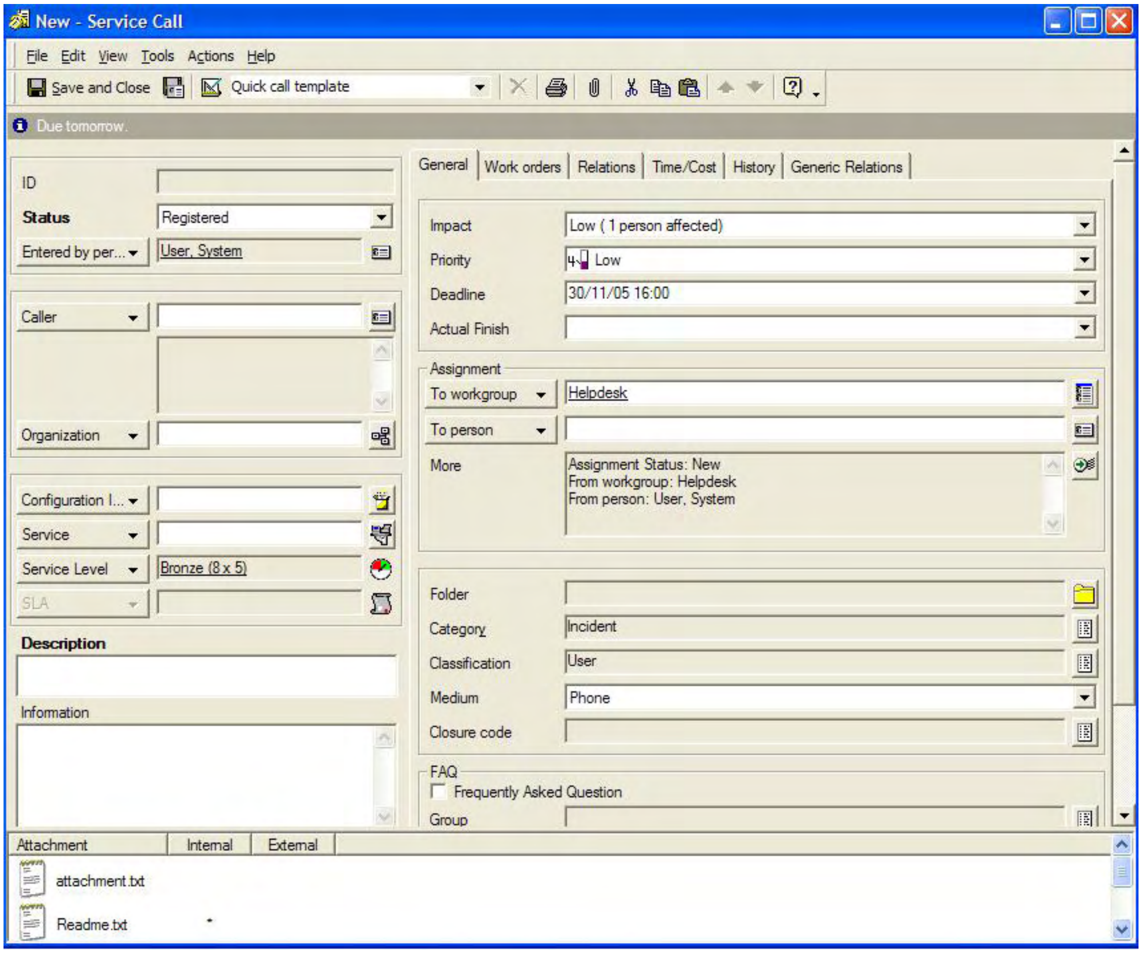


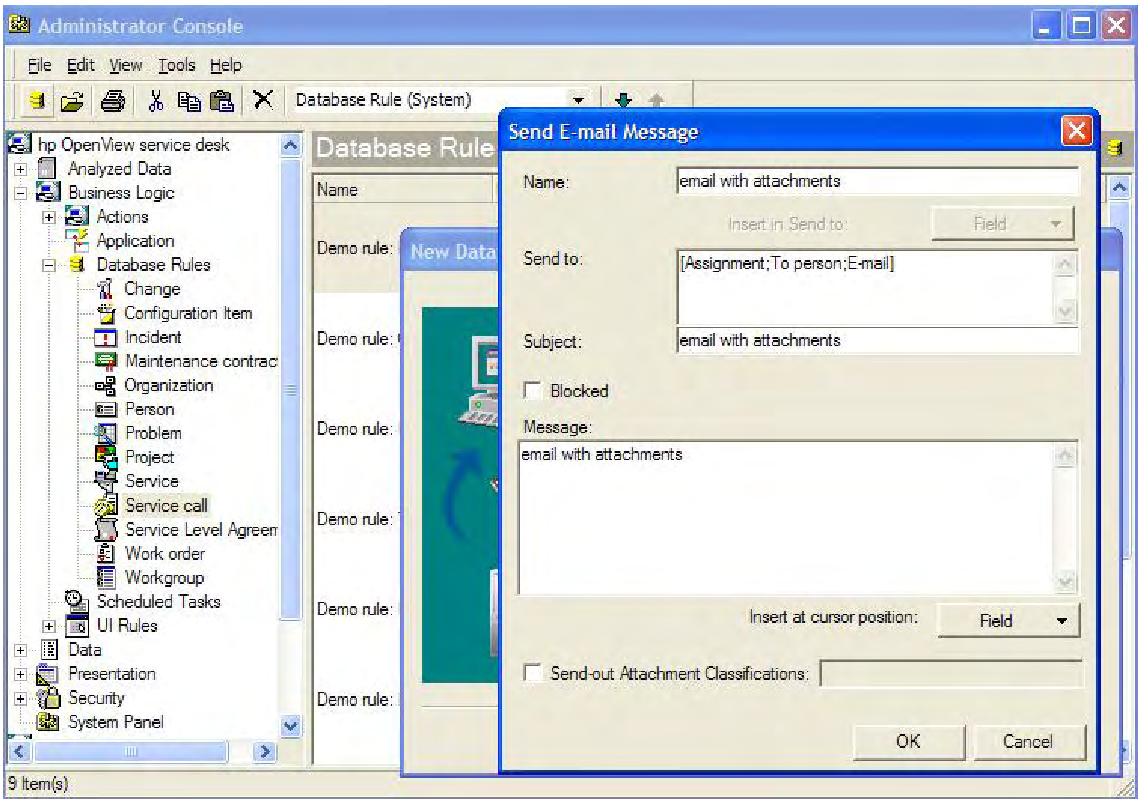
Figure 6


New - Service Call
File Edit View Tools Actions Help
Save and Close
Quick call template
Due tomorrow.
ID
Status
Registered
Entered by per...
User, System
Caller
Organization
Configuration I...
Service
Service Level
Bronze (8 x 5)
SLA
Description
Information
General
Work orders
Relations
Time/Cost
History
Generic Relations
Impact
Low (1 person affected)
Priority
Low
Deadline
30/11/05 16:00
Actual Finish
Assignment
To workgroup
Helpdesk
To person
More
Assignment Status: New
From workgroup: Helpdesk
From person: User, System
Folder
Category
Incident
Classification
User
Medium
Phone
Closure code
FAQ
Frequently Asked Question
Group
Attachment
Internal
External
attachment.txt
Readme.txt

Send Email with Attachment in Database Rule

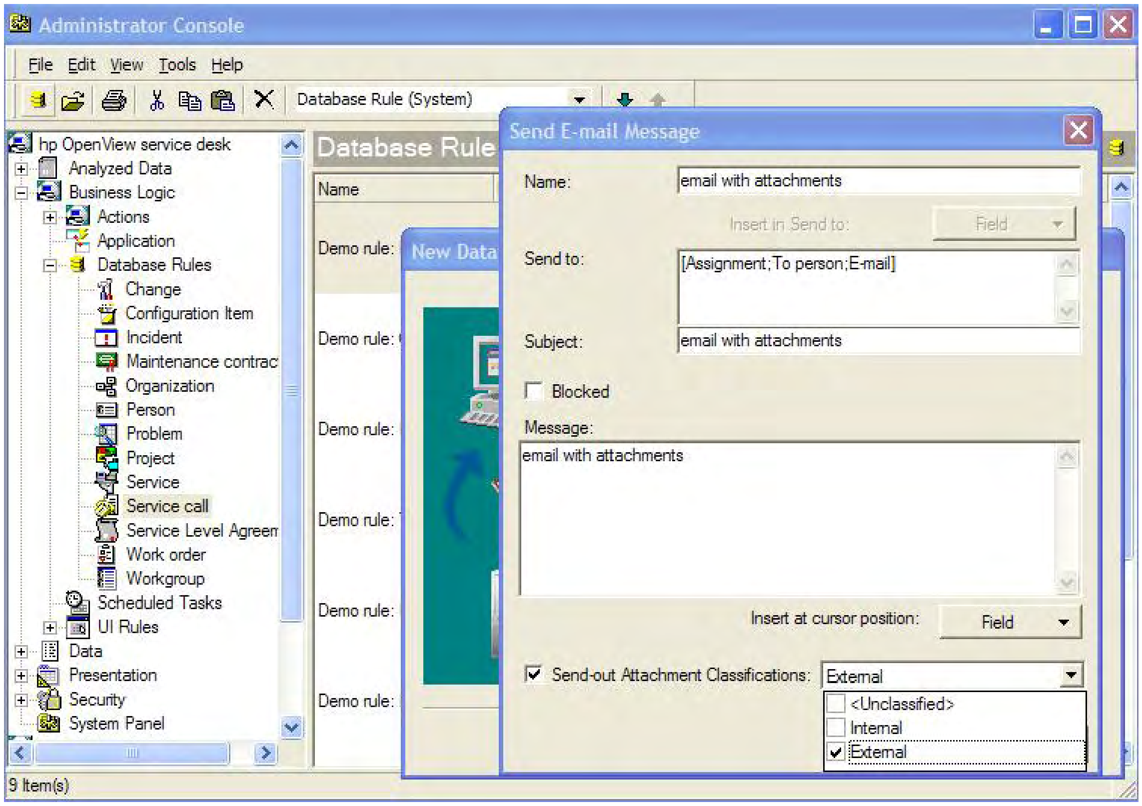
Logon with system administrator, open administrator console, go to **Business Logic** → **Database Rules** → **Service call**, create a new Service call Database Rule with an action of "Send e-mail message":

Figure 7



In "Send e-mail Message" form, a new checkbox named "Send-out Attachment Classifications" has been added. After checking the checkbox, choose classification code in the right drop down list:

Figure 8



There is a default attachment classification code "<Unclassified>", it is used for those attachments which have no attachment classification code.

When this Database Rule is triggered, it will send an outbound mail with attachments of selected classification codes.

Generic Association Mechanism

Introduction

Generic Relation is introduced to establish relation between two arbitrary items in Service Desk.

Generic Relation Group is to implement N to M relations between items.

A new UI Rule action "Limit field value range by relations" is added to limit available values of code field in a form by pre-defined Generic Relations.

Getting Started

Generic Relation and Relation Type

Generic Relation

A Generic Relation is a relation between two arbitrary items in Service Desk. For example, with Generic Relation you can establish relations between CI (Configuration Item) and Service Call Classification.

Generic Relation Type

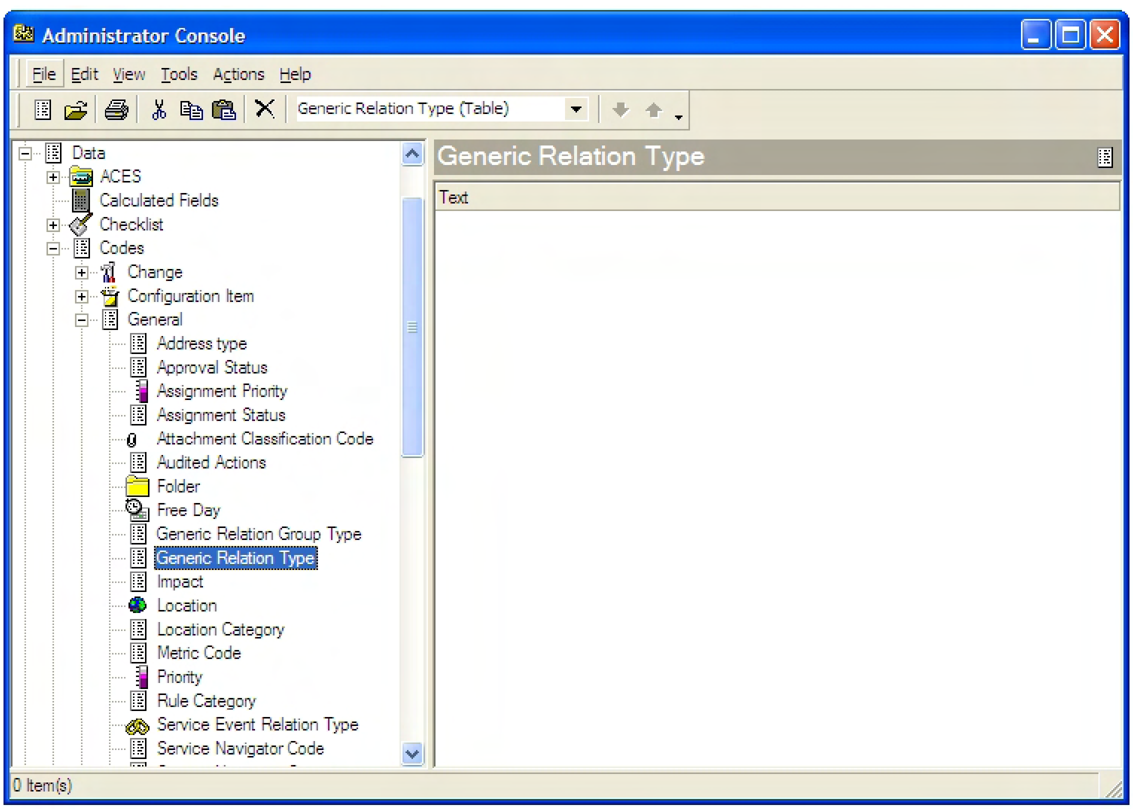
Generic Relation Type is to describe the nature of relations, which contains Name, From Item and To Item. For example, Name is "Incident to Form", From Item is "Incident" and To Item is "Form". Many relations share one Generic Relation Type.

In general, system administrator predefines Generic Relation Types and relations created by normal user is restricted to these Generic Relation Types.

Example

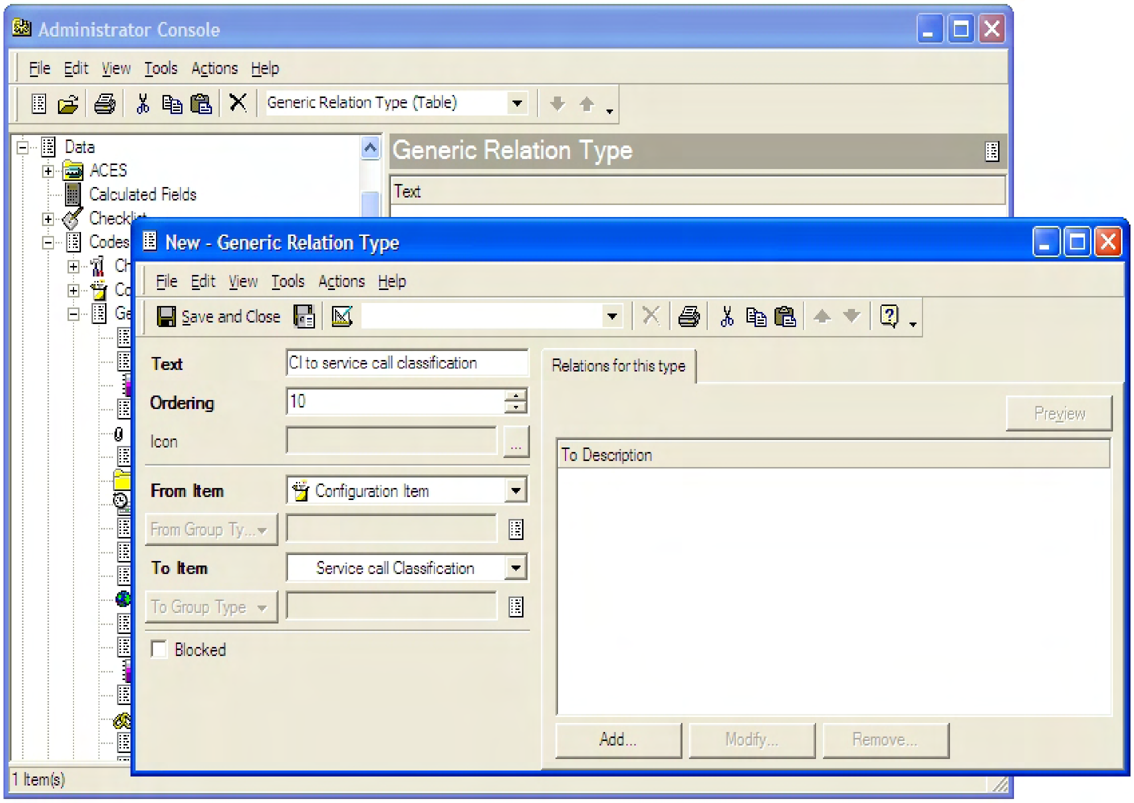
Logon with system administrator, open administrator console, go to **Data** → **Codes** → **General** → **Generic Relation Type**:

Figure 9



Create a Generic Relation Type:

Figure 10



Customize the CI form and Service Call Classification form to add field "Generic Incoming Relations" and "Generic Outgoing Relations":

Figure 11

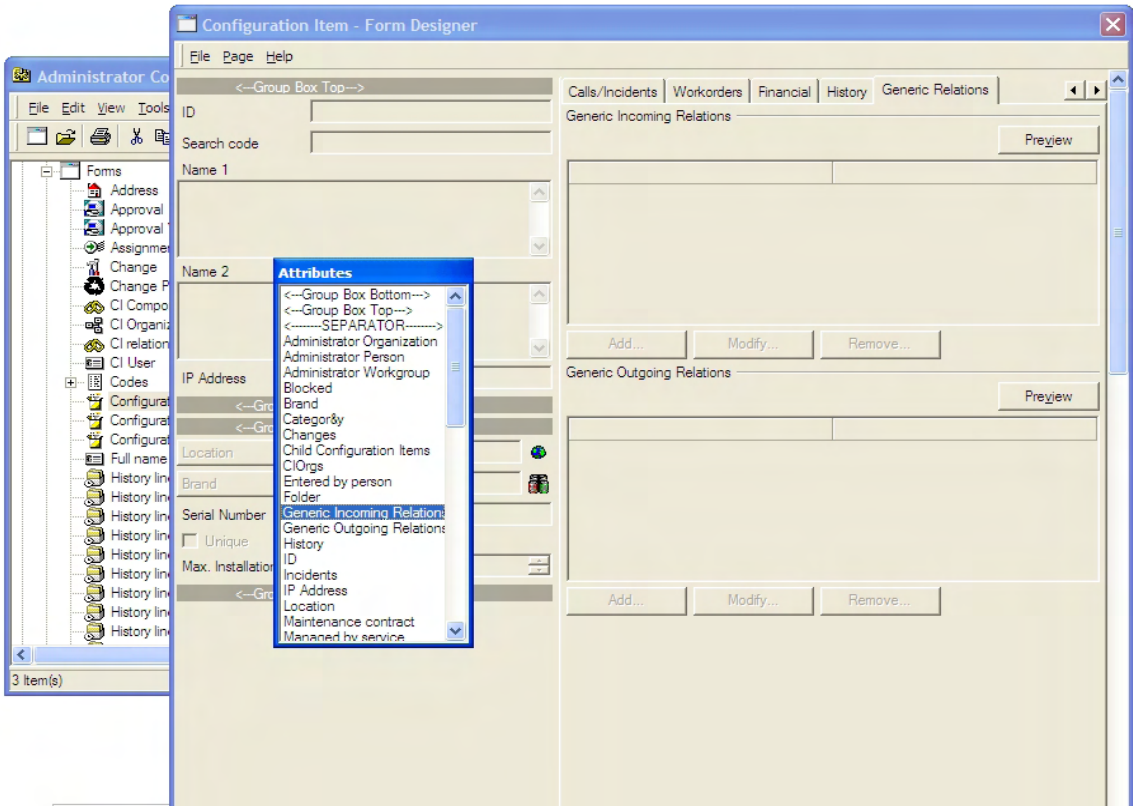
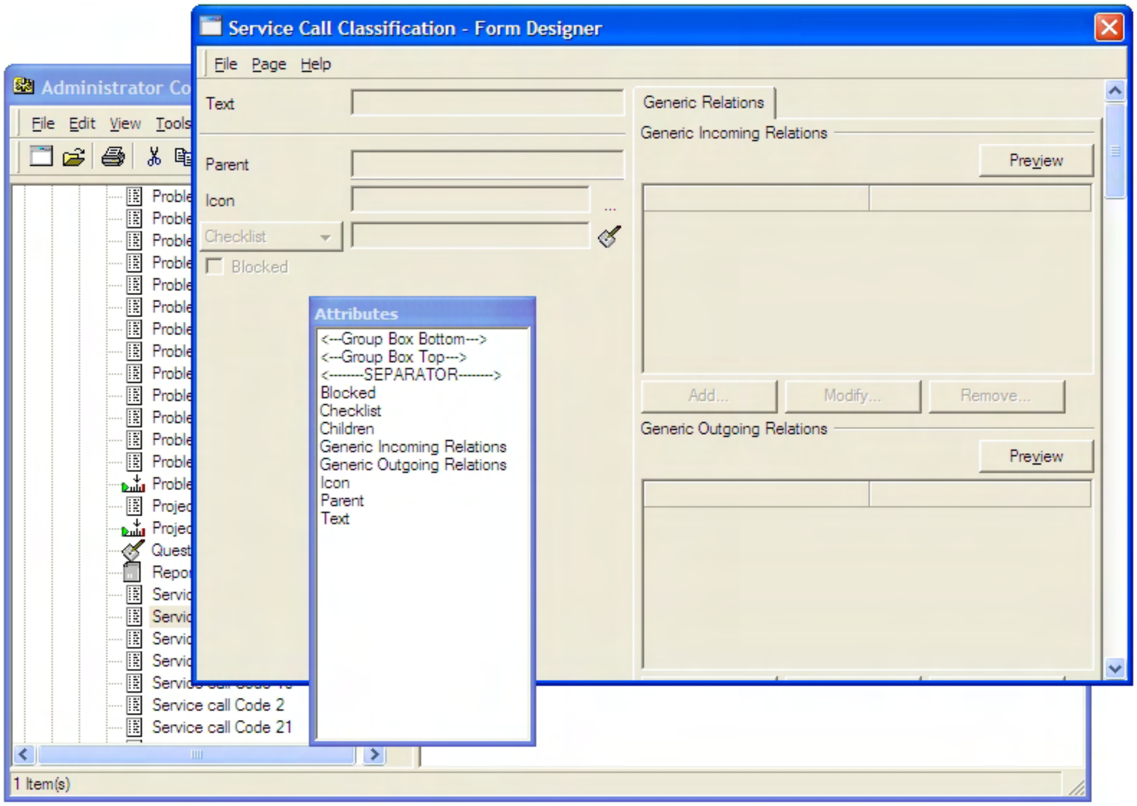


Figure 12


Service Call Classification - Form Designer
File Page Help
Text
Parent
Icon
Checklist
Blocked
Generic Relations
Generic Incoming Relations
Preview
Add...
Modify...
Remove...
Generic Outgoing Relations
Preview
Attributes
<---Group Box Bottom--->
<---Group Box Top--->
<--------SEPARATOR-------->
Blocked
Checklist
Children
Generic Incoming Relations
Generic Outgoing Relations
Icon
Parent
Text
Administrator Co
File Edit View Tools
Service call Code 2
Service call Code 21
1 Item(s)

Normal service desk user with write access to "Generic Incoming Relations" and "Generic Outgoing Relations" of CI can modify Generic Relations on CI form:

Figure 13

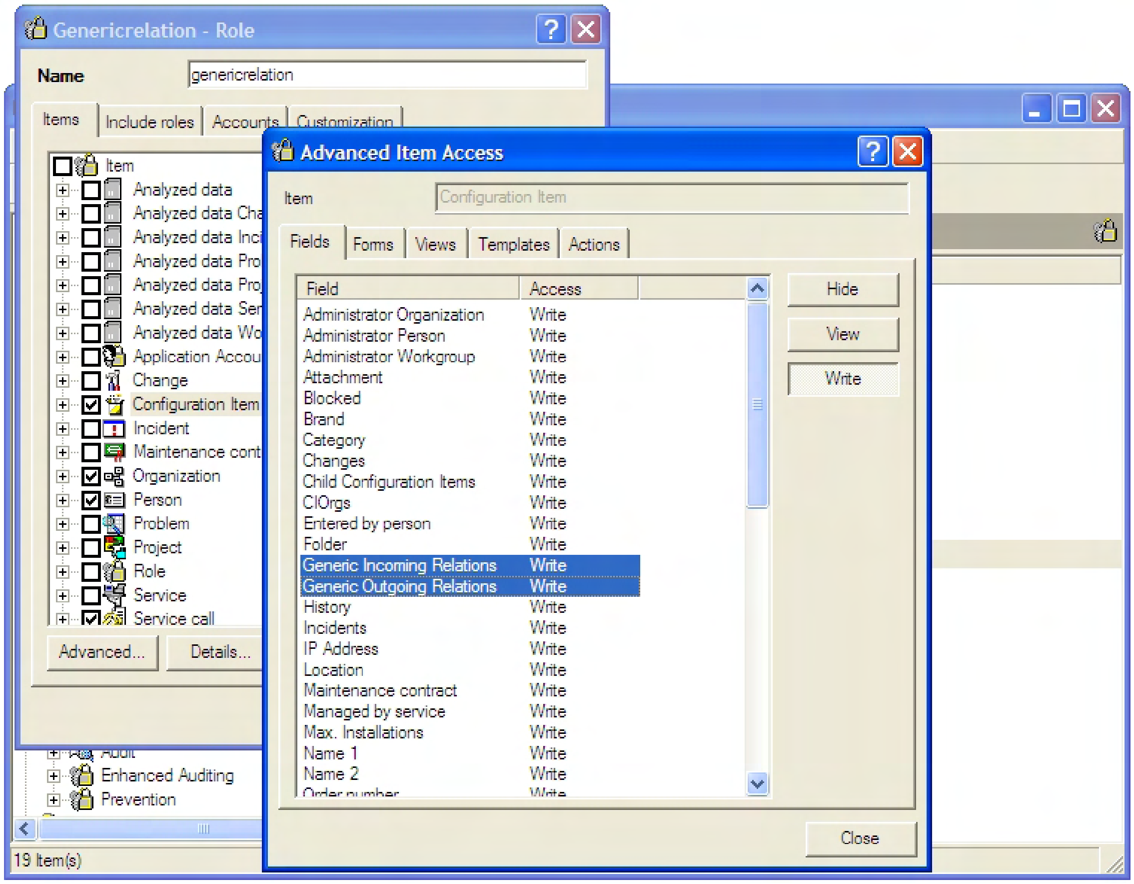
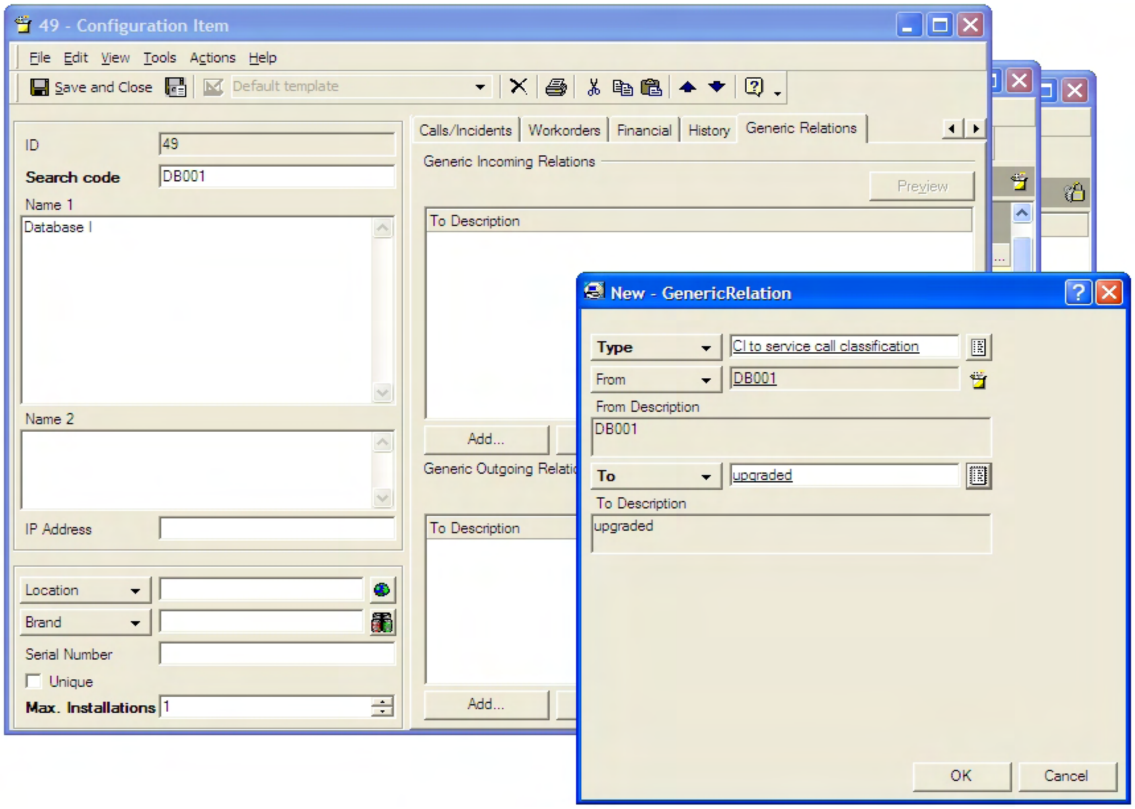
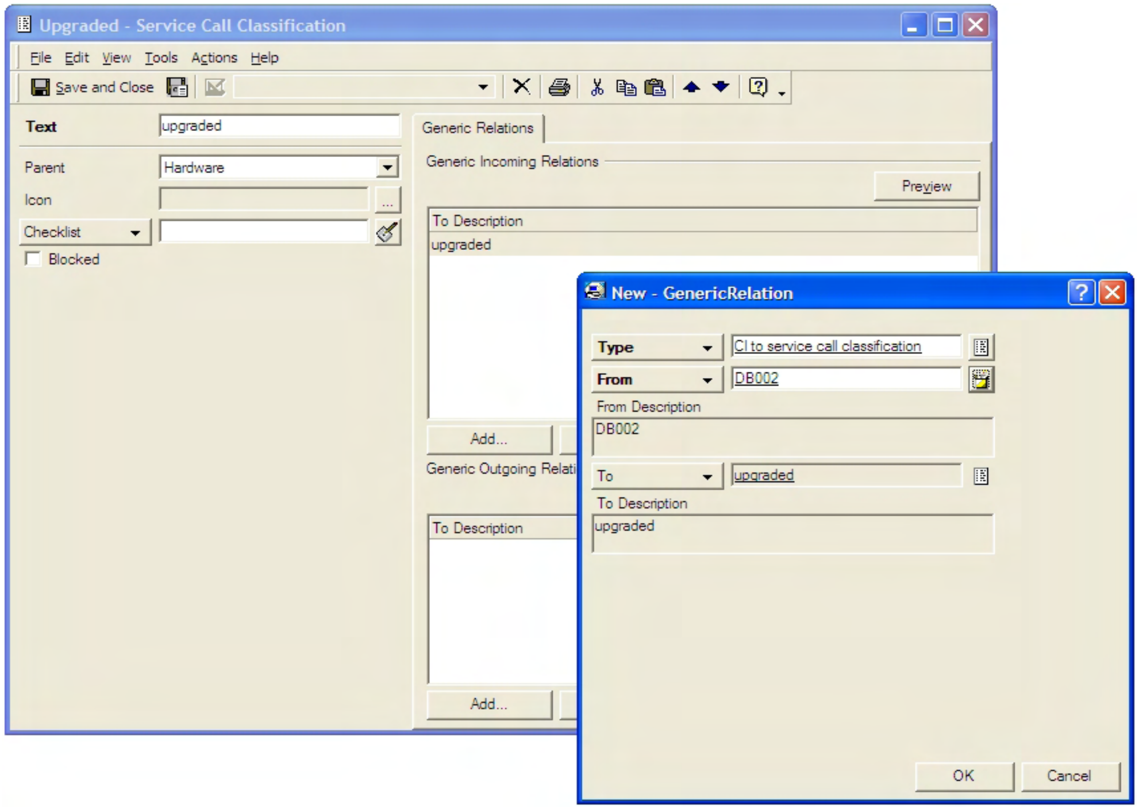


Figure 14


49 - Configuration Item
File Edit View Tools Actions Help
Save and Close
Default template
ID
49
Search code
DB001
Name 1
Database I
Name 2
IP Address
Location
Brand
Serial Number
Unique
Max. Installations
1
Calls/Incidents
Workorders
Financial
History
Generic Relations
Generic Incoming Relations
Preview
To Description
Add...
Generic Outgoing Relatio
To Description
Add...
New - GenericRelation
Type
CI to service call classification
From
DB001
From Description
DB001
To
upgraded
To Description
upgraded
OK
Cancel

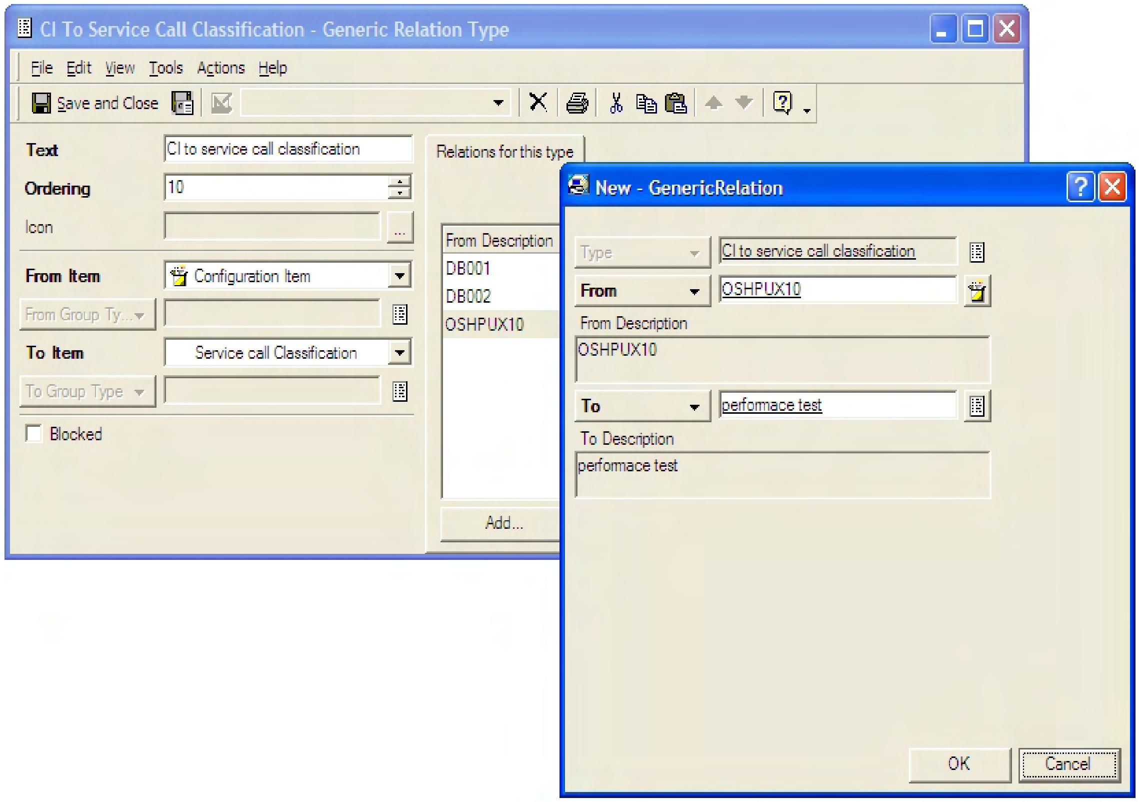
System administrator can modify Generic Relations on Service Call Classification form:

Figure 15



System administrator can also directly view and modify all relations of this type in the Generic Relation Type form:

Figure 16



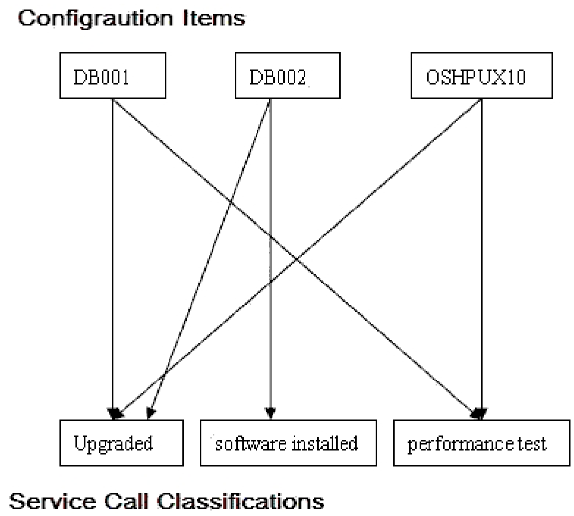
Generic Relation Group and Group Type

Generic Relation Group

Generic Relations are very practical for one to one relations between Service Desk items. For N to M relations with repeating groups, they are less practical because each relation has to be explicitly defined. To reduce the number of relations required, Generic Relation Group is introduced.

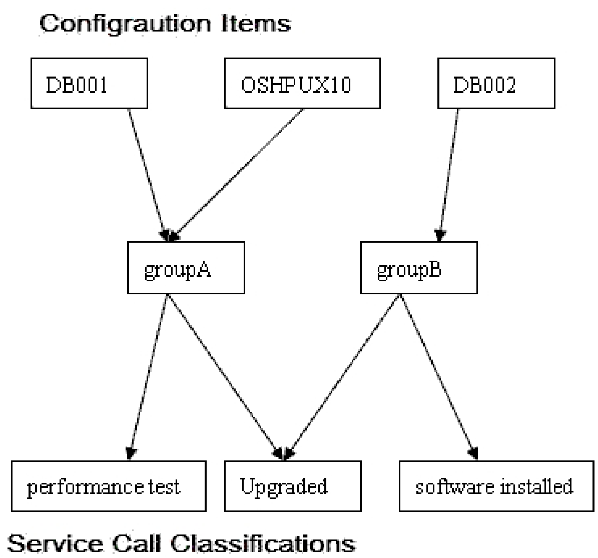
For example, there are three computers need to be upgraded or installed software or have a performance test. With generic relations, each relation should be defined explicitly:

Figure 17



With Generic Relation Groups, repeated N to M relations can be expressed with single relations.

Figure 18



Generic Relation Group Type

Generic Relation Group Type is to classify usage of different Generic Relation Groups.

For example, system administrator can define two types of group to be used between CI and Service Call Classification, "Hardware Maintenance" and "Hardware Application".

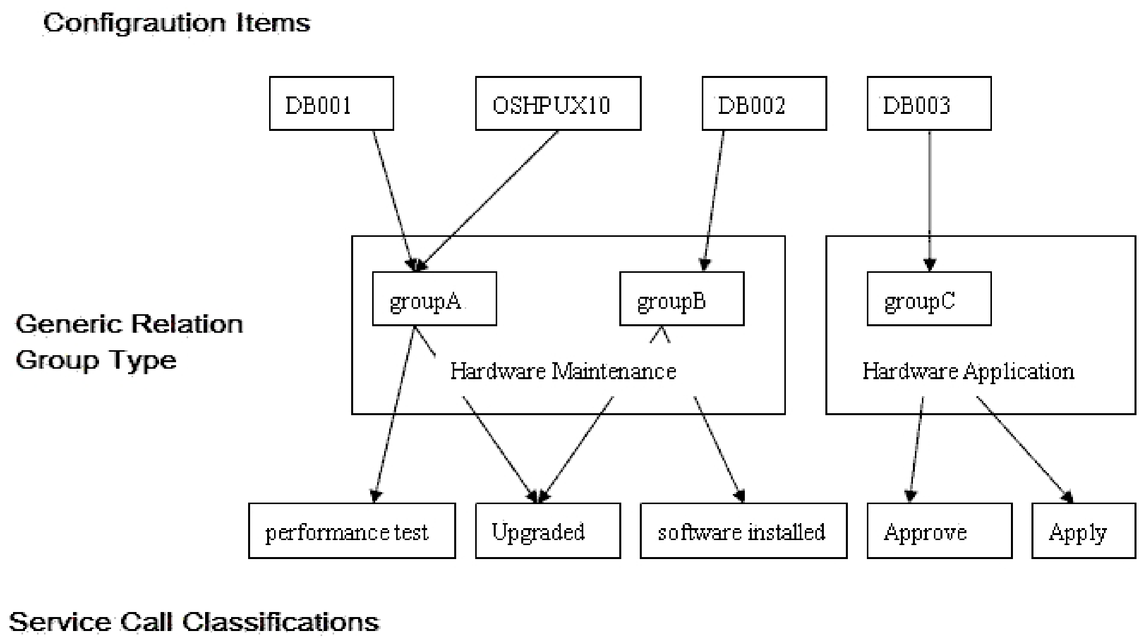
With these two Generic Relation Group Types, system administrator can define the following Generic Relation Types:

- "CI to Maintenance group"
- "CI to Application group"
- "Maintenance group to Service Call Classification"

- "Application group to Service Call Classification"

When adding Outgoing relations from CI, user can first choose the Relation Type "CI to Maintenance group" or "CI to Application group" then choose the group correspondingly.

Figure 19



Also, Generic Relation Group Type can be used to limit creation of relations.

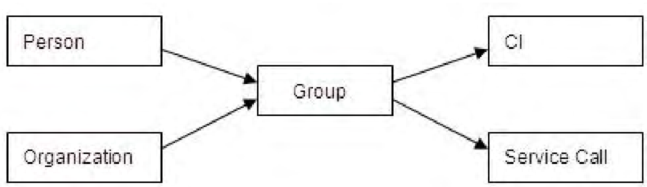
Imagine a case where Persons can only be related to CIs, and Organizations can only be related to Service Calls. With Generic Relation Groups, the following relation types are created:

- "Person to Group"
- "Group to CI"
- "Organization to Group"
- "Group to Service Call"

Now it is possible to relate Persons to Service Calls, and organizations to CIs.

- "Person to Group" -> "Group to Service Call"
- "Organization to Group" -> "Group to CI"

Figure 20



With Generic Relation Group Type, we can define two group types.

- "Group A" (used between Person and CI)

- "Group B" (used between Organization and Service Call)

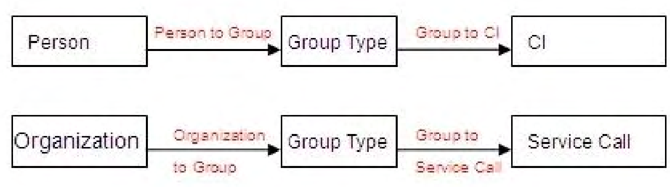
And we define the following relation types

- "Person to Group A"
- "Group A to CI"
- "Organization to Group B"
- "Group B to Service Call"

As a result, the possible chains are limited to:

- "Person to Group" -> "Group to CI"
- "Organization to Group" -> "Group to SC"

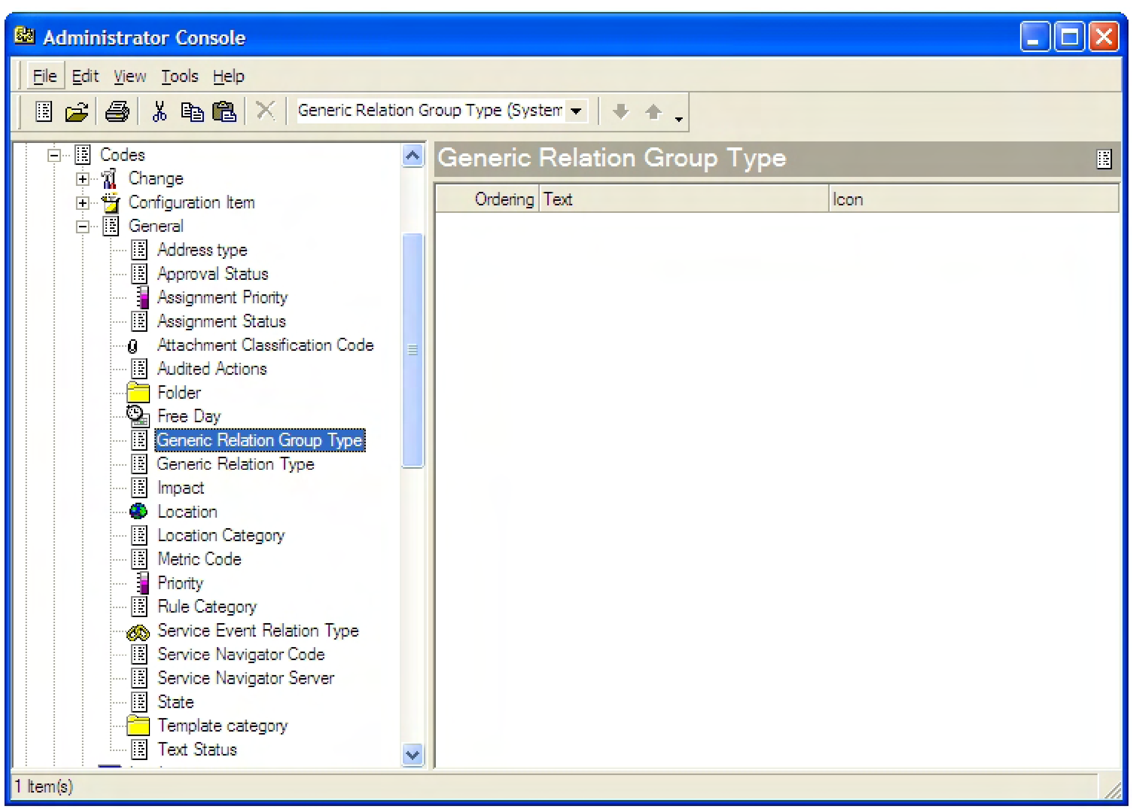
Figure 21



Example

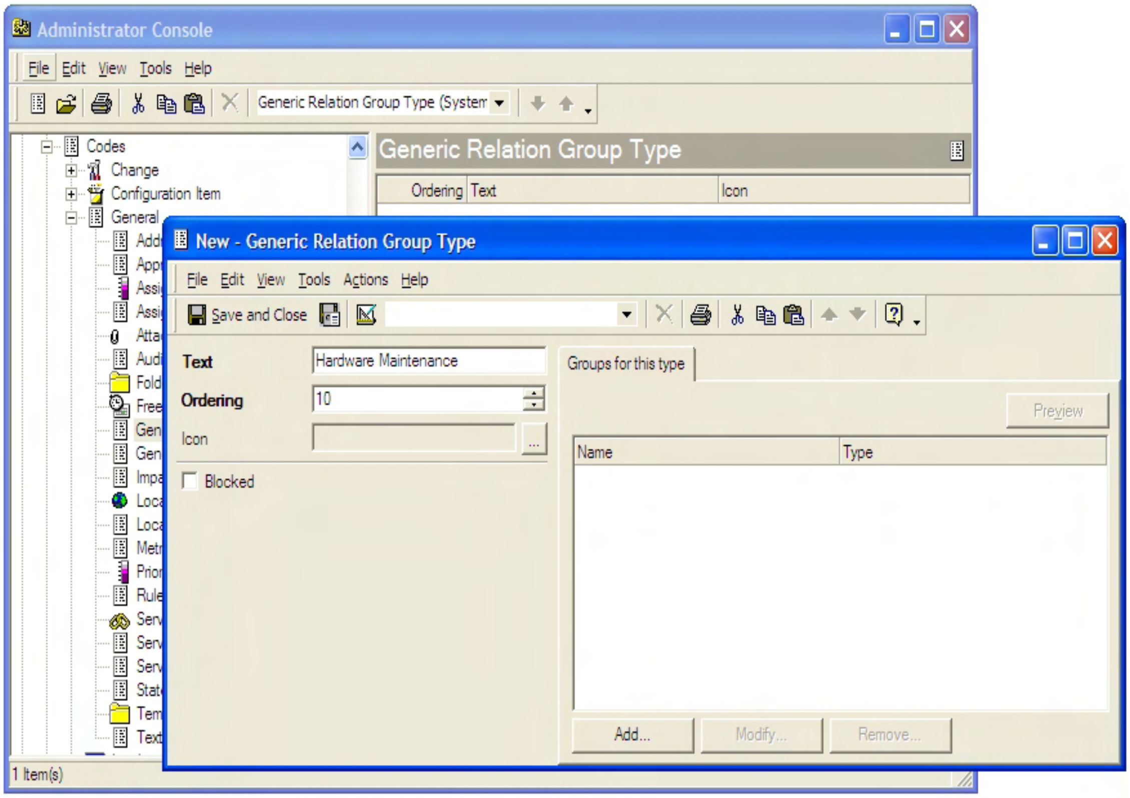
Logon with system administrator, open administrator console, go to **Data** → **Codes** → **General** → **Generic Relation Group Type**:

Figure 22



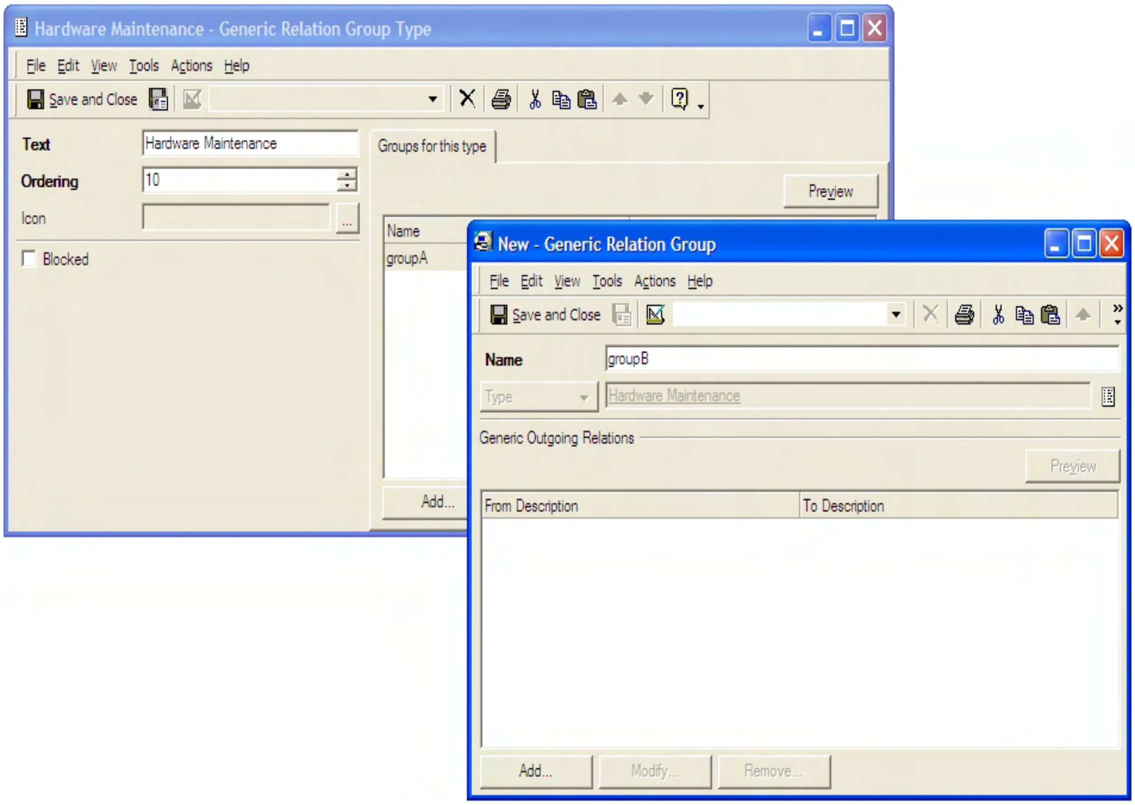
Create a Generic Relation Group Type:

Figure 23



Add groups for this group type:

Figure 24



Define Relation Types using this Group Type:

Figure 25

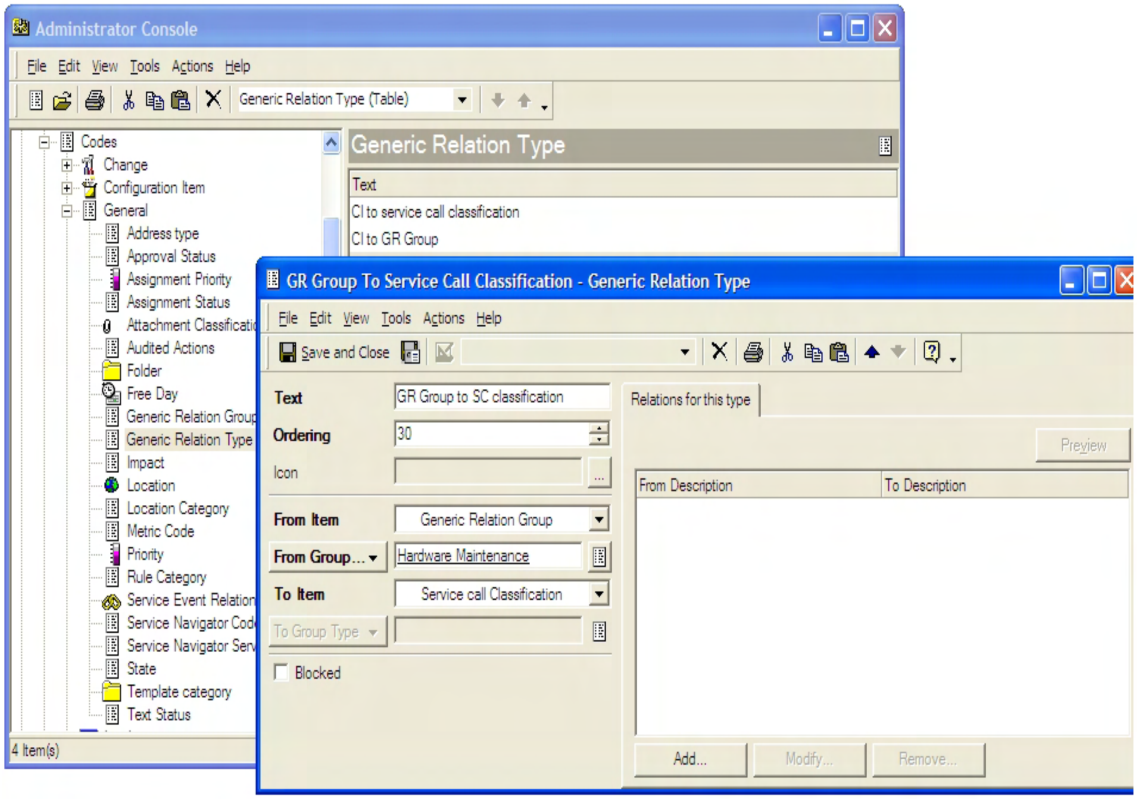
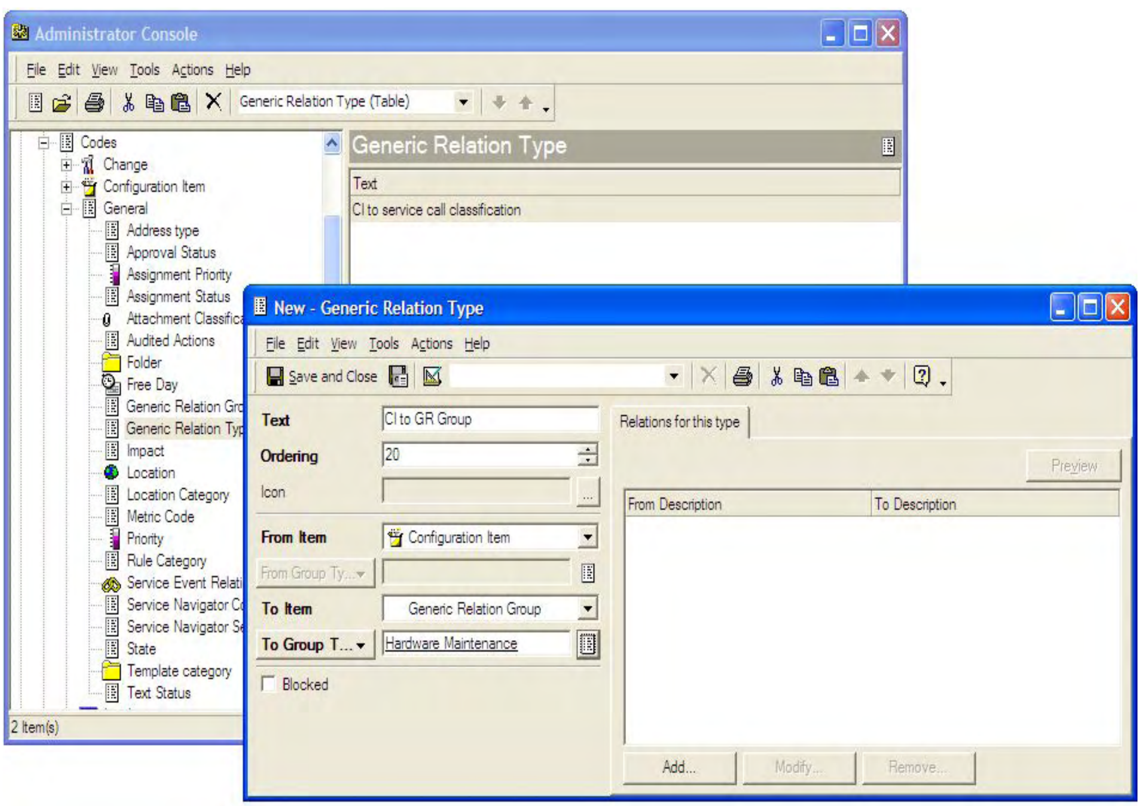


Figure 26



Customize the CI form and Service Call Classification form to add field "Generic Incoming Relations" and "Generic Outgoing Relations":

Figure 27

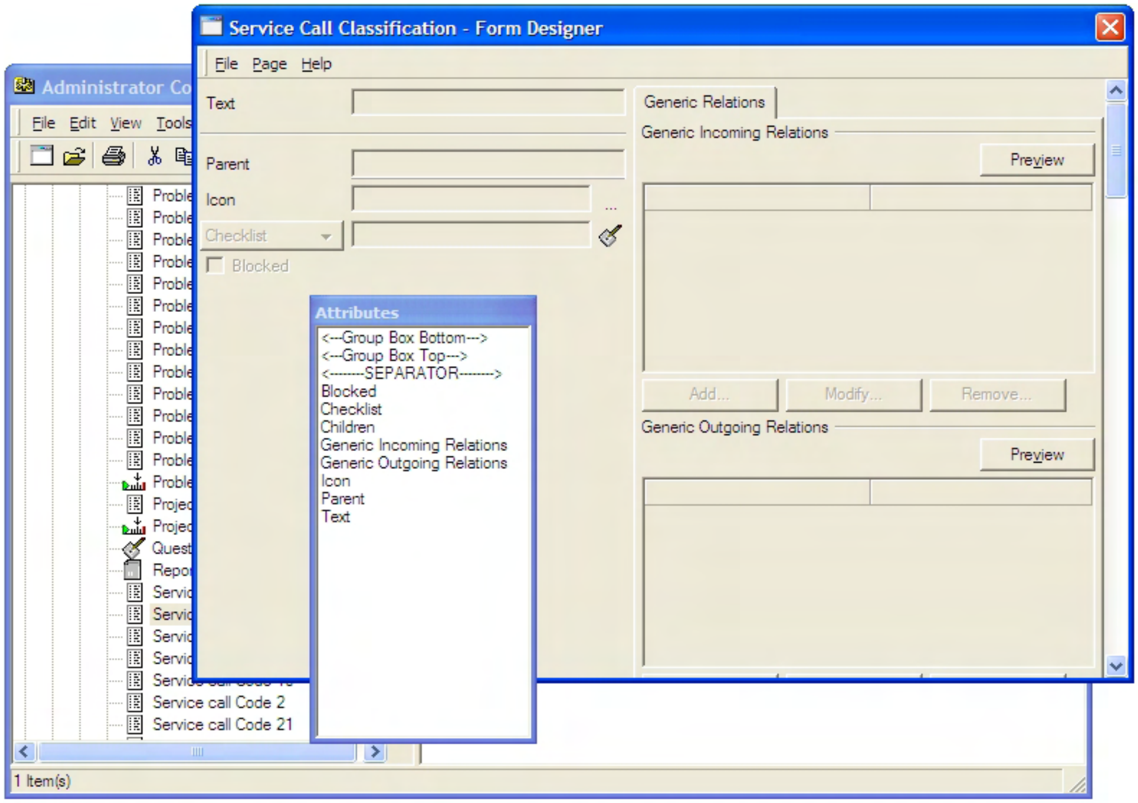
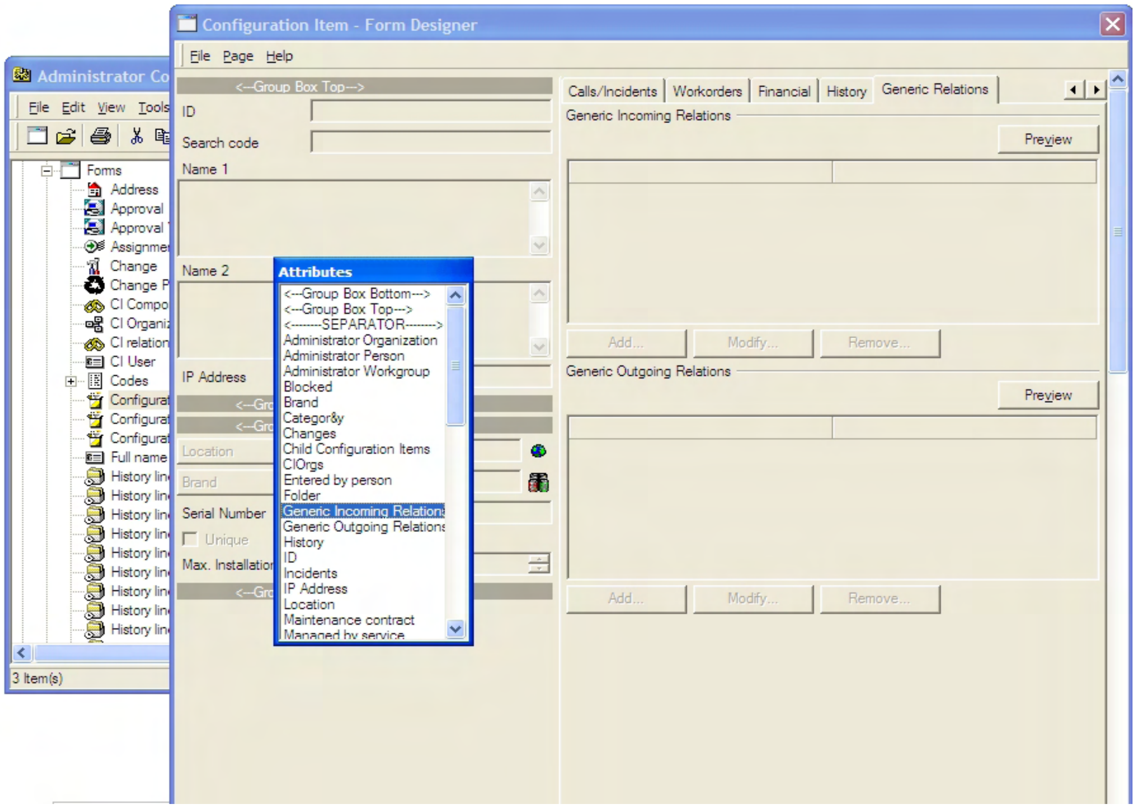
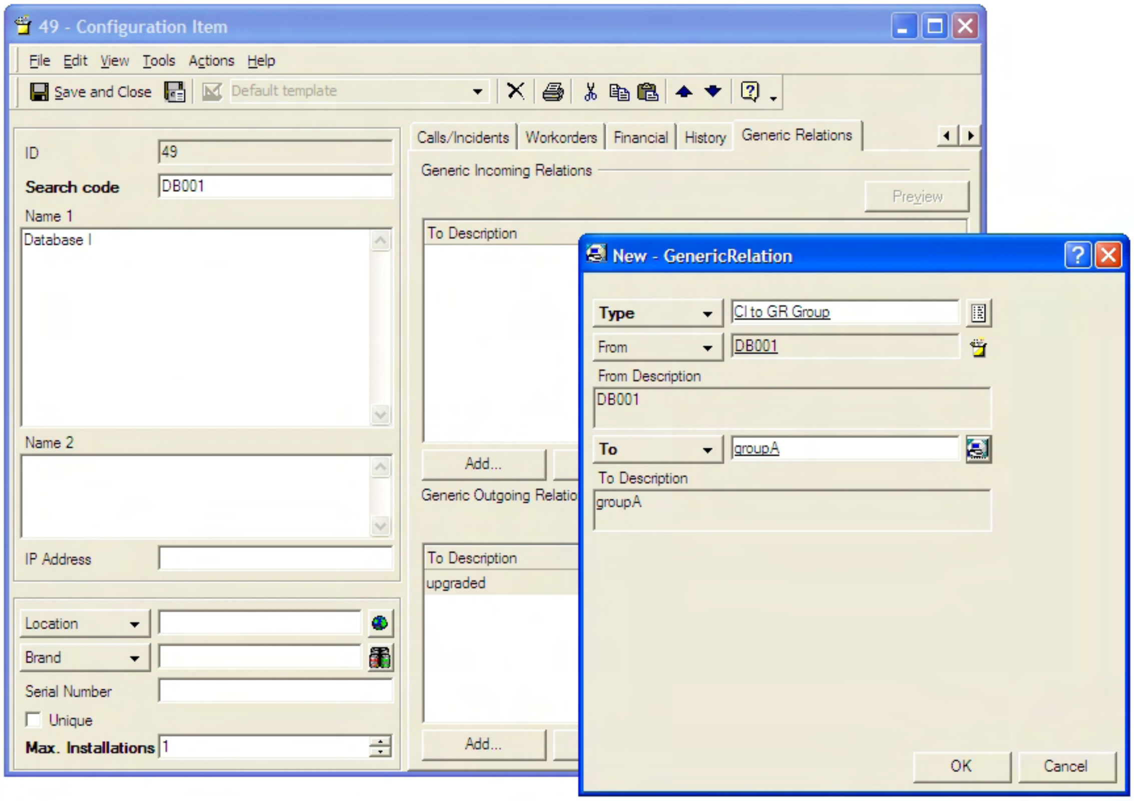


Figure 28


Configuration Item - Form Designer
File Page Help
<--Group Box Top-->
ID
Search code
Name 1
Name 2
IP Address
Location
Brand
Serial Number
Unique
Max. Installation
Attributes
<--Group Box Bottom-->
<--Group Box Top-->
<--------SEPARATOR-------->
Administrator Organization
Administrator Person
Administrator Workgroup
Blocked
Brand
Category
Changes
Child Configuration Items
CIOrgs
Entered by person
Folder
Generic Incoming Relations
Generic Outgoing Relations
History
ID
Incidents
IP Address
Location
Maintenance contract
Managed by service
Calls/Incidents
Workorders
Financial
History
Generic Relations
Generic Incoming Relations
Preview
Add...
Modify...
Remove...
Generic Outgoing Relations
Preview
Add...
Modify...
Remove...
Administrator Co
File Edit View Tools
Forms
Address
Approval
Approval
Assignme
Change
Change P
CI Compo
CI Organi
CI relation
CI User
Codes
Configurat
Configurat
Configurat
Full name
History lin
History lin
History lin
History lin
History lin
History lin
History lin
History lin
History lin
3 Item(s)

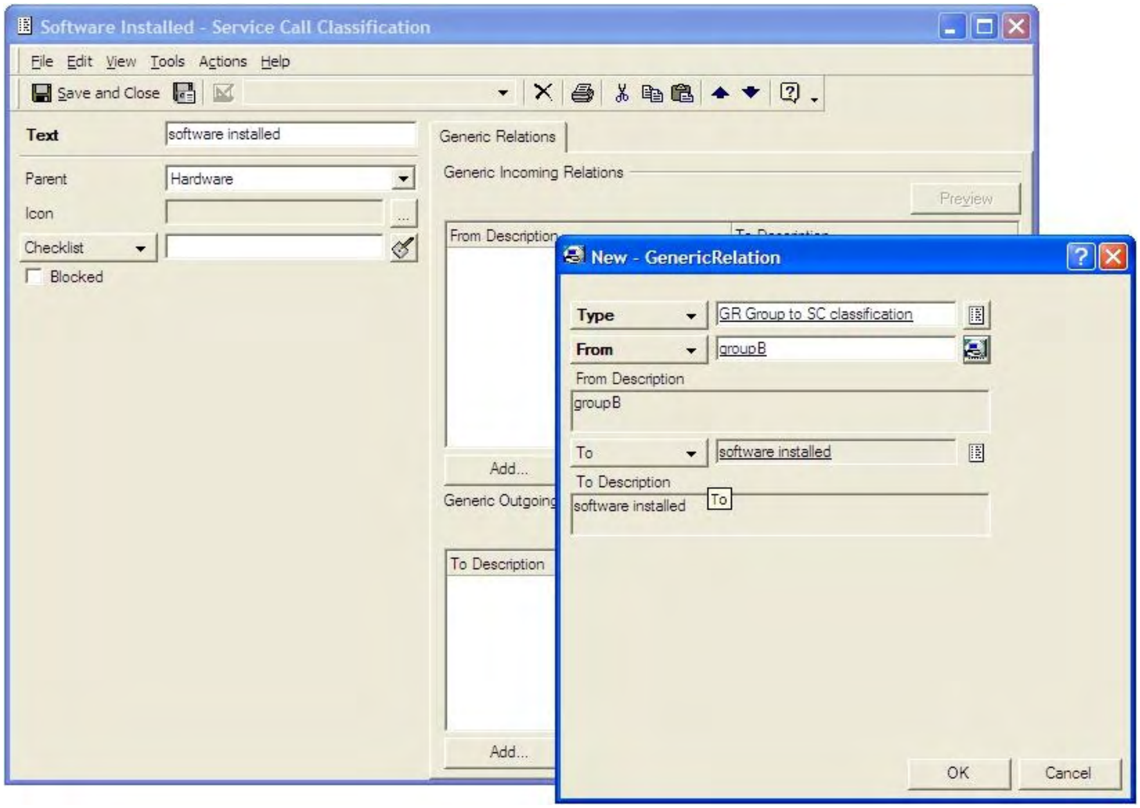
Normal Service Desk user with write access to "Generic Incoming Relations" and "Generic Outgoing Relations" of CI can modify Generic Relations on CI form:

Figure 29



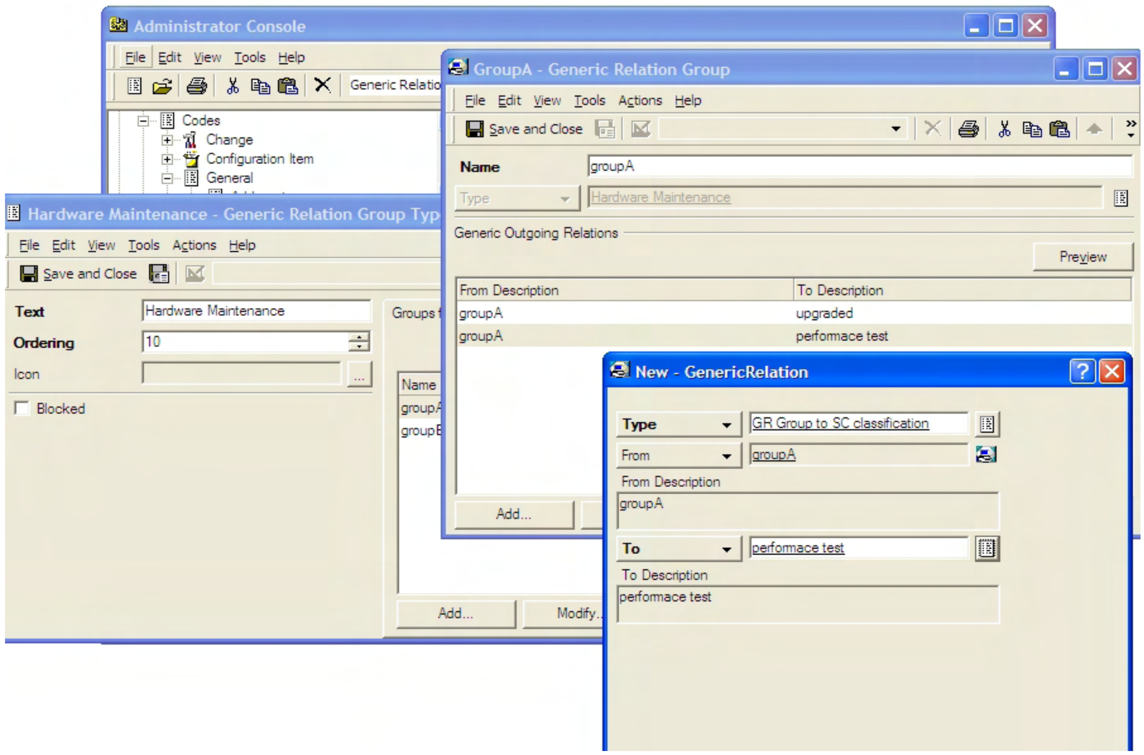
System administrator can modify Generic Relations on Service Call Classification form:

Figure 30



System administrator can also directly view and modify all relations based on this group type in the Generic Relation Group Type form:

Figure 31



New UI Rule Action: Limit Field Value Range by Relations

UI Rule Action: Limit Field Value Range by Relations

A new UI Rule action: "Limit field value range by relations" is added to Service Desk based on the Generic Relation feature.

For example, in a Service Call, the available selection of Service Call Classification can be restricted by selected CI value base on the Generic Relations from CI to Service Call Classification.

There are two typical scenarios:

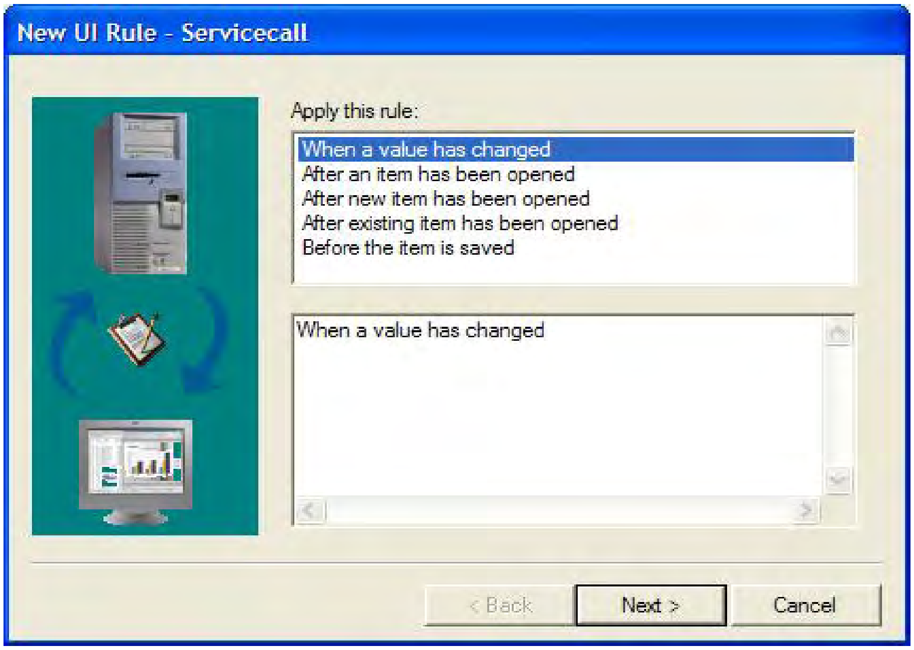
- Use direct Generic Relations between Source Field and Target Field.
- Use Generic Relations incorporate with Generic Relation Groups.

Example: Limit Field Value Range by Generic Relations

This example is based on the *Example of Generic Relation*. I.e. Generic Relation Type from CI to Service Call Classification has been defined and several relations has been added based on this type.

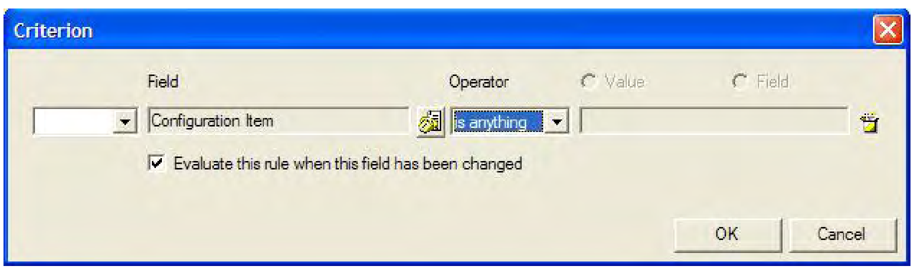
Create a Service Call UI Rule when a value as changed:

Figure 32



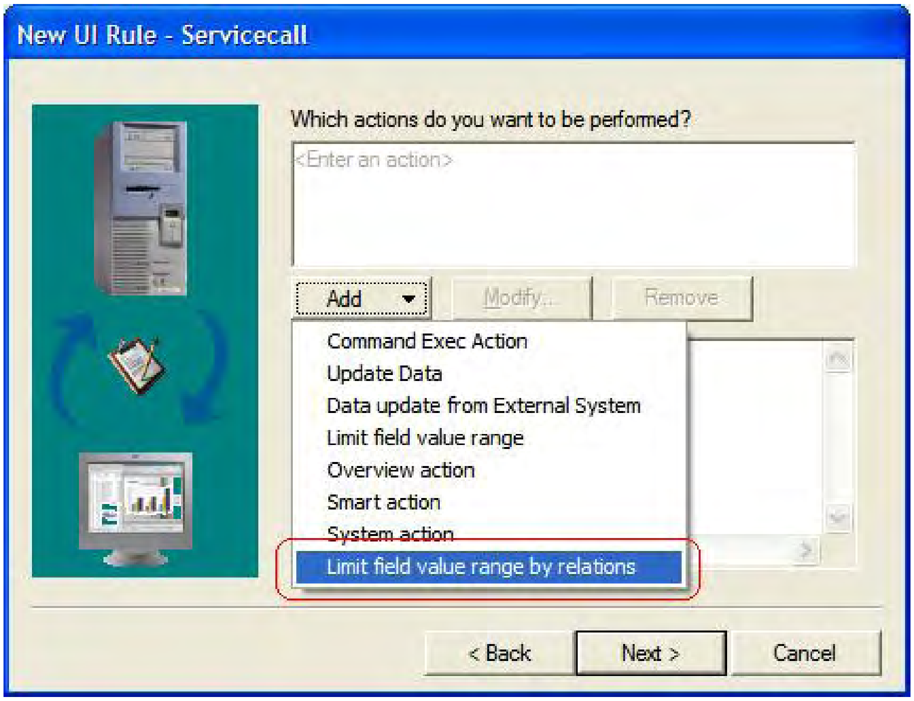
Add trigger Criterion CI is changed:

Figure 33



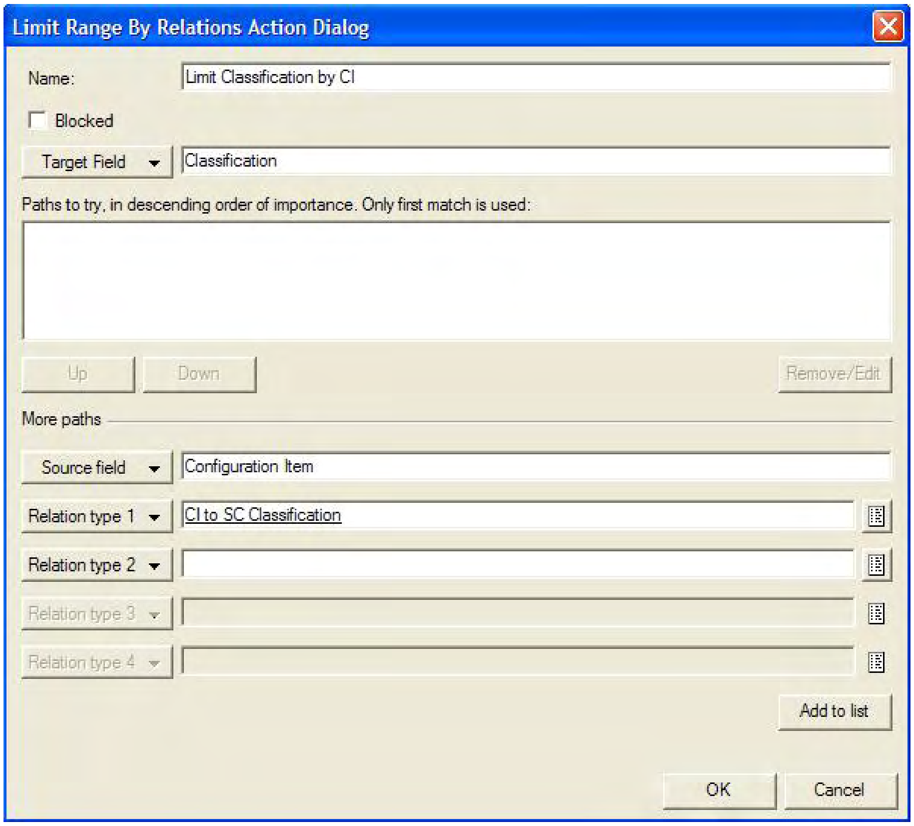
Add action "Limit field value range by relations":

Figure 34



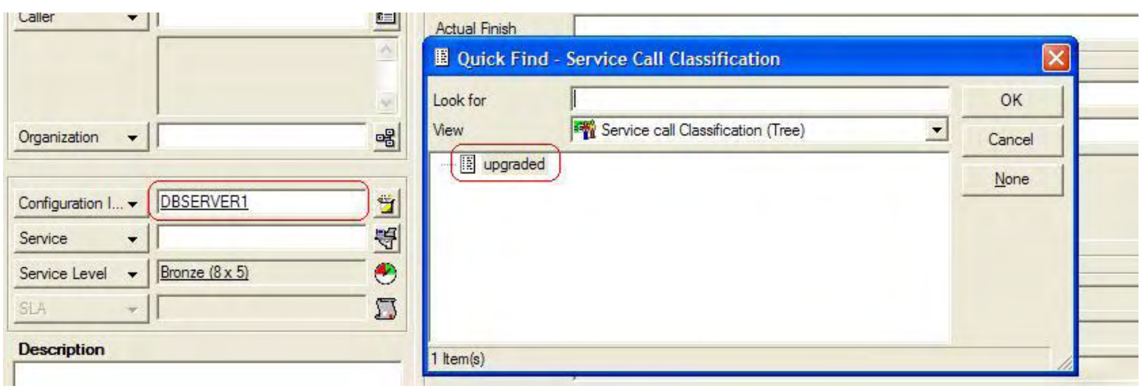
Limit target field "Classification" by source field "Configuration Item" base on Generic Relation Type "CI to SC Classification":

Figure 35



In a service call form, modify CI field and UI rule will be triggered. A quick find dialog will popup with restricted Classification Code values for selection:

Figure 36



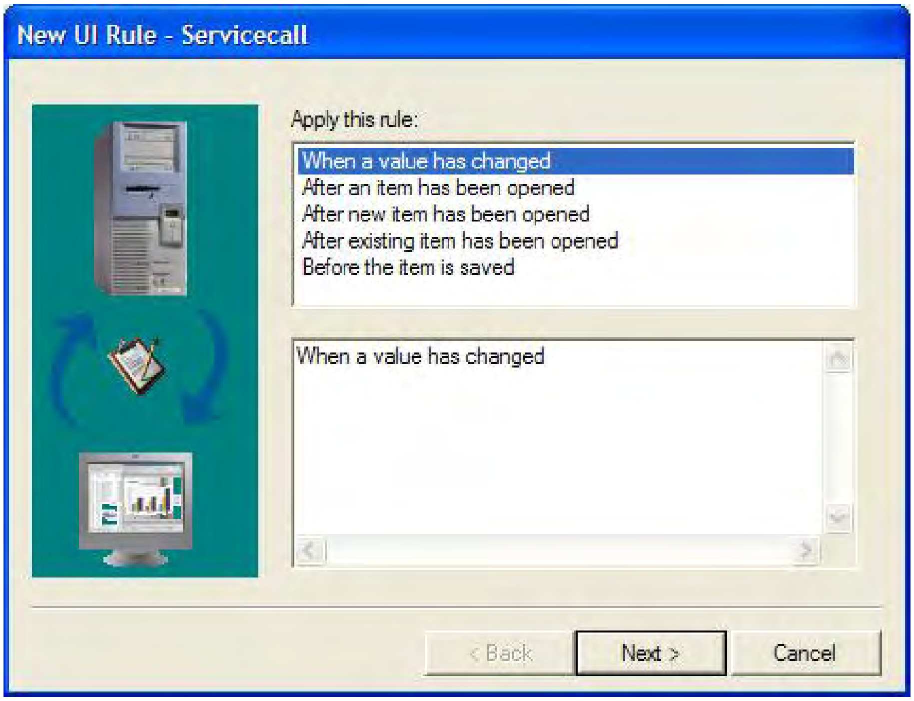
The value list will also be limited if user click "Quick Find" button next to Classification field. Value list of Classification will be refreshed when CI in service call is changed.

Example: Limit Field Value Range by Generic Relations based on Generic Relation Groups

This example is based on the *Example of Generic Relation Group*. I.e. Generic Relation Group Type and Generic Relation Types between CI and Service Call Classification have been defined and several relations have been added.

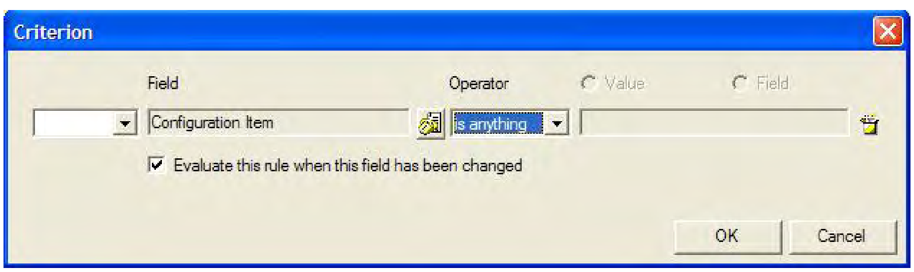
Create a Service Call UI Rule when a value as changed:

Figure 37



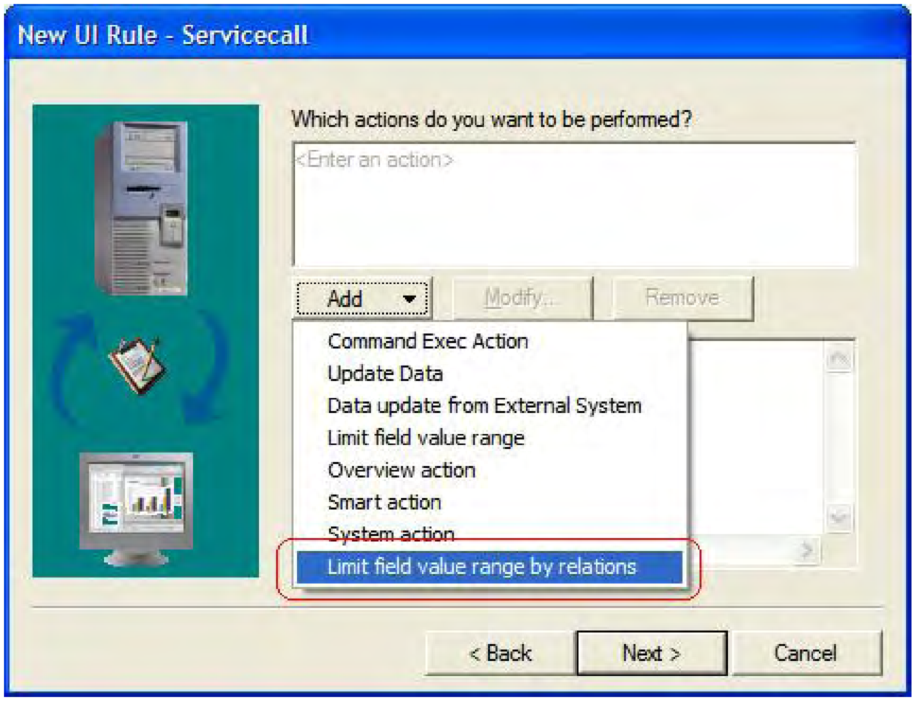
Add trigger Criterion CI is changed:

Figure 38



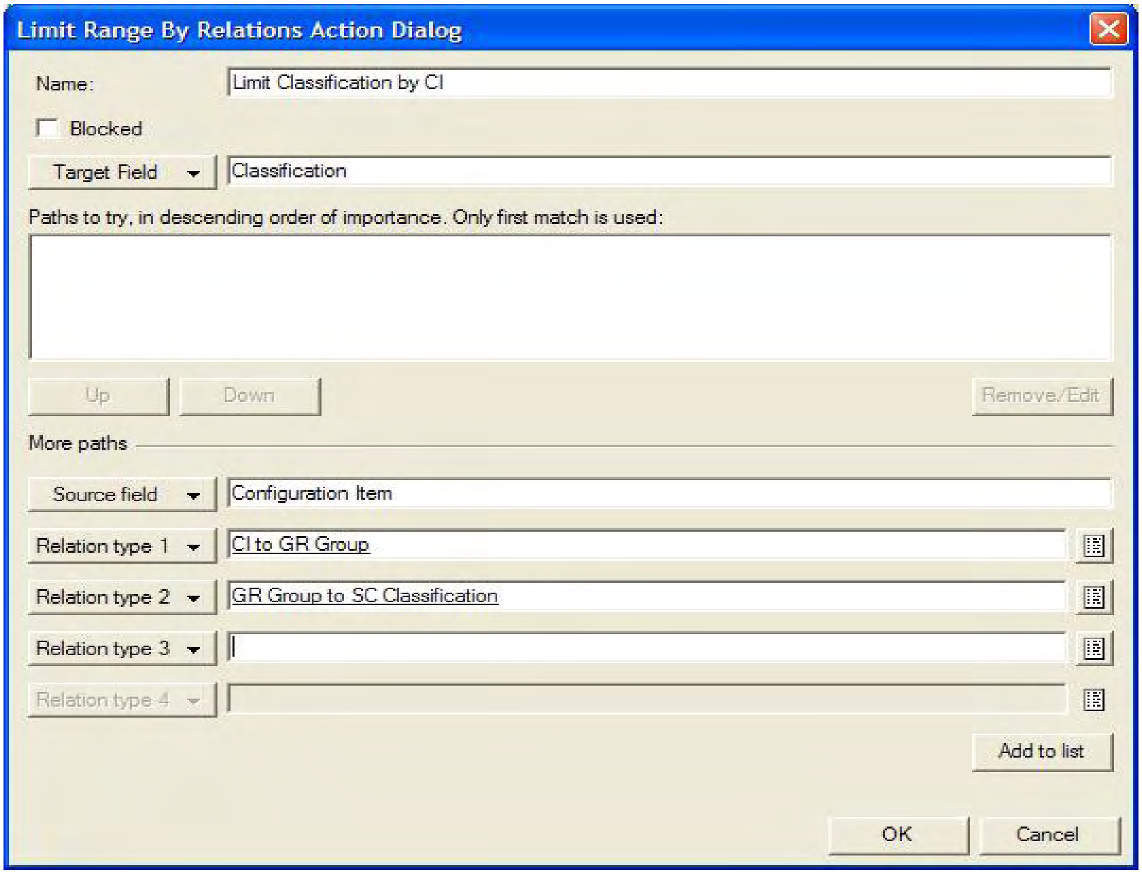
Add action "Limit field value range by relations":

Figure 39



Limit target field "Classification" by source field "Configuration Item" based on Generic Relation Types "CI to GR Group" and "GR Group to SC Classification":

Figure 40



In a service call form, modify CI field and the UI rule will be triggered. A quick find dialog will popup with Classification Code values restricted by chain "CI to GR Group" -> "GR Group to SC Classification"

Figure 41

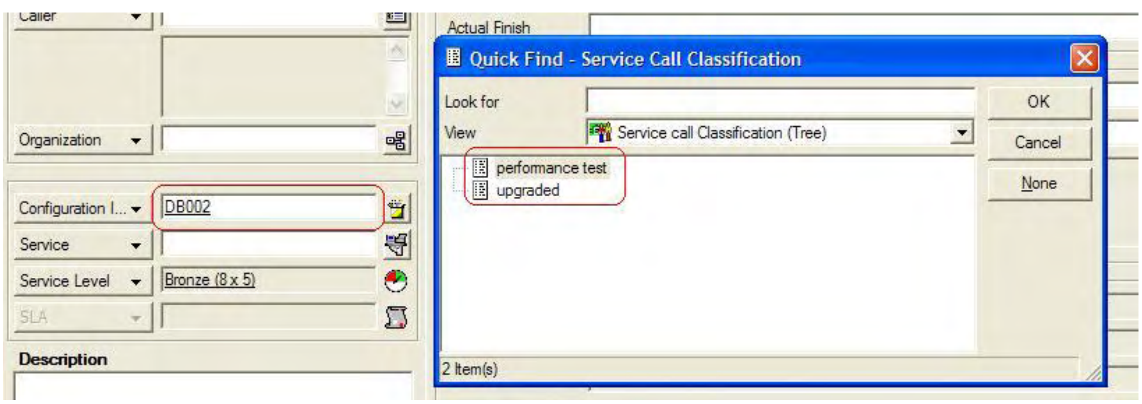


Figure 42

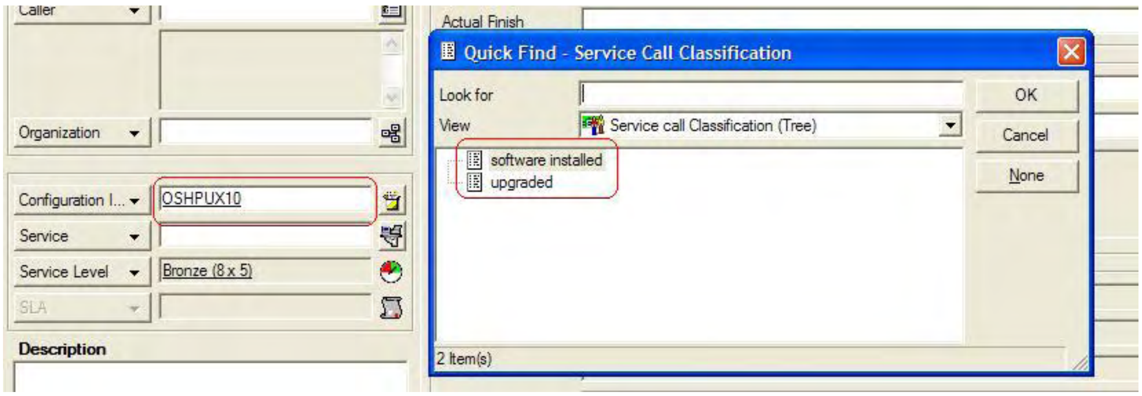
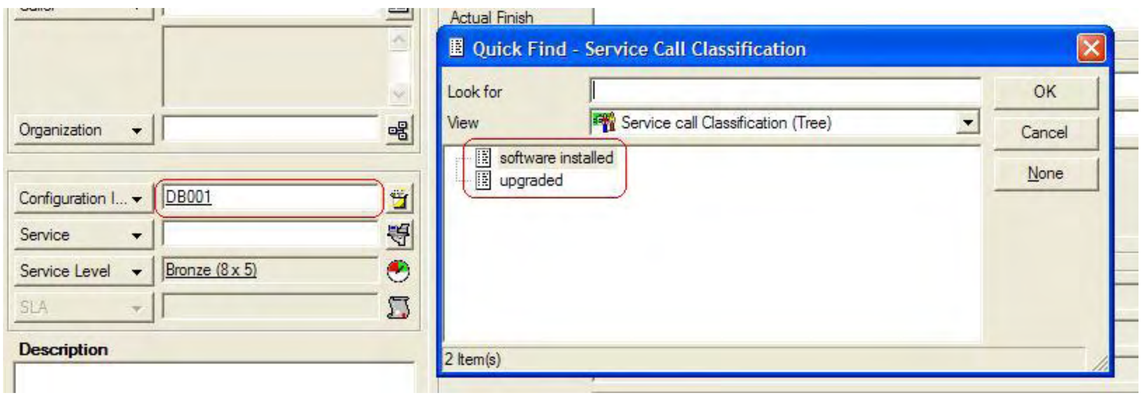


Figure 43



Known Errors and Limitations

Generic association mechanism is different from existing relations in Service Desk.

Service Desk has defined many relations according to industrial best practices which express the standard business relations. E.g. service call relate with problem, problem relate with change, change can have a set of work orders, work order is related with CIs, etc.

Generic Relation is intended to allow user add additional customized relations between any Service Desk items. The relations can be viewed and also used in UI Rule Action "Limit field value range by relations".

The additional Generic Relations is independent with existing relations.

8 Service Pack 18

This chapter contains information previously provided with Service Pack 18.

Dynamic Forms

Introduction

Before Service Pack 18, the form for each item type in Service Desk had a fixed layout. As a consequence, a form often contained many fields that were only used under specific circumstances. For example, some fields are only used when the item has a certain status, while other fields are only used when the item is in a given category. Even though an item may never be put in a state where those fields would need to have a value, they had to be put on the form anyway, simply for the reason that the item could potentially be put in such a state.

With many fields on a form, the likelihood of input in the wrong field increases: when there are many fields to choose from, it is very well possible that one field is mistaken for another. With many fields on a form, it is also likely that fields are overlooked, causing them to remain unused.

Dynamic forms are a new feature of Service Desk that make it possible to change the layout of the currently opened form based on values selected in the form. It allows the administrator to make fields visible only within the context in which they are needed, and not anywhere else.

Getting Started

The form layout change is controlled by a UI rule action that uses a table that maps field values to forms. This table is called a generic relation type. Whenever the appropriate field value is chosen, the UI rule searches the generic relation type for the field value. When the current field value is found, it effectively inserts the fields and tab-pages from the associated form into the opened form. Please refer to *ITSM008088* for detailed information on Generic Relations.

To control the position on each tab page where fields from a form selected by the UI rule are put, a new layout element has been introduced, the insertion marker. Next to controlling the position where fields are to be inserted on each tab page, the insertion marker also serves as an indication that the current form is eligible to be changed by the new UI rule action.

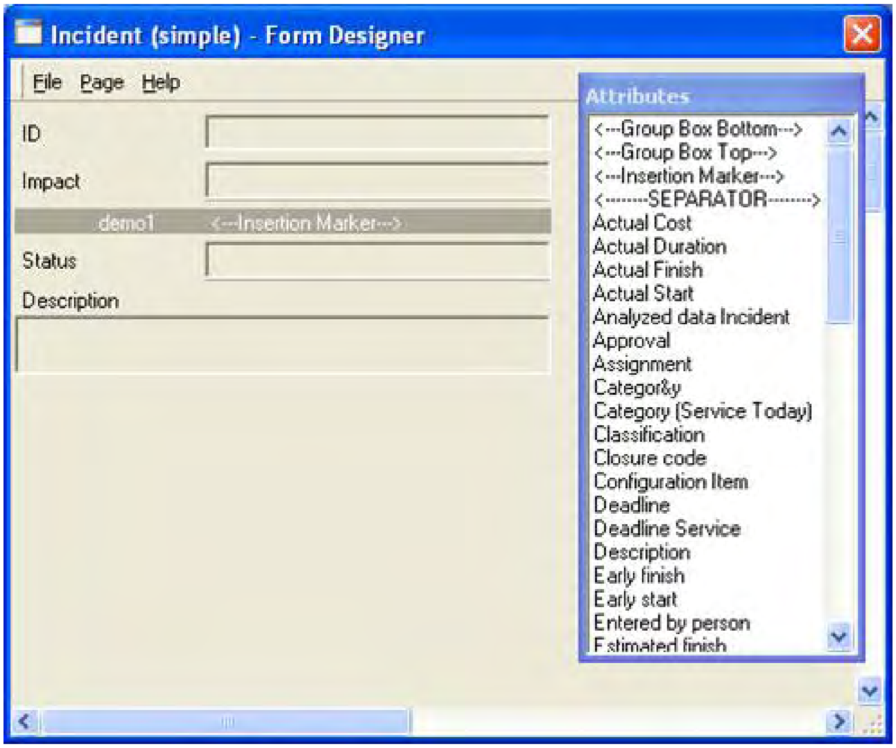
The next sections describe the components that comprise the dynamic forms feature in more detail.

Insertion Markers

The insertion marker is a new type of field that you can use in the form designer. The form designer is accessed by going to the **Administrator Console** → **Presentation** → **Forms**;

selecting an item category; and then opening a form. The next screenshot shows the Form Designer with the Insertion Marker field available in the Attributes list:

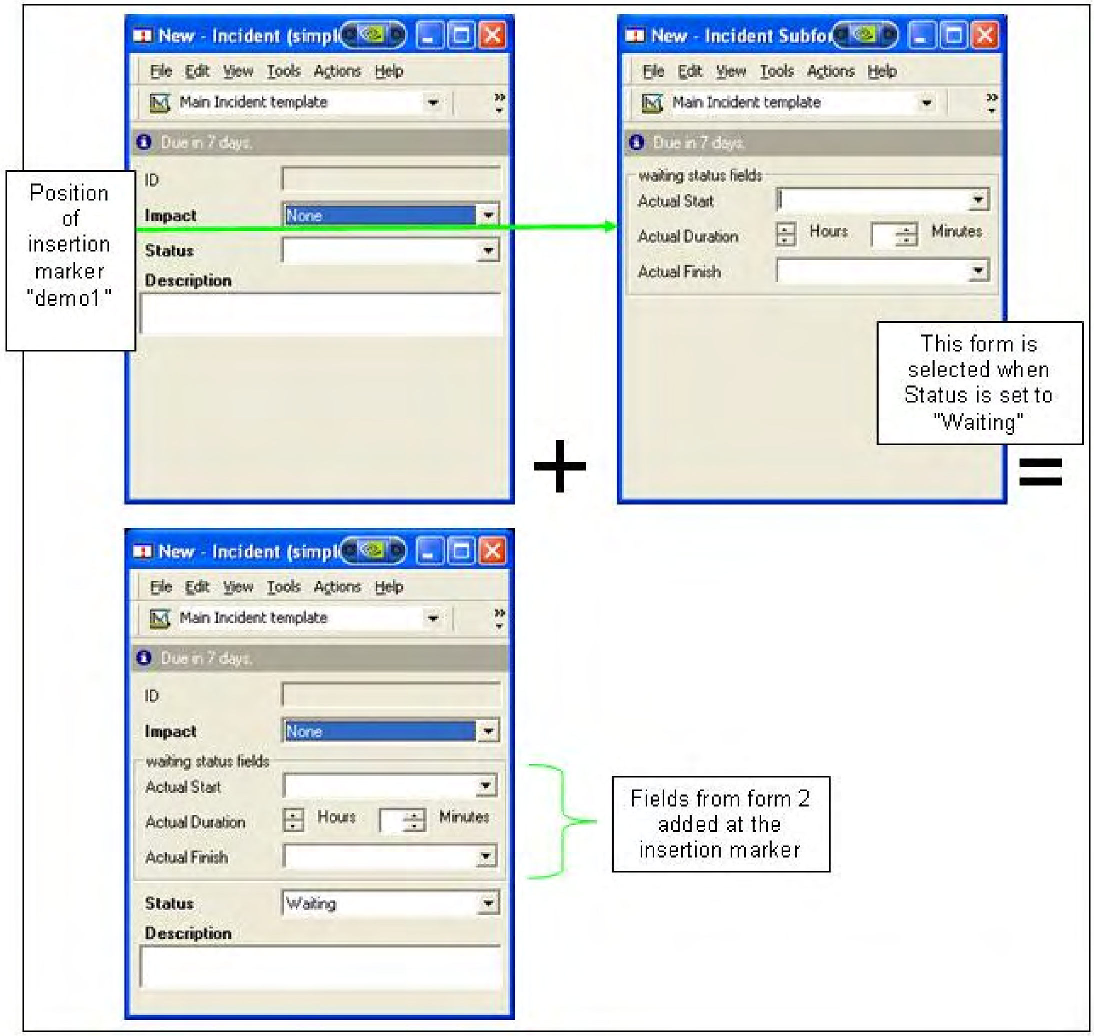
Figure 1



The insertion marker is used as more than a mere indication that a form may have fields added to it after the form was opened. It also marks the spot on the current page where fields from the inserted forms are added in case the tab page on which the insertion marker appears has the same name as a tab page of a form that is inserted.

The following screenshots illustrate how the insertion marker controls how one form is inserted into another, thus changing the layout dynamically. Figure 1 shows the form as it is rendered when used to open a form. The second shows fields to add to this form. The third picture shows the two combined: the fields from the second form are inserted at the marker in the first form.

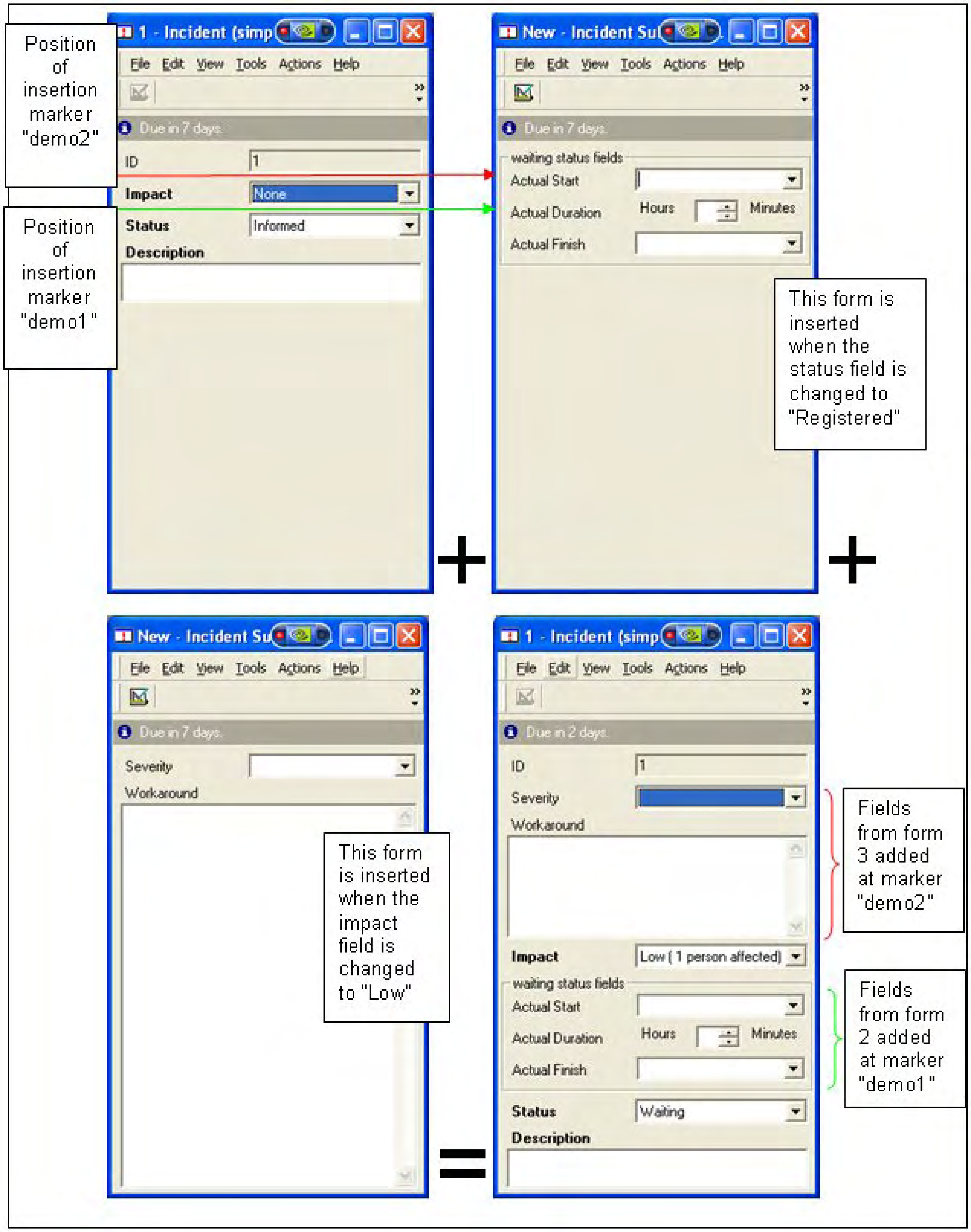
Figure 2



Every insertion marker must have a name. UI rules that change the currently opened form use the insertion marker name to identify any form on which they are to be used. A UI rule action that changes the current form will only trigger if it contains a reference to the name of the insertion marker on the current form. In the example above, the currently opened form would not have changed if the UI rule did not reference the insertion marker named "demo1" (as seen in Figure 1).

You can put multiple insertion markers on a single tab page. This allows multiple UI rules to manipulate part of the form. The following figure illustrates how a page may change through value changes on more than 1 field. Expanding on the previous example, it shows how introduction of an insertion marker called "demo2" allows a second rule to be changed.

Figure 3



You can use a maximum of 4 insertion markers on one form and each marker should be uniquely named.

Form Insertion Rules

The section on "Insertion markers" contained a few examples on how dynamic form changes combine two form definitions to change the layout of the currently opened form based on user input. These examples demonstrate a few of the rules by which forms are combined. The next subsections summarize all the insertion rules.

The form designer has a clear distinction between the main page and the tab pages of a form. As far as the form insertion rules are concerned, the main page is simply the first of a series of pages. It is only different in the way it is rendered on screen and in the fact that the main page has no title. To the form insertion rules, the main page and the tab pages are conceptually the same. Therefore, there is no difference between the main page and the tab pages in terms of how they are affected by the insertion rules.

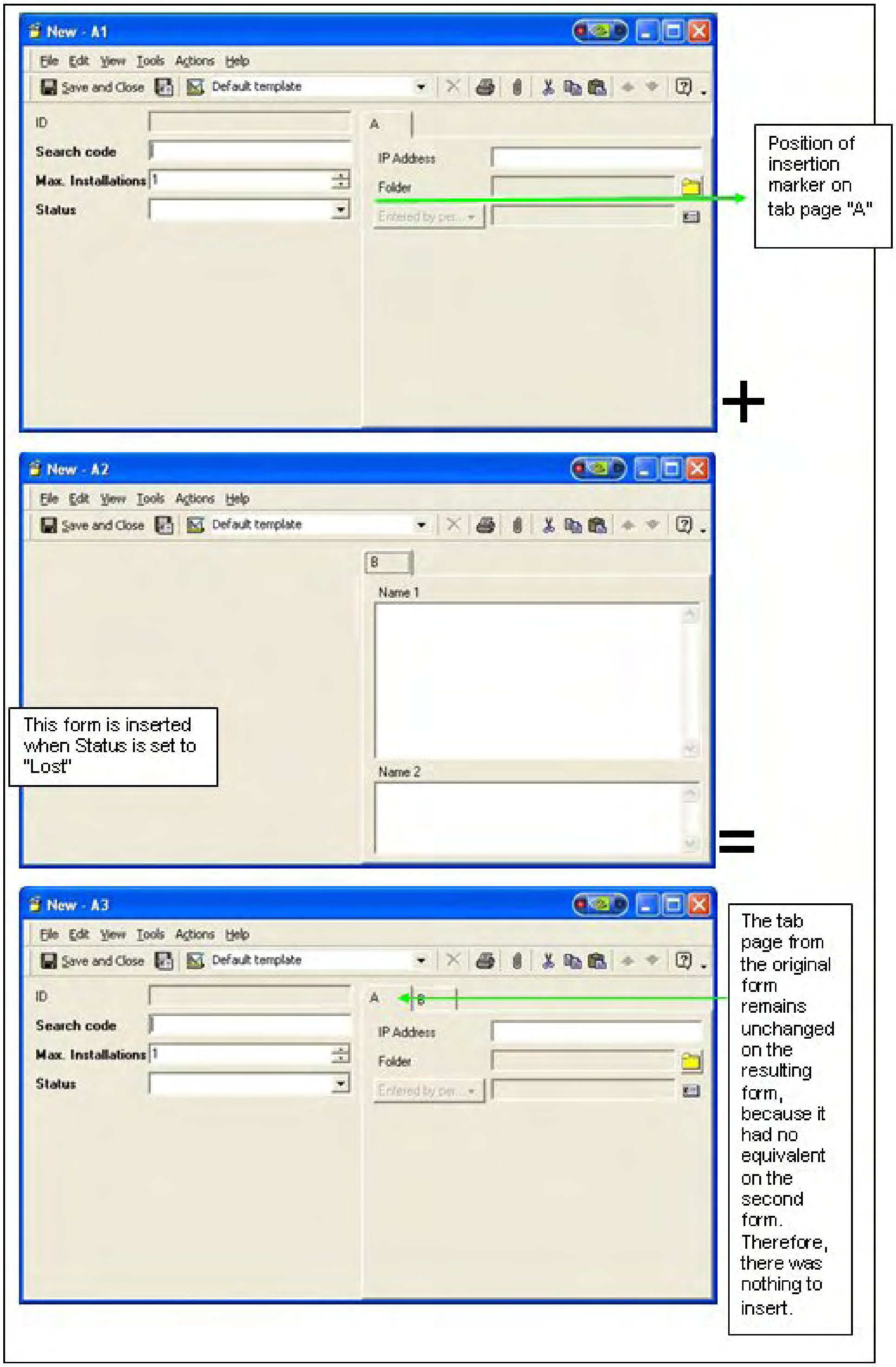
Dynamic Form Changes Only Affect Forms with Predefined Markers

A Form will never be changed by a UI rule unless it contains an insertion marker. No Form will ever be changed until there are UI rules that change them. The currently opened Form is only changed when it contains an insertion marker that is referenced by a UI rule action that executes a form change. However, even though an insertion marker may appear only on a single page of a form, all the pages of the current form will be affected when a form changes, even the pages without markers.

Dynamic Form Changes are Evaluated Page by Page

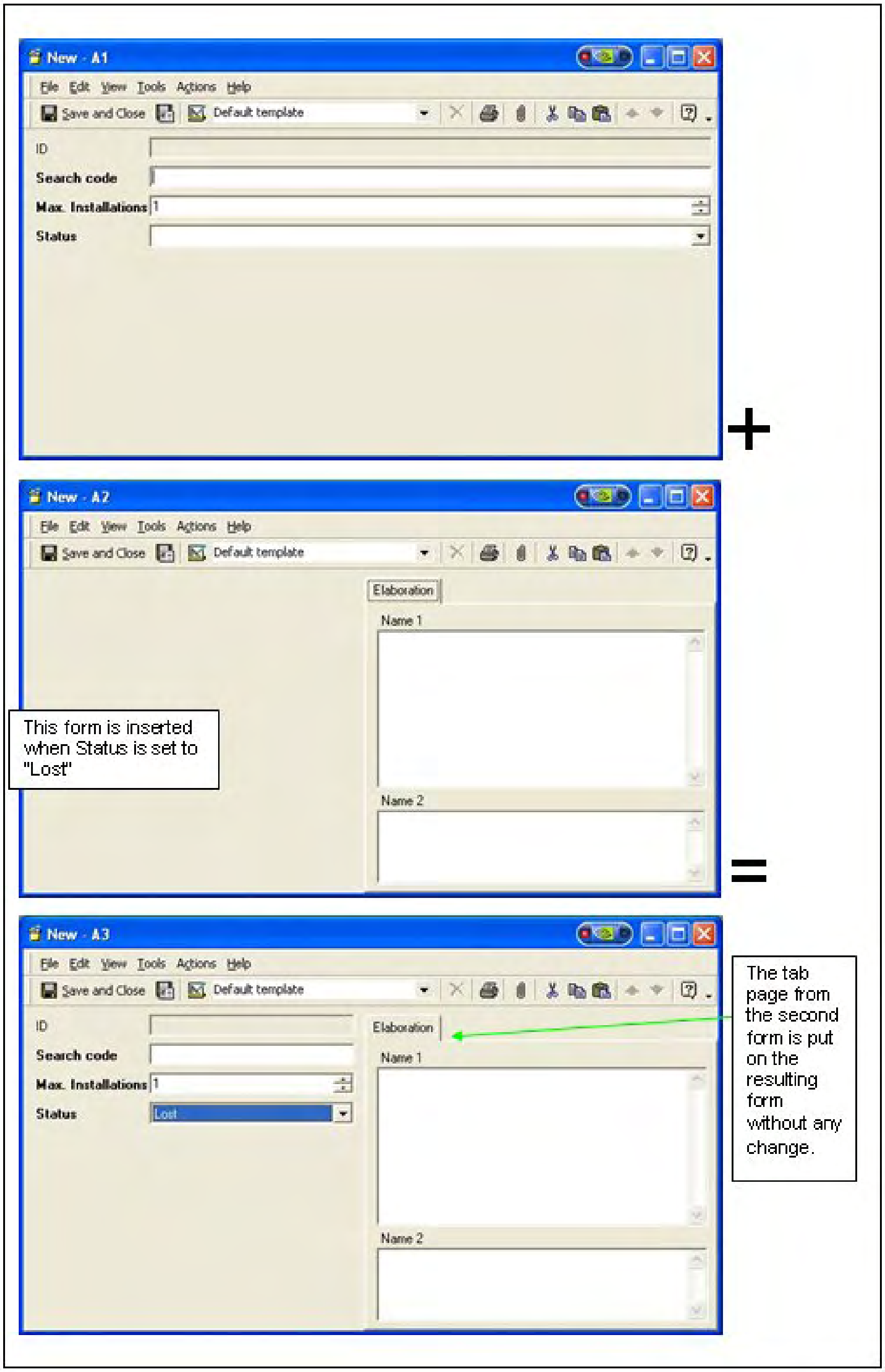
At the insertion marker, fields are inserted on the same tab page of the form selected by the UI rule (if any). If a tab page with a given title contains an insertion marker, but the inserted form does not have a tab page with the same title, the tab page on the resulting form will not change. This is because there is nothing to insert on that tab page. The following diagram illustrates this:

Figure 4


New - A1
File Edit View Tools Actions Help
Save and Close
Default template
ID
Search code
Max. Installations
Status
A
IP Address
Folder
Entered by per...
Position of insertion marker on tab page "A"
New - A2
B
Name 1
Name 2
This form is inserted when Status is set to "Lost"
New - A3
The tab page from the original form remains unchanged on the resulting form, because it had no equivalent on the second form. Therefore, there was nothing to insert.

Pages on the form selected by the UI rule are added to the resulting form when there is no page with the same title on the original form definition. This is illustrated by the diagram below:

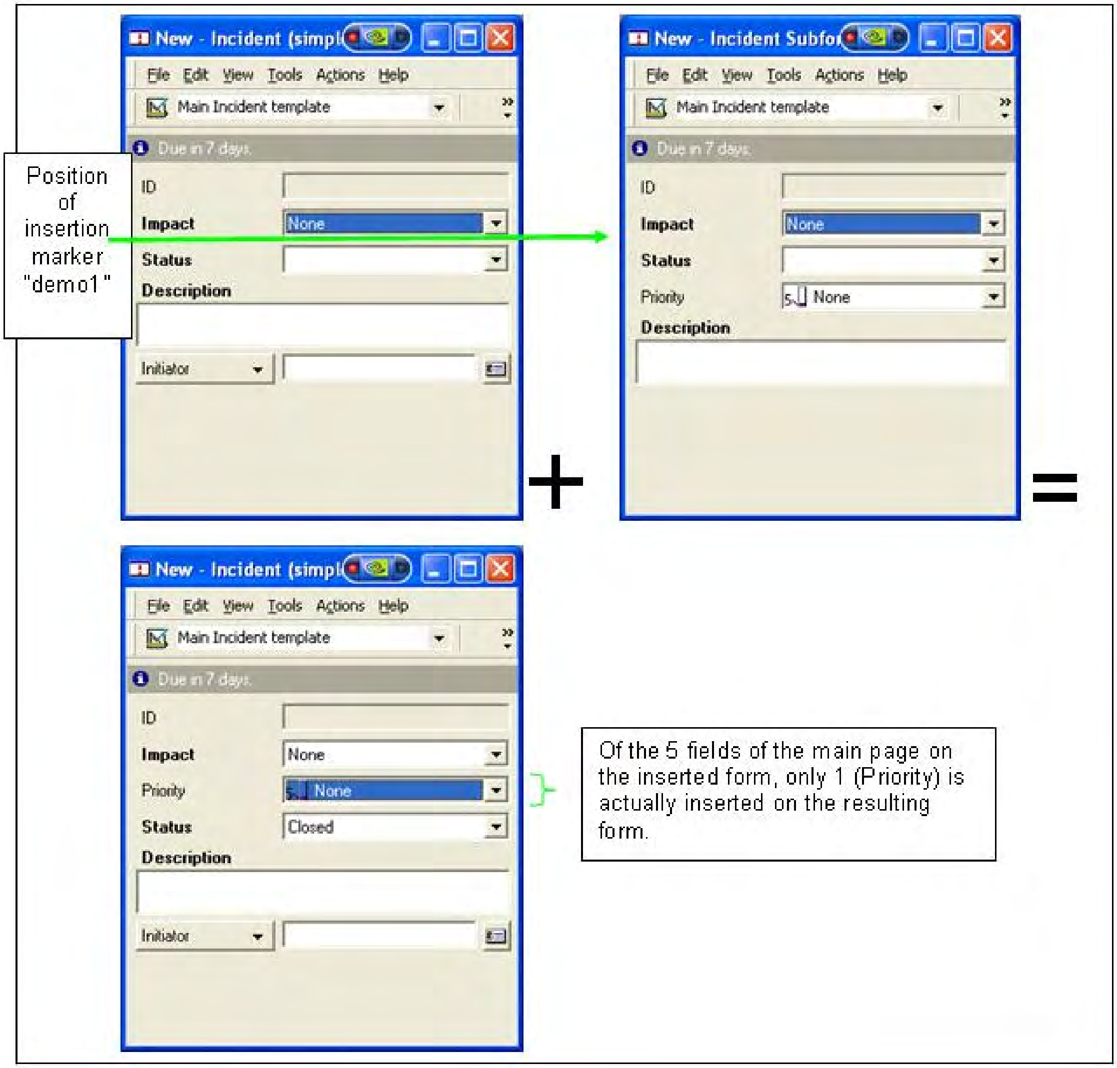
Figure 5



Dynamically Added Fields Exclude Those that are Already on the Same Page of the Original Form

Fields from the form selected by the UI rule that are already on the same page of the currently opened form are not put on it again. The following diagram illustrates this:

Figure 6



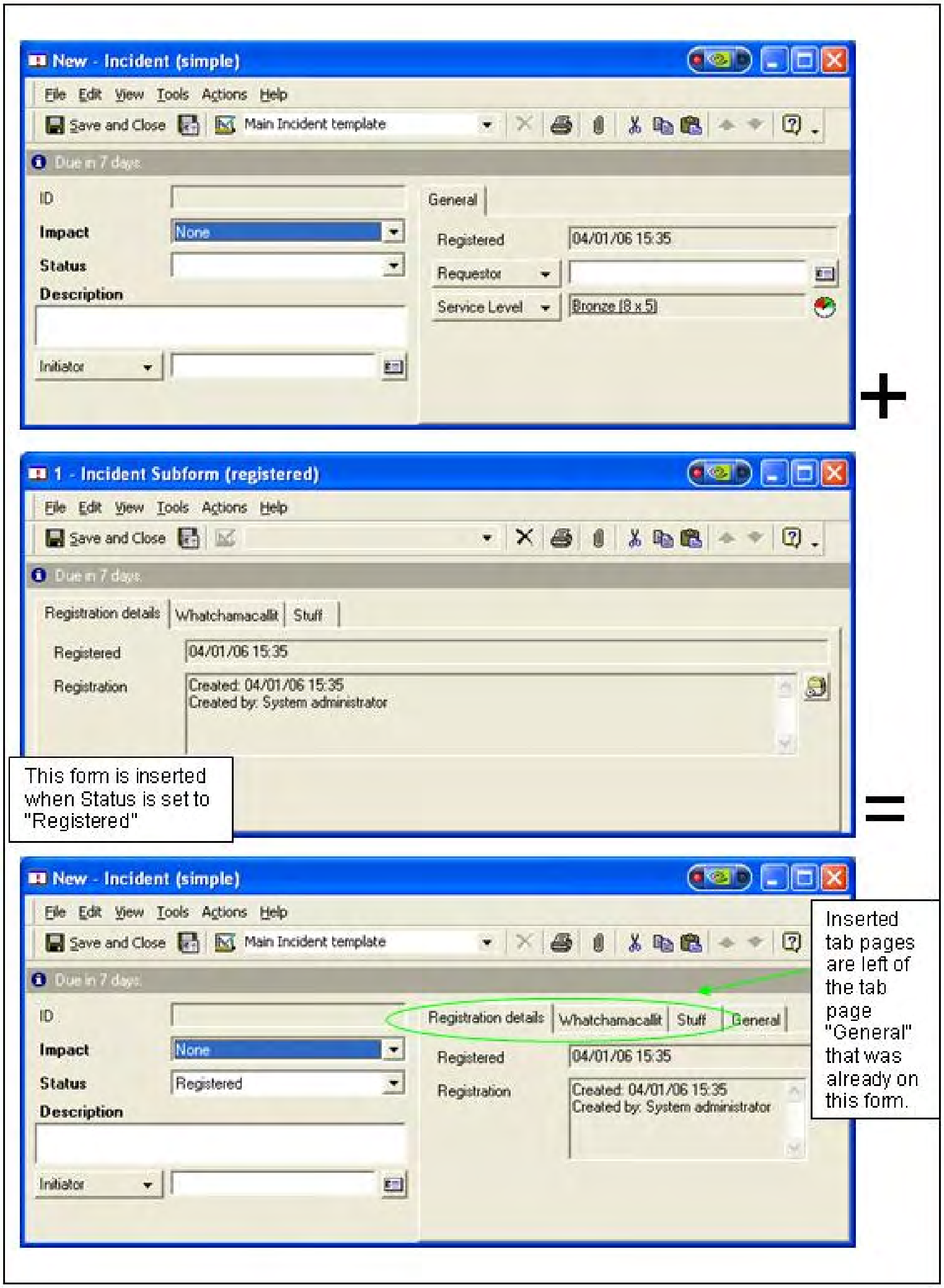
Please note that if multiple markers are used and the UI rule selects a field to be inserted at the second marker that has already been inserted at the first marker, the field will appear on both markers. This is because the field was not on the original form definition.

If it is likely that two insertion markers on the same page introduce the same fields in a certain scenario, it makes much more sense to only introduce new tab pages using these markers.

Dynamically Added Tab Pages are Inserted on the Left Side of the Form

If a tab page on the form selected by the UI rule does not appear on the currently opened form, it will be inserted as the left-most tab page. This is illustrated in the following figure:

Figure 7



Fields That Need to be Dynamically Added to a Page Without a Marker are Appended to the Page

If a tab page on the currently opened form has the same title as a tab page without the referenced insertion marker on the form selected by the UI rule, the fields from the tab page of the selected form will be appended to the tab page.

Effectively, this works the same as when the insertion marker is the last field of a page.

Insertion Markers on Inserted Forms are Ignored

When fields are inserted into a page, any insertion markers among the fields to add are stripped away. This prevents a new insertion marker from being introduced inside the section that was put in place of the insertion marker. In other words, it is not possible to nest insertion markers.

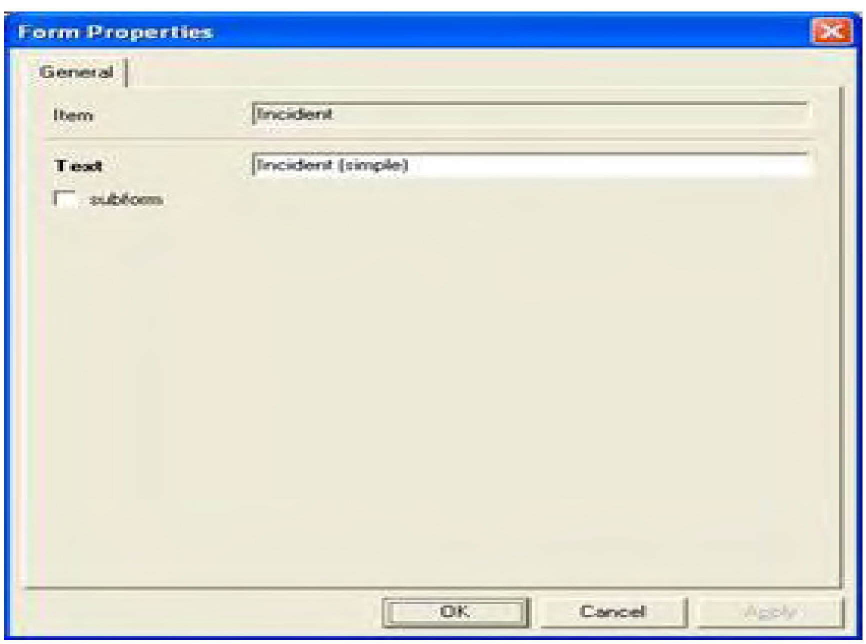
Sub-Forms

As explained in the previous chapter, dynamic forms combine form definitions to change forms on the fly. As a consequence, the number of form definitions can increase drastically with dynamic forms. Many of the new form definitions used for dynamic forms only have a use within the context of dynamic forms. These forms will never be used by themselves to open an item with. For that reason, forms are now divided into two types: regular and sub-forms.

Regular forms have always been in Service Desk. Sub-forms are a new type. These are different to regular forms only in that they cannot be used to open an item with. Another small difference is that the form designer will warn the administrator when a sub-form is saved that has an insertion marker on it. Insertion markers only have a use on regular forms: they are ignored on sub-forms.

Because the difference between regular and sub-forms is very small, any form can be changed into a sub-form and any sub-form can be turned back into a form. To change the type of a form, select the "Properties" from the menu in the form designer (which is what you see when you open a form from the **Administrator Console** → **Presentation** → **Forms**).

Figure 8

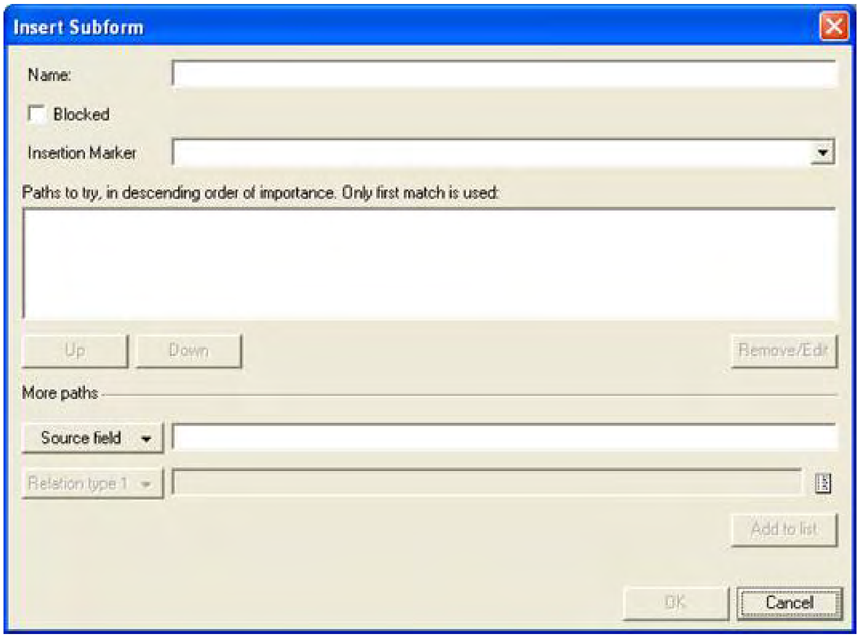


You are advised to define a form to be a sub-form when it is used within another form only.

The Insert Subform Action

At the heart of the dynamic forms functionality is the "Insert subform" action. This action can be invoked by a UI rule to change the layout of the currently open form. The following screenshot shows the dialog in the UI rule wizard that is shown when an "Insert subform" action is selected. The meaning of the fields of the dialog that is shown in the screenshot is explained below.

Figure 9



The Name field is used to describe this action. It is the name by which this action will be known in the UI rule wizard on the Actions page.

The blocked field allows you to (temporarily) disable this action.

The "Insertion marker" field will contain the names of all the insertion markers of Forms defined for same item category as for which the current UI rule is defined, in as far they are not used by other "Insert subform actions". For example, for Person, three forms are defined. The first form contains an insertion marker called "mark1", the second one has a marker called "mark2" and the third one a marker called "mark3". For Organization, a form is defined with an insertion marker called "m1". When you define an "Insert subform" action for a Person, the dropdown box will contain "mark1", "mark2", and "mark3". It will not contain "m1" because it is on a form that is used in a different item category. Once you have defined a UI rule for Person that executes an "Insert subform" action that uses "mark1", the UI rule wizard will not show "mark1" in any dialog defining an "Insert subform" action for another UI rule on Person. This is to prevent two UI rules that change the current form from intervening with one another.

"Paths to try" is a way to select multiple values easily with Generic Relations. Each path has a "Source field" and a "Relation type".

When you click on the "Source field", it will show a dropdown box of all fields of the current item type for which outgoing generic relation types have been defined to a form. Once a value is selected for the "Source field", the "Relation type 1" field will become active. It will be populated by all outgoing generic relation types from the field selected as "Source" field to

Form. Effectively, these are mapping tables, specifying which form to use whenever a given value is active.

When a value is chosen for the "Relation type 1" field, the "Add to list" button becomes active. When pressed, the value selected in the "Relation type 1" field is added to the "Paths to try" box.

Effectively, the "Paths to try" can be considered a secondary type of condition of the UI rule. For this reason, you are advised to use the field that is selected as "Source field" as part of the condition of the UI rule that triggers this action as well. In the condition, it should only be ascertained that the field has a value. The administrator should not limit execution of the UI rule that controls dynamic form changes based on any particular value of a particular field, otherwise the "Insert subform" action will not be triggered when any other than the value in the condition is chosen in the form.

For example, a generic relation type defines relations between the Status field and Forms. This generic relation type is used in an "Insert subform" action. The UI rule that calls this action should then include the condition "Status is anything". If the condition would include "Status equals 'somespecificvalue'", the only generic relation in the generic relation type that will be used is the one that relates from 'somespecificvalue' to a form. In this case, even when the generic relation type defines other outgoing relations of the Status field as well, they would never be used, because the action is never executed when another value is chosen.

You can add multiple paths but only the first match is used.

Configuring Dynamic Forms

This section outlines the steps to take to configure dynamic forms.

Create Forms with Fields for a Specific Context

Go to **Administrator Console** → **Presentation** → **Forms**; select an item category; right-click to create a new form:

This opens a window like the one seen below:

Figure 10

- As name, use a prefix that includes the item type (service call, problem, CI, etc) for which the form is created. This will make it easier to identify the form later on. Once you have entered a name, the OK button will be activated.
- Check the sub-form checkbox.
- When you click OK, you will enter the Form Designer. Add tab-pages and fields as required. It is usually easiest for the users when a dynamic form change introduces new tab pages.

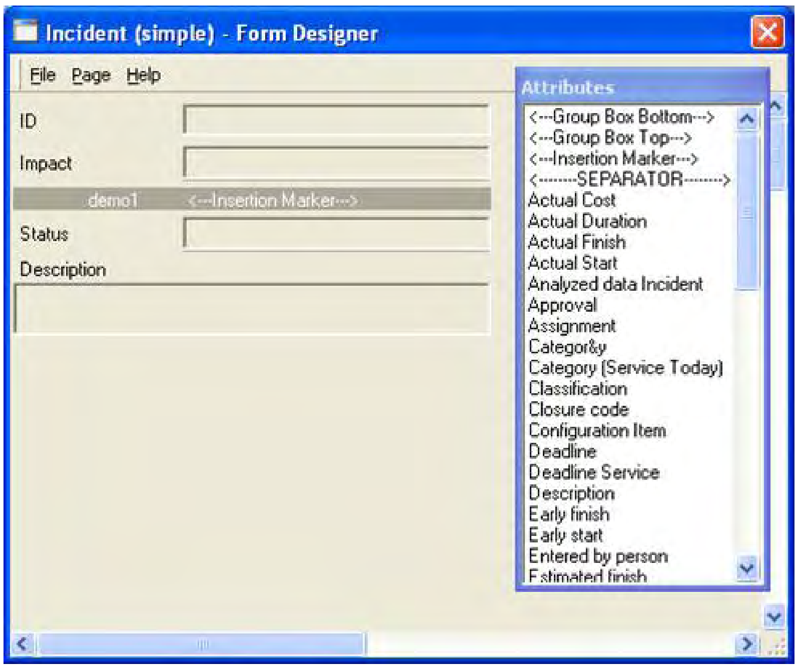
It is recommended to use type name as prefix which will make it easier to locate the subform when defining relations. For example, use "subsc_" as the prefix for Service Call subforms.

Make a Form Ready for Form Insertion

Go to Administrator Console > Presentation > Forms, edit an existing form (that is not a sub-form) or create a new form.

This opens a form like the one seen below:

Figure 11



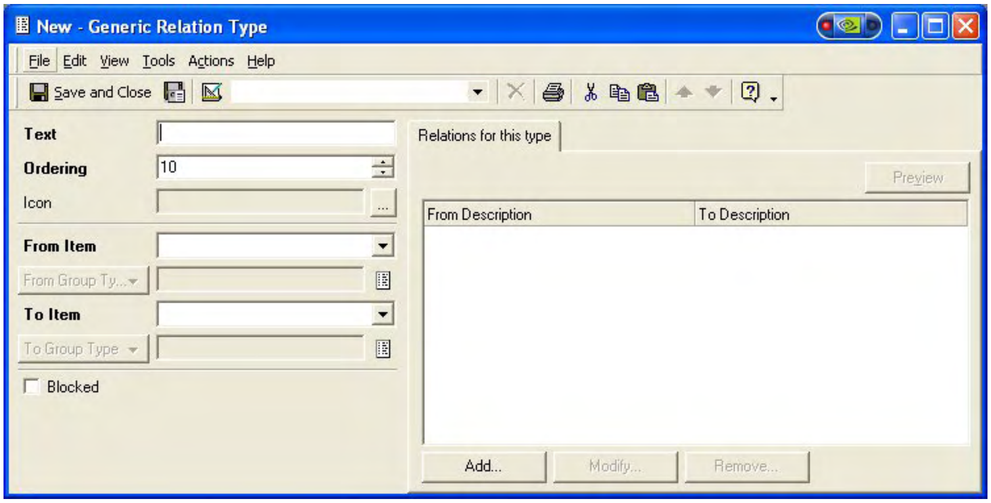
- Place an insertion marker field on any page of the form. As explained earlier in this document, the insertion marker indicates the spot on that page where fields will be inserted from the page with the same page title on a sub-form (if any). Insertion markers must have a name. This name is unique per page, because this name uniquely identifies a spot on the page where new fields must be inserted. You can use a maximum of 4 uniquely named insertion markers on a form.
- Make sure that the field that you want form changes to depend upon is present on this form.

Create a Generic Relation to Associate Field Values with Forms

Go to **Administrator Console** → **Data** → **Codes** → **General** → **Generic Relation Type**, right-click to create a new generic relation type.

This opens a dialog like the one seen in the screenshot below:

Figure 12



- As "from item", select a field of the item category for which you have been editing Forms up to this point. This field will determine which sub-form is to be used. For example, Service Call Status, CI Category, Classification. Some fields appear in multiple item categories (e.g. Folder), so they do not have a prefix indicating the item type in the "from item" list.
- As name, it is recommended to use the convention "<field> to form" where <field> is substituted by the name of the field that is the "from item" side of the generic relation type (in other words, for which the generic relation type is an outgoing one).
- As "to item", select "Form".

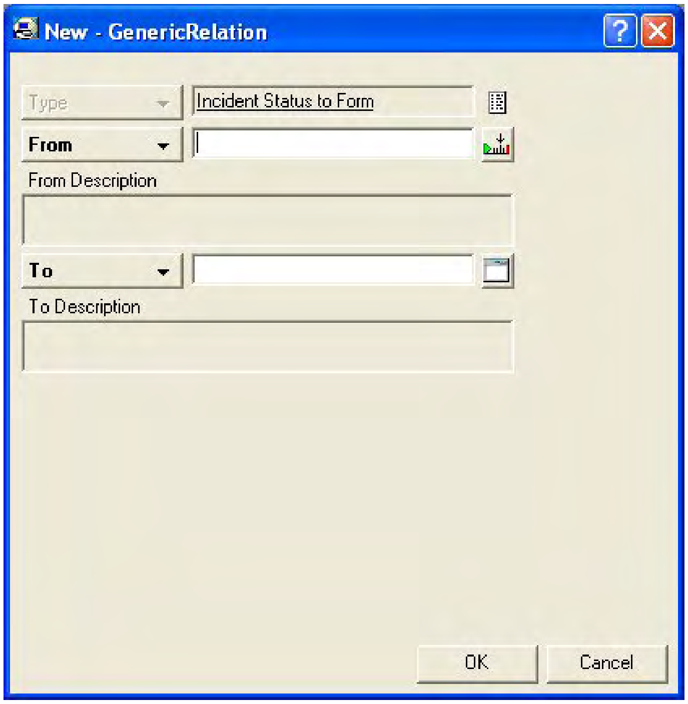
Keep the generic relation type open for the next step.

Set Up Mappings from Item Field Values to Forms

In the generic relation type created in the previous step, press the "Add" button in the right pane.

This opens a dialog like the one seen in the screenshot below:

Figure 13



- Choose a from-item from the list of values. If a "from item" value is used more than once, only the first "to-item" value found will be used by the dynamic form.
- Choose a to-item from the list of forms. Forms hold no reference to the item for which they are created. Therefore, you will see a list of all forms when you set up the mapping here. It helps to sort on the sub-form property first, next on the name. If a specific prefix is used for subforms, first type the prefix (e.g. "subsc_") in lookup field and then click the quick find button. This will help to limit the list of forms.

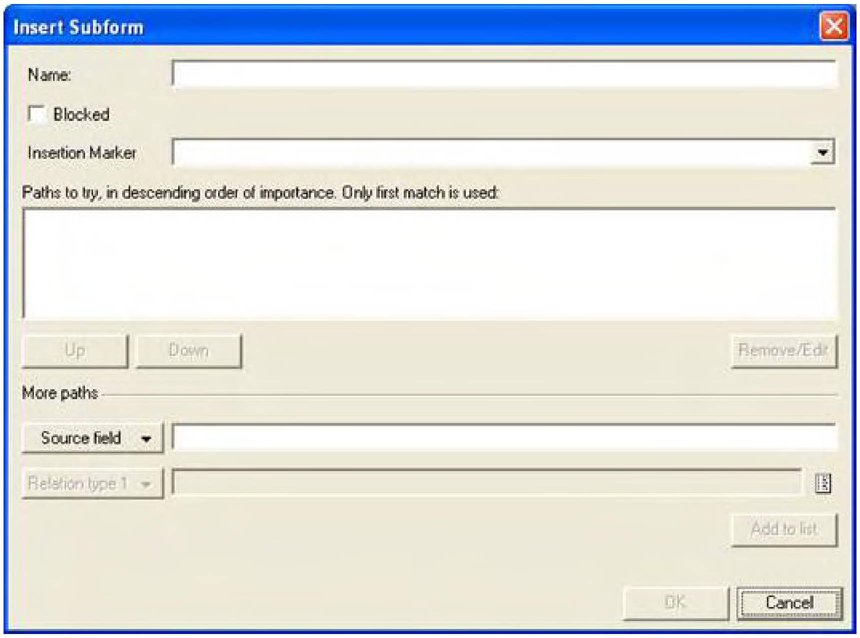
Create a UI Rule to Trigger Form Insertion

Go to **Administrator Console** → **Business Logic** → **UI Rules**. Select the item category for which you created the sub-forms. Create a new UI rule:

- As triggers, select all except "when a form is closed", press next.
- As condition, select "<field> is anything", where <field> is substituted by the name of the field used as "from item" when you created the generic relation type earlier. Press next.

Add an "Insert subform" action as seen in the screenshot below.

Figure 14



- The value for Name field is the text that you will see in the list on the actions page of the UI Rule wizard. It can be anything you like.
- The dropdown box of the "Insertion marker" field will contain all names of insertion markers (such as the one added to the default form, as described in the section "Make a form ready for form insertion") for which no "Insert subform" action has been defined yet. In other words, as soon as you define an "Insert subform" action for a marker called "mark1", you can no longer create any other "Insert subform" action to operate on the same insertion marker.
- As "source field", select the field name used as "from item" when you created the generic relation type earlier.
- Press the icon to the right of the relation type field. If you only have one generic relation type defined to map fields values to forms, this value will be selected automatically, otherwise you will see a list of generic relation types. Alternatively, you could use advance find to add a generic relation type.
- Press "add to list". Next, press ok to save the "Insert subform" action.
- Press next in the UI rule wizard. (Note: in case you selected the event type "when a form is closed" despite the instructions not to earlier, the UI rule wizard will complain that you may not use this event type in combination with an "Insert subform" action. You will then have to go back a couple of pages to deselect this event type).
- Choose a name and rule category, press finish.

The form you changed in the step "Make a form ready for insertion" will now respond to changes of values in the field you selected when you carried out the instructions in the step "Set up mappings from item field values to forms".

Known Errors and Limitations

It is possible to define a UI rule so that the same rule has multiple "Insert Subform" actions that update the same insertion marker. The UI rule wizard prevents another UI rule from having an "Insert subform" action that updates an insertion marker that is aleady used by another "Insert subform" action in another UI rule. Unfortunately, the check fails when the two actions are part of the same rule. While this will not break the functioning of the UI rule, it may have unexpected results because it is not determined which action is executed first.

It is possible to delete a generic relation type that is referenced by an "Insert subform" action. For now, you should examine your rules to see if it uses a specific generic relation type before you decide to remove it.

While a form is being changed on screen, users may temporarily see some fields appear as writable that are supposed to be read-only according to the permissions granted by the current user's roles. It is purely a cosmetic issue: there is no way for the user to abuse the fact that the fields seem writable. Furthermore, once the form change is complete, all read-only fields will be grayed out as expected.

ACES cannot be used to migrate generic relation type instances defined on one Service Desk database to another. For now, you will have to recreate these manually when migrating this part of the Service Desk configuration from a staging to a production environment.

Unlimited Custom Fields

Introduction

Service Desk now allows user to create additional custom fields when pre-configured custom fields cannot satisfy their requirement. This document describes how to manage custom fields, and what you need to take into account when configuring custom fields.

Included in this document are instructions for the following:

- Managing custom fields (viewing, creating, activating, and deactivating)
- Migrating from a previous HP OpenView Service Desk version or another Service Desk instance with the same version
- Regenerating the Web API library, which is specific to your own Service Desk environment

The main audience for this document is Service Desk administrators authorized to configure custom fields.

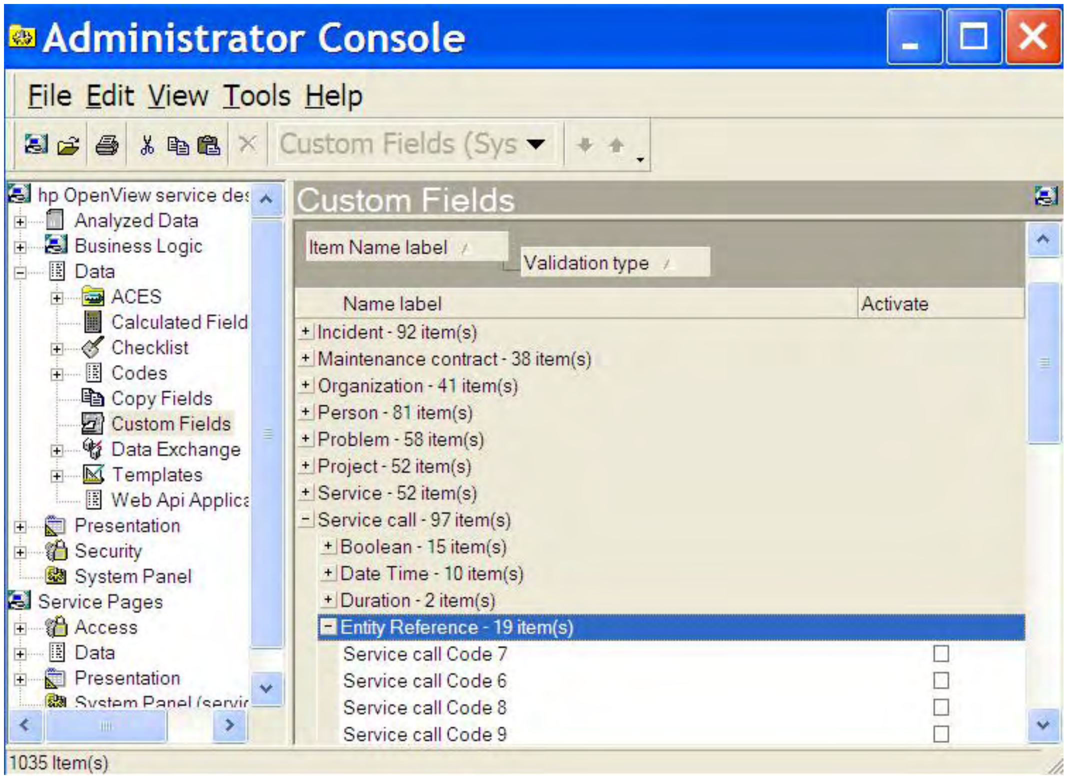
Managing Custom Fields

View of Custom Fields

Before Service Pack 18, HP OpenView Service Desk version 4.5 offered a fixed number of custom fields that could be reconfigured for a specific use. With the new custom fields feature, administrators get the ability to add extra fields to an Service Desk item. This way, the use of custom fields is no longer limited by the number of custom fields offered by Service Desk out of the box. To provide a better overview of these custom fields, this feature also introduces a new view of Custom Fields in the Administrator Console (see Figure 15)

View of Custom Fields can be customized by System Administrator temporarily but will be reset after client restarted.

Figure 15



When and How to Add New Custom Field

The administrator should only add a custom field in case there is no unused custom field of the required type available. To see if there are still unused custom fields for an item, open the Administrator Console, and navigate to Data, Custom Fields (see Figure 15). In the right pane, select the item for which you want to add a custom field. Click on the item to open the tree of existing custom fields per type. If the type of field for which you want to use a custom field does not exist under the item, you will need to add a custom field. If there are custom fields of the type that you require under the item, check if there is a field that is not activated. If there is such a field, re-use it. If all fields are activated of the type you need for the item, you will need to add a custom field.

To add a new custom field, click the **File** → **New** menu item, or select the **New custom field** option from the context menu (which you see when you right-click on an item in the view). This opens the New Custom Field dialog box. In principle, there are two main types of custom fields:

- *Primitive type custom fields*: These are fields that only contain a single piece of data of a pre-defined type. An example of how a primitive type custom field is configured can be seen in Figure 16.
- Entity reference type custom fields: These are lookup fields, meaning that these fields refer to another item in Service Desk. An example of how a entity reference type custom field is configured can be seen in Figure 18.

Adding a Primitive Type Custom Field

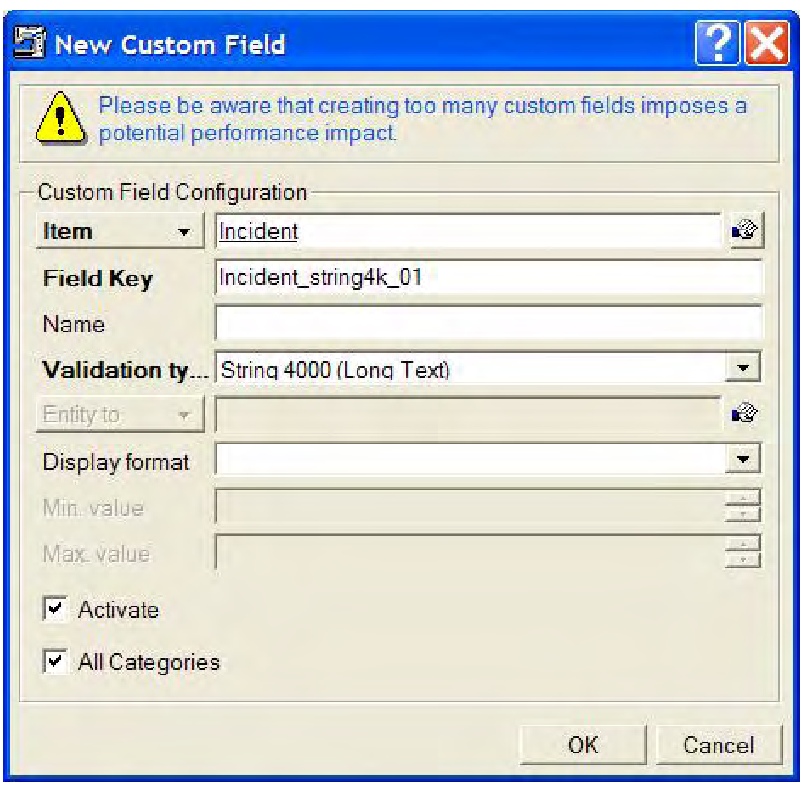
As explained in the previous section, a primitive type custom field is used to hold a single piece of data. The most important in configuring this type of field is determining its validation type. The validation type limits what users can enter in a field.

For example, if you introduce a field called "Item has been inspected", it may seem logical to you that you only expect values to indicate that the inspection has happened or has not, in other words: you are only interested to know if the assertion "Item has been inspected" is true or false. A user could interpret the label of the field differently though when entering data. If the "Item has been inspected" field would accept any form of user input, somebody could enter a time and date on which it has happened, or "I will do this next week", or "I don't know". Now, say that in your business process, you want things to happen once the "Item has been inspected" field is set to "Yes". If the values of the field could be anything, it is hard to set up a database rule or UI rule to respond to this appropriately. Also, reporting may become troublesome when a field can contain many different values that are equivalent to "Yes". By giving it a validation type of "Boolean", it is ensured that only values indicating true or false are entered. Of course, there are fields in which you want the user to be able to enter "I don't know" or "Next week". In that case, you would use a validation type "String". If you only want dates to be entered, you could use a validation type "Date".

Related to the validation type is the display format. In the example of the "Item has been inspected" field, there are different ways in which the values true or false could be shown in a form. You could show this as a checkbox, or a radio button that allows the user to switch between Yes and No. The display format allows you to switch between different representations of the same value in the field. If you do not choose a display format, a default display format will be used.

The screenshot below shows an example of a definition of a primitive type custom field.

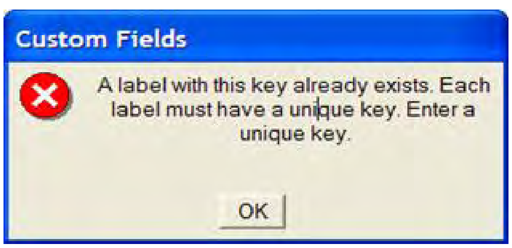
Figure 16



To define a primitive type custom field, do the following:

- Make sure you select values for the fields with names in **bold** in the above screenshot. These fields are mandatory.
- As "Item", choose the Service Desk item to which you want to add a custom field.
- As "Field Key", choose a unique name for the custom field. This is the name that Service Desk uses for this field internally, and also the name that you use to access this field in the Web API.
- Please note that after you have added a custom field, you cannot change the item and field key. It is strongly recommended that you decide on a naming convention for custom field keys, to keep the names generic and readable yet meaningful. Do not put the description of what the field is used for in the field key, because this may be unpractical when you want to re-use to field for a different purpose later. The description of what the field is used for goes into the "Name" field. The field key name should describe what the field is instead. This can be seen in Figure 16 for example: the key "Incident_string4k_01" indicates that it is a custom field of the Service Desk item category Incident, with validation type String 4000, and the first of its kind.
- Select an appropriate value for "Validation type". See *Select Valdation Type* for help on choosing one.
- Please note that after you have added a custom field, you cannot change validation type.
- The remaining fields are optional, and are used the same as in the dialog to edit a custom field.
- Click **OK** to add this custom field, or click **Cancel** to cancel the operation. If the Field Key name is already in use, a dialog box (see Figure 17) informs you that the operation has failed.

Figure 17



Selecting the Right Validation Type for a New Primitive Type Custom Field

The option to create a new primitive type custom field makes it possible to introduce custom fields of types that were not available as custom fields in Service Desk in previous versions. For example, currency and percentage fields were not available as custom fields. As of Service Pack 18, a Service Desk Administrator can introduce these and other new types as new custom fields for a Service Desk item.

The following table lists the available validation types for new custom fields, and what kind of database column is used for their values in the back-end database of Service Desk. Persons that are familiar with the column types of the back-end databases should be able to use those types to determine which validation type to choose. To help the administrator determine the right validation type, the fourth column of the tables provides examples of where this validation types would be used for.

Table 1

Data Type	Column Type		Example of usage and comments
	MS SQL Server	Oracle	
Boolean	NUMBER(1)	DECIMAL(1)	Checklists. Was available as custom field before SP18
Currency	NUMBER(10,2)	DECIMAL(10,2)	Cost of an item
Date	DATE	DATETIME	A date without a time. Useful when the time on the date does not matter.
Date Time	DATE	DATETIME	A date plus a time. Was available as custom field before SP18
Duration	FLOAT(32)	FLOAT(32)	Length of a time interval. Was available as custom field before SP18
Entity Reference	NUMBER(18)	DECIMAL(18)	Any lookup field. Please refer to next section for more information.
Number (Double)	FLOAT	DECIMAL	Any number where the part behind the decimal point matters. Its use is preferable over Number (Float).
Number (Float)	FLOAT	DECIMAL	Any number where the part behind the decimal point matters.
Number (Integer)	NUMBER(10)	DECIMAL(10)	Numbers no longer than 10 positions wide. Was available as custom field before SP18
Number (Long)	NUMBER(18)	DECIMAL(18)	Numbers no longer than 18 positions wide.
Percentage	NUMBER(10,2)	DECIMAL(10,2)	Capacity used of an item
String 1 (character)	VARCHAR2(1)	VARCHAR(1)	Drive letters or initials
String 40 (Short Text)	VARCHAR2(40)	VARCHAR(40)	Short text. Was available as custom field before SP18
String 50 (Name)	VARCHAR2(50)	VARCHAR(50)	Name
String 80 (Text)	VARCHAR2(80)	VARCHAR(80)	Text
String 255 (Description)	VARCHAR2(255)	VARCHAR(255)	Description. Was available as custom field before SP18
String 4000 (Long Text)	VARCHAR2(4000)	VARCHAR(4000)	Comments. Was available as custom field before SP18
String 64K (Memo)	LONG	NTEXT	Very long comments. Use this type in moderation, adding this validation type specifically has the most negative impact on performance. Service Desk is not a word processor: there is usually no need to include long texts as fields (attachments will do nicely). Was available as custom field before SP18.

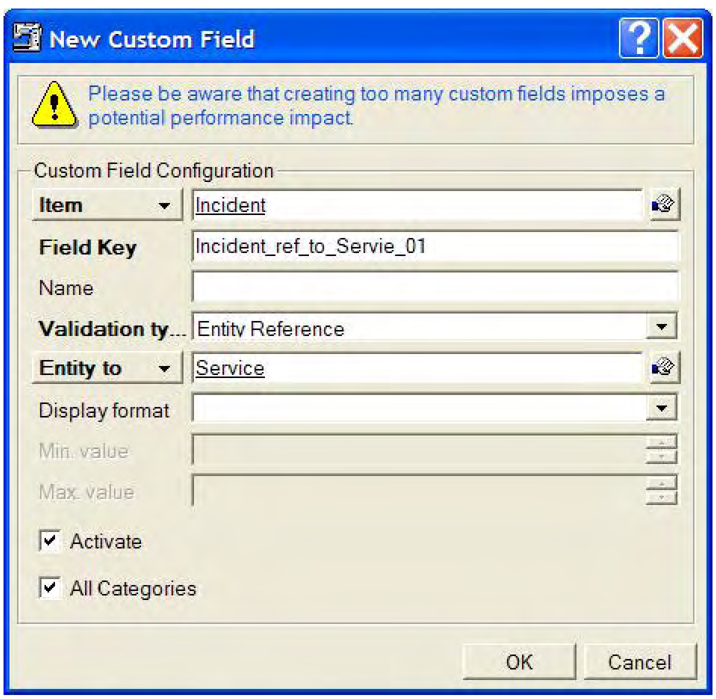
Data Type	Column Type		Example of usage and comments
	MS SQL Server	Oracle	
Time	DATE	DATETIME	A time without a date, useful to indicate something that needs to be done on a given time regardless of the day.
Time of Week	FLOAT(32)	FLOAT(32)	A weekday plus a time, useful for weekly recurring events

Adding an Entity Reference Type Custom Field

An entity reference type custom field is a lookup field. In other words, it holds a reference to another Service Desk item. Internally in Service Desk, the generic term for an Service Desk item is entity, hence the name "entity reference". In the view of custom fields (as seen in Figure 15), all lookup fields of an Service Desk item are now grouped under the validation type "entity reference".

The screenshot below shows an example of a definition of a new entity reference type field.

Figure 18



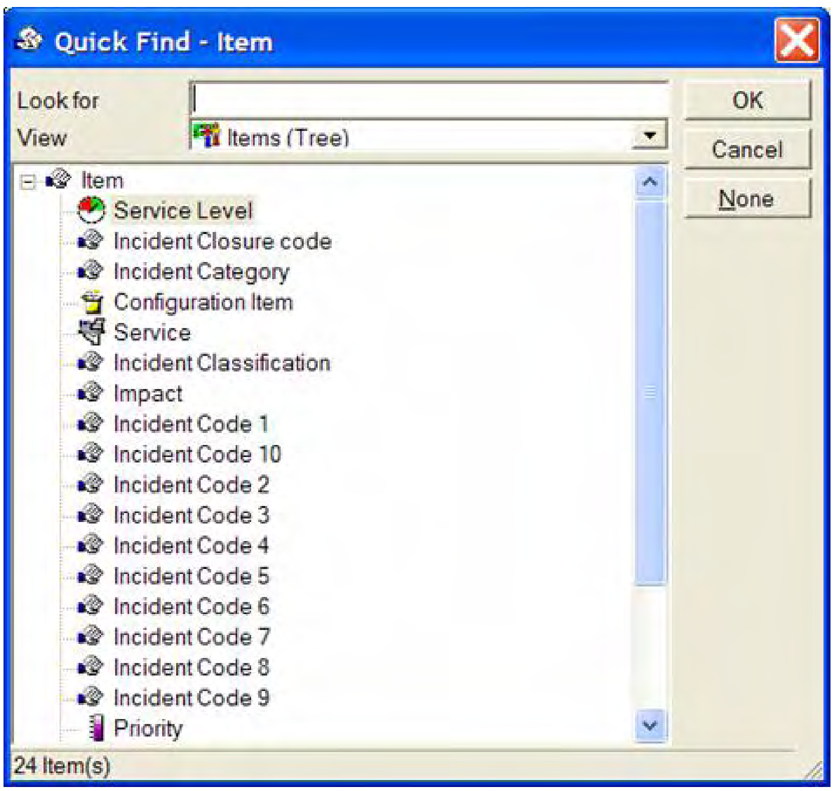
Adding an entity reference type custom field is similar to adding a primitive type custom field. The difference is in mainly in the fields that need to be defined for each. To define an entity reference type field, do the following

- Make sure you select values for the fields with names in **bold** in the above screenshot. These fields are mandatory.
- As "Item", choose the Service Desk item to which you want to add a custom field.
- As "Field Key", choose a unique name for the custom field. The *notes about the field key* found in *Adding a Primitive Type Custom Field* apply here as well.
  An example of a naming convention for field keys for an entity reference type custom field

can be seen in Figure 18. *In this example, the key "Incident_ref_to_Service_01" indicates that this custom field key is used by a field of the Incident item that refers to the Service Desk Item (a.k.a. entity) Service.*

- As "Validation type", choose **Entity Reference**. This causes the label of "Entity to" to become bold, indicating that this field is now mandatory.
- Select the value for "Entity to" using the Quick Find dialog box (see Figure 19). An entity reference type custom field is a lookup field, and this determines what Service Desk item you will see in the lookup field. A custom field of an Service Desk item can only refer to another Service Desk item that is already used in another field. For example, for a Configuration Item, you can select a Person or a Service as value for "entity to". This is because Configuration Item already has lookup fields for those items. You cannot select Service Call or Problem as "entity to", because Configuration Item does not have fields to lookup values of those items.
- Please note that after you have added a custom field, you cannot change validation type and entity to.

Figure 19

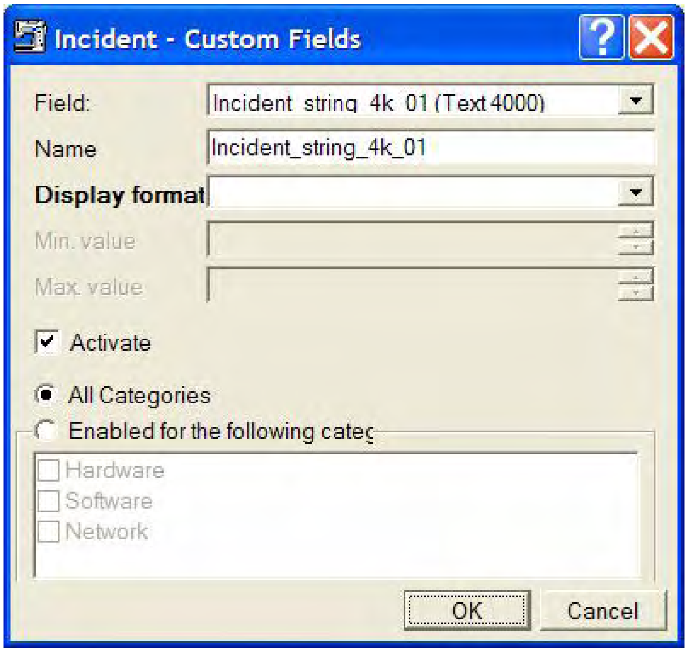


- The remaining fields are optional, and are used the same as in the dialog to edit a custom field.
- Click **OK** to add the custom field, or click **Cancel** to cancel the operation. If the Field Key name is already in use, a dialog box (see Figure 17) informs you that the operation has failed.

Activate or Deactivate a Custom Field

Nothing is changed in this operation. To edit an existing custom field, you can double-click it on the view. Alternatively, you can right-click the custom field, then click the **Edit** menu in the context menu. This opens a dialog box (see Figure 20), from which you can activate or deactivate the custom field, change its name label, change its display format, or restrict its use to a number of categories.

Figure 20



Custom Fields Migration with ACES

When you add a custom field to an Service Desk item, you effectively change the definition of the Service Desk item itself. This definition, which is called the Object Model, is kept in tables of Service Desk. In addition to maintaining these definitions, Service Desk translates the definitions to save the fields of the Service Desk items to tables and columns in the back end database.

Administrators can use ACES to copy custom field definitions from a staging to a production environment. If the source ACES XML file contains some new custom field definitions that are unknown to the target server, those custom fields can be created during ACES importing. The procedure to export and import custom fields in this way is no different from how it was in previous versions of Service Desk.

When custom field definitions are migrated between two systems using ACES, any newly defined custom field from the source system is transferred as a change to the Object Model of the target Service Desk server. If an error occurs, it is possible that some of the changes made to the Service Desk back-end database cannot be rolled back, leaving it in a corrupted state. It is strongly recommended that you back up the database every time before importing new custom field definitions.

Custom field migration with ACES is backward compatible. XML files exported from a previous version can be successfully imported into the new version.

When importing using ACES, make sure that the option "Overwrite Existing Items" is checked if you want these existing custom fields updated. Note that overriding existing items behaves differently in this version than it did in previous versions. In previous versions, ACES updated the existing custom fields when the option Overwrite Existing Items was not checked. The following table shows how ACES handles the custom field importing in different situations.

Table 2

Problems/Situations	Solutions/Rules
Custom field already exists in the target database (same OID).	If you have checked the option Overridden Existing Items, the custom field is updated. Otherwise, the update fails and an error is logged accordingly.
Custom field does not exist in the target database.	Import this custom field.
Custom field key exists in the database, related labels do not have the same OID, and OIDs of custom fields are not the same.	Importing this custom field fails, and an error is logged.

To avoid problems that come from missing items, it is recommended that you make importing custom fields the first task among all ACES importing tasks to be run (for example, importing DB Rule and UI Rule required fields, and so on).

Given that no error occurs during exporting and importing of custom fields using ACES, their definitions will be the same on the source and target Service Desk instance once the operation is completed. However, the data mapping model may be different. This means that for the same custom field, a different column name or even a different table could be used in the back-end database. This is because during the import of custom fields, Service Desk automatically creates the table and column in which the custom field's data is stored, and the chosen column and table name depend on database local factors that cannot be controlled by this process.

Using a Custom Field in Reporting

From the Administrator Console, the administrator has the ability to generate database views in the Service Desk back-end database. These views allow external reporting tools to more easily generate reports on Service Desk. If you use the reporting views and you want to include a newly created custom field a report, you will have to regenerate the reporting views.

The use of the reporting views is described in the section "Generating Database Views" of Chapter 9 of the *Service Desk 4.5 Administrator's Guide*. For your convenience, the information found there is summarized below.

To (re)generate the reporting views, log in to a Service Desk client as an administrator. Open the Administrator Console, and navigate to **System Panel** → **Report Settings**. This opens a window. On the General tab, you will find the option to regenerate the database views. Please note that to (re)generate the reporting views, you need to enter a time zone. This is because the internal representation of times stored by Service Desk may use a different time zone than the one you are generating the reports for.

New Custom Fields with Non-Latin Text Support

Whether the new custom fields get non-Latin text support depends not only on the situation in which the custom fields are created but also on the database type of the target database. The new custom fields get non-Latin text support only if all of the following conditions match:

- It is an Oracle Database, and its Charset is UNICODE (UTF8, AL32UTF8, or UTFE).

- The option Using Extra Space for Non-Latin Text has been applied to the current database.
- It is not a long text custom field. (The current String 4K and String 64K are long text fields.)

  Note that creating custom fields in this way does not ensure full consistency as does developing custom fields manually. In some special operation scenarios, new custom fields never gain non-Latin text support. For this reason, the administrator should consider this issue before creating new custom fields. An example of such a special situation is shown below:

- Configure a Service Desk database. Do not select the Using Extra Space for Non-Latin Text option.
- Upgrade the database. Do not select the Using Extra Space for Non-Latin Text option.
- Create new custom fields.
- Upgrade the database. Select the Using Extra Space for Non-Latin Text option.

Generate the Web API

To access newly added custom fields using the Web API, Java developers need to regenerate the Web API, compile it, and replace the original Web API jar file with the new Web API jar file in their Java applications.
To generate the Web API, developers follow these steps:

- Configure Web API compilation environment.
- Generate new Web API.
- Edit build properties and compile the Web API.

System Requirements and Preparation

Install JDK 1.3 or 1.4

To compile the Web API, users need Java™ Development Toolkit (JDK) 1.3 or 1.4 installed on the Service Desk server machine. For instructions, refer to the *Service Desk Installation Guide*. Suppose you installed JDK1.4 into `C:\j2sdk1.4.2_09`.

Install Apache Ant

Download a binary distribution of Apache Ant from **http://ant.apache.org.** At the time of this writing, Ant version 1.6.5, was the most recent version available for **download**. You should be able to use any version since 1.6.0, however. In the subsequent steps, it is assumed that you extracted the archive with the binary distribution so that its top level directory is called `C:\dev\ant`. If you use a different value, substitute it wherever `C:\dev\ant` is mentioned from this point on.

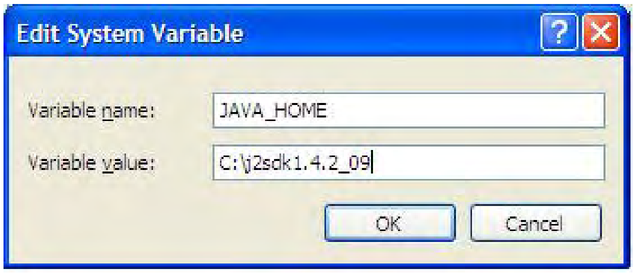
Set Windows Environment Variables (Optional)

Configuring Windows environment variables is an optional task. If users do not change default system environment variables, they must set the variables manually each time before proceeding to the compilation task. The system variables to be configured are `JAVA_HOME`, `ANT_HOME`, and `PATH`. Go to **Start** → **Preference** → **Control Panels** → **System**, open the

System Properties dialog box, navigate to the Advanced tab (See Figure 21), double-click Environment Variables, and add the following system variables:

```
JAVA_HOME= C:\j2sdk1.4.2_09
ANT_HOME= C:\dev\ant
PATH= %JAVA_HOME%\bin;%ANT_HOME%\bin;….
```

Figure 21



The last ellipse (…) in the "PATH" variable represents the original content of that variable.

After setting environment variables, users can open a command console, and execute the following commands for verification.

```
C:\> java -version
```

Figure 22


```
C:\>ant -help
ant [options] [target [target2 [target3] ...]]
Options:
 -help, -h print this message
 -projecthelp, -p print project help information
```

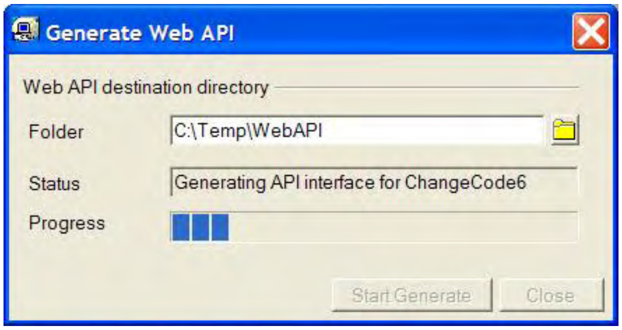
Users will see similar output if these environment variables have been set properly.

Compile the Web API and Javadoc

Generate the Web API

To generate the Web API, open the Service Desk client using the system administrator account. From the administrator console, choose **File** → **Generate Web API**. A dialog box for generating the Web API (see Figure 23) appears. Set the Web API destination directory (for example, `C:\Temp\WebAPI`), then click **Start Generate**. If generation succeeds, users will find all the Web API sources in the destination directory.

Figure 23



Configure Build Parameters

Two additional files named `build.xml` and `build.properties`, which are used to build new Web API, are shipped together with this special feature. They reside in the `doc\itsm008292` subdirectory of the unpacked Service Pack. Copy them to the Web API destination directory: `C:\Temp\WebAPI`. Open `build.properties`, which contains the following contents:

```
# Web API build property file. Hewlett-Packard, Dec 2005

###########################################################
# NOTE: Please change the values of below properties to #
# match your specified environment. #
###########################################################

# web api sources output directory
sd_webapi_output_dir=C:/Temp/WebAPI
# The official web api jar file
sd_official_webapi=C:/Temp/sp18/api/web-api.jar
# The official web api javadoc zip file
sd_official_webapi_javadoc=C:/Temp/sp18/doc/Web API
javadoc/web_api_javadoc.zip
```

Users need to change the values of all the properties defined in this file to match their specific environment. These properties are described in below table.

Table 3

Property Key	Description
sd_webapi_output_dir	Directory containing the Web API sources.
sd_official_webapi	Qualified file name of official web api jar file.
sd_official_webapi_javadoc	Qualified file name of official Web API javadoc zip file.

In this sample file, the Service Pack is extracted to `C:/Temp/sp18`.

Compile the Web API and Javadoc

Open a DOS command console, change the current directory to the Web API destination directory (in this example, `C:\Temp\WebAPI`). Users can configure these system variables manually. If they have not done this (see *Set Windows Environment Variables*), type the following commands to set these variables:

```
SET JAVA_HOME=C:\j2sdk1.4.2_09
SET ANT_HOME=C:\dev\ant
SET PATH= %JAVA_HOME%\bin;%ANT_HOME%\bin;%PATH%
```

To show the help text of the Web API build tasks, type `ant help`.

```
usage: ANT <target>
target:
o [clean ] : delete all files generated when run web api build task
o [config] : initiate the target properties
o [build-webapi] : build web api
o [webapidoc] : build web api javadocs
o [build-all] : build all the targets
o [rebuild] : rebuild all targets

Before building web api, user must configure the properties defined in
build.properties. Those properties are:
    sd_webapi_output_dir: Web API source output directory
    sd_official_webapi : The official web api jar file
    sd_official_webapi_javadoc: The official web api javadoc file

After target [build-all] is successfully run, new web api jar file and
web api javadoc file can be found in C:/Temp/WebAPI.
```

To build Web API jar file and Javadoc, type `Ant`, and press **Enter**. After the build succeeds, the new `Web API jar` file will be found in the Web API destination directory. The newly built files are `web-api.jar` and `web_api_javadoc.zip`.

The new `web-api.jar` file can be used on multiple platforms (for example, Windows, HP-UX, or Solaris). Developers should replace the old jar file with the new one to access new custom fields using the Web API in their own web applications.

Known Errors and Limitations

The unlimited custom fields feature has the following limitations:

Please be noticed that the amount of customer fields in your environment could impact the performance.

This feature is not intended to be used in a multi-user environment because it modifies the database structure. It is highly recommended that no other client connects to the application server during the import. Also, it is also recommended to have only one application server instance running on the Service Desk database when you define or import new custom fields. Server needs to be restarted after all expected custom fields are created or imported.

When creating new custom fields, information like "You have to add a label with oid ... in the language with LCID 1033" will appear in server log. These information can be ignored. During creating new custom fields, two SQL files `CustomFieldDMLLog.sql` and `CustomFieldDDLLog.sql` will be generated in Service Desk server installation directory.

When a great number of new custom fields are defined for a single item category, the popup menu in Advanced Find for selecting Fields will get too many entries that it no longer fits on the screen. The actual limitation differs according to screen resolution. It also happens with other popup menu for selecting Fields, e.g. rule wizard condition dialog.

Custom fields can be created on master items only, which currently include the following:

- Service Call
- Problem

- Incident
- Change
- Work Order
- Project
- Configuration Item
- Service
- Service Level Agreement
- Maintenance Contract
- Organization
- Workgroup
- Person

Any item can have a maximum of 128 long-text custom fields, including the original long-text fields.

It is allowed to create new custom fields referring to existing code types but creating new code types is not supported.

Using ACES to import new Custom Fields will probably change the data mapping model, please refer to Section *Custom Fields Migration with ACES*.

The new view of Custom Fields (see Figure 15) shows custom fields per item category sorted by validation type. In the dialog to define a new Custom Field, you can select a validation type. In the dialog to edit an existing custom field (which existed before SP18, see Figure 20), the validation type is shown behind the name of the field between parentheses. However, in this dialog (which opens when you double click and existing custom field), the validation types are categorized differently compared to the view of custom fields: many of the new validation types are lumped together under the name of a more generic validation type. Consequently, you could see a different validation type between the brackets of the edit dialog than you see in the tree view. For example, the validation types "number (double)", "number (integer)" and "duration" are all shown as type "number" in the custom field edit dialog, and there are other validation types that are shown as "number" in the edit dialog as well.

Likewise, custom fields of validation type "Entity reference" will be shown in the custom field edit dialog with the Service Desk item that they refer to between brackets. This is also where the categorization differs between the new custom fields view and the old dialog to edit custom fields.

The newly created custom fields can be deactivated if it is no longer used. However, delete function of custom fields will not be supported until future notice.

It is possible that the newly generated web-api cannot be compiled. When this happens, you will see an error message similar to "getLbXXXX() is already defined in ..." This problem is caused by unexpected data structures, which can be introduced with database changes made for the implementation of enhancement requests. For example, such changes could cause the database to contain a localized text with a duplicate key, which is something the web-api generation process cannot deal with. To solve the problem, either remove redundant data from the database and generate the web-api again, or remove the duplicate methods from the generated web-api java files. Please refer to *ITSM008406* for more information.

Rule Enhancements

Introduction

Service Desk has provided the possibility to define one single Database Rule (or UI Rule) that triggers for more than one event (such as when item is created and when item is deleted for Database Rule). Hence, rules with same conditions and actions can be combined into one which is easier to use.

Existing rules can be used without explicit migration operations. Existing compound rule trigger will be translated to multiple selection of single triggers automatically.

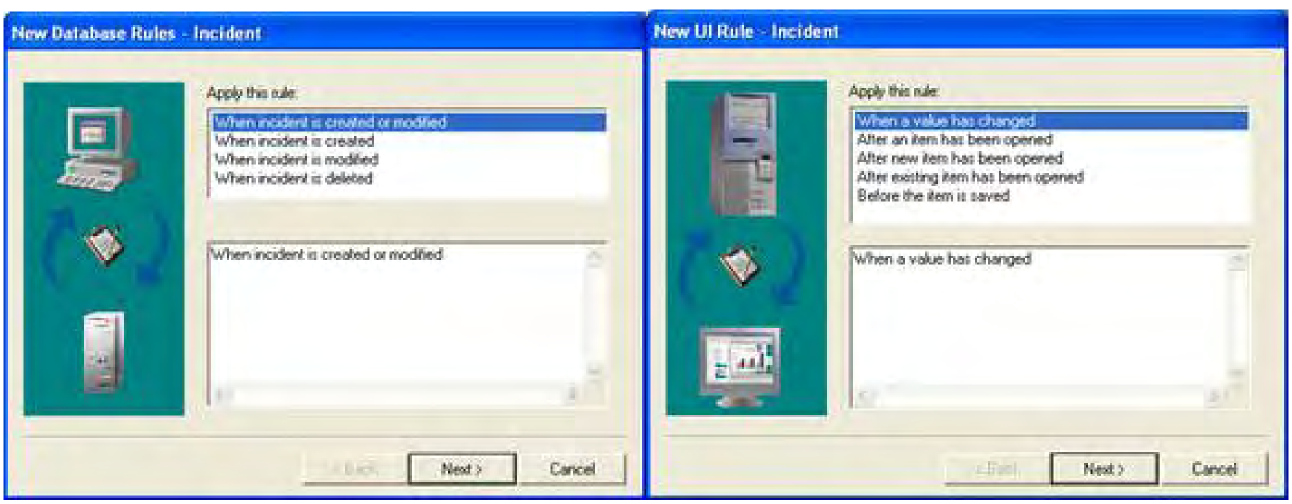
The validation of a Rule (e.g. a Database Rule when item is deleted cannot have update data action) will take all selected triggers into account.

Getting Started

New UI and Database Rule Start Conditions

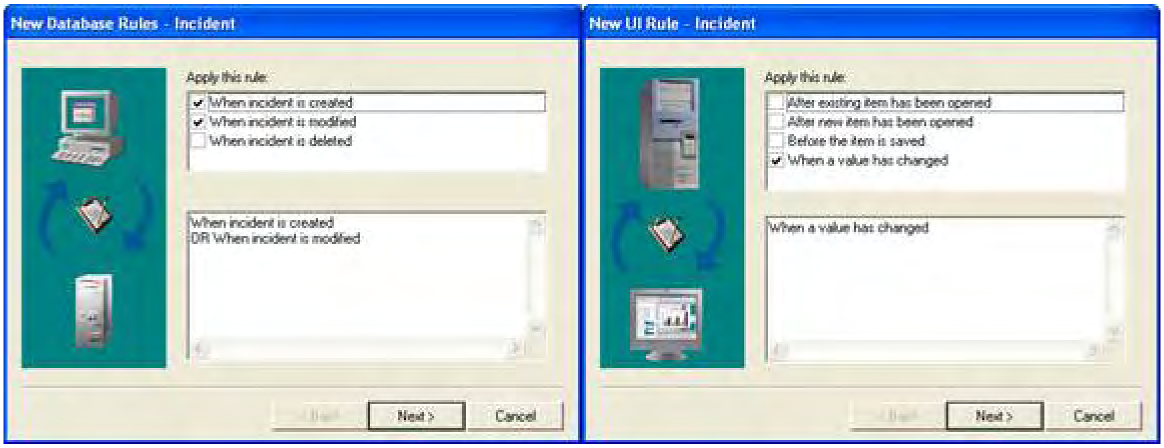
UI rules trigger when a certain event (such as when a form is opened or closed, or when the user enters data in a field) occurs in a form in the Service Desk client. Similarly, Database rules trigger on any event that changes an item in Service Desk (a new items is saved, an existing item is changed, or an item is deleted). Before Service Pack 18, UI rules and Database rules could only be triggered by a single event, as seen below:

Figure 24



Starting from Service Pack 18, it is possible to define multiple events that trigger a rule. This is shown in the following screenshots:

Figure 25



The possibility to define more than one event for the rule to respond to makes the rules easier to use. Before, actions that had to be undertaken as the result of multiple starting conditions (for example, a rule that fires when an items is opened and when it is updated) had to be implemented by multiple rules. Now, a single rule will suffice.

About Existing Rules

Existing rules can be used without explicit migration operations.

Existing compound rule trigger will be translated to multiple selection of single triggers automatically.

Table 4

Rule Type	Trigger Before Upgrade	Triggers After Upgrade
UI Rule	After an item has been opened	Multiple selection: After new item has been opened After existing item has been opened
Database Rule	When item is created or modified	Multiple selection: When item is created When item is modified

Validation of Rule Actions and Conditions

The actions available to a UI rule or a Database rule have always depended on the event that triggers the rule. For example, a Database rule that fires when an item is deleted cannot trigger an "Update data" action, because there is no data to update when an item is deleted. When multiple event types are chosen as trigger, only those actions are allowed that are available for each of the event types individually.

For example, the "Update data" action is not available for a Database rule that responds to event type "When item is deleted" but the "Command exec action" is available. When the event type "When item is created" is chosen as trigger of a Database rule, the "Update data" action as well as the "Command exec action" can be used both. Consequently, a trigger that responds to both "When item is deleted" as well as "When item is created" cannot invoke a

"Update data" action, because only one of the two event types supports it. However, it can use a "Command exec action", because it is available for both event types.

Furthermore, some event types can be fine tuned with conditions. Conditions specify that in addition of the triggering event type, the field values of the currently open item should meet certain specified criteria. Some event types cannot depend on conditions. When an event type that can be triggered with a condition is used in combination with an event type that cannot accept a condition, it affects the way the rule is executed. When the event occurs to which conditions do not apply, Service Desk executes the action associated with the rule, without evaluating the condition, even though the condition is part of the rule.

For example, when you define a Database rule that triggers only on the event "When an item is deleted", the Database rule wizard will not show a conditions page. When you define a Database rule that triggers on the event "When an item is modified", you will have the option to specify conditions based on values of the item. When you are editing a Database rule that triggers both on "When an item is deleted" and "When an item is modified", you will see the conditions page. However, when the rule is executed, the conditions are not evaluated if the triggering event was "When an item is deleted". The occurrence of the triggering condition alone is in this case enough to execute the actions associated with the Database rule.

For this particular example, a warning in the Database rule wizard will bring this to the attention of the administrator:

Figure 26

Another example is when you define a UI rule triggers on the event "When a value has changed" or "After new item has been opened", it is possible to enable "Evaluate this rule when this field has been changed" for a rule condition. When the rule is being evaluated, this option is only effective for event "When a value has changed" and does not work for event "After new item has been opened". This is consistent with original UI Rule behavior.

9 Service Pack 19

This chapter contains information previously provided with Service Pack 19.

UI Rule Enhancements

Introduction

A new system action has been introduced called "Commit" which allows the user to save current form as result of a UI rule. To fully support the new feature, actions of a UI rule now have an explicit execution order.

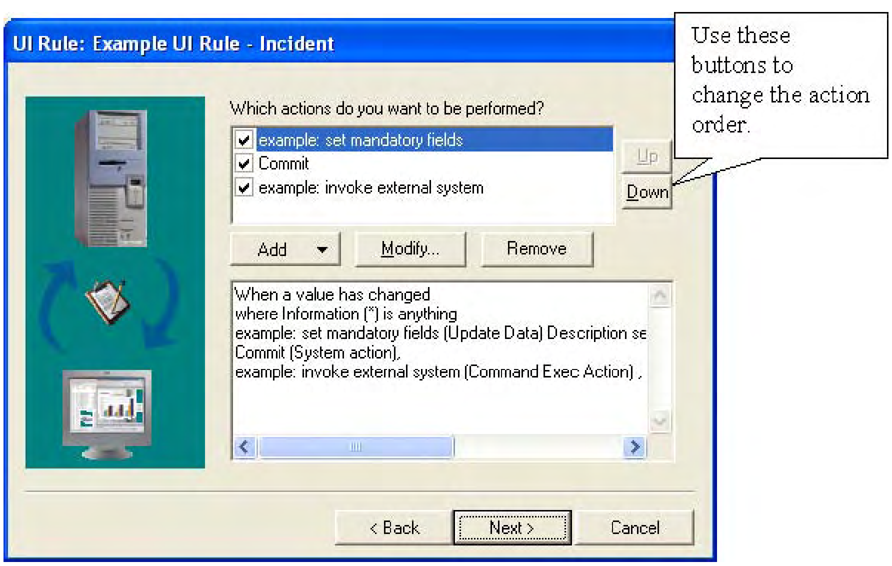
This document details these changes, their intended use and things to consider when using the solution.

Ordered UI rule actions

The actions configured to be the result of a UI rule now have an explicit order in which they are executed. Before that, the execution order was arbitrary. By giving actions an order, it is now possible to have one action respond to the outcome of another action.

The order in which UI rule actions are executed is the same as the order in which they are shown in UI rule wizard. You can change the order using new **Up** and **Down** buttons that have been introduced for that purpose. Following screenshot shows the page of the UI rule wizard where actions are defined with the new buttons:

Figure 1



Please note that "command execute actions" and "smart actions" may initiate processes that are not completed before initiating subsequent action. These actions run external commands which in turn may launch a new operating system process before returning control to Service Desk client.

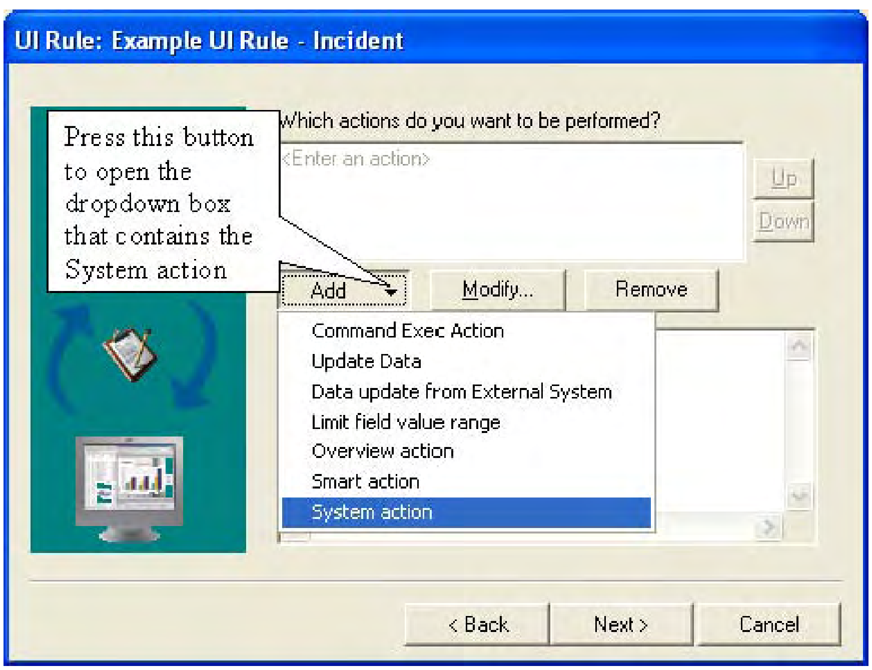
For example, if an action is to launch text editor notepad in Windows XP using a "command execute action", Service Desk will not wait until the user closes notepad. Instead, Service Desk waits until notepad is started and then continues to perform next action. If your business logic requires that a file is created through notepad after it has been started by UI rule, there is no way for Service Desk to know when this has actually happened. Therefore, implementation of that type of business logic through Service Desk is not supported. This has always been the case since introduction of UI rules themselves. The introduction of ordered UI rule actions does not change this.

New System Action: Commit

The new Commit system action allows administrators to save entity in currently open form as the result of a UI rule. In itself, executing Commit system action does the same as the Save button. What is new with the system action is that the open form can be saved not only when the Save button is pressed but also in response to other form events, such as selection of a specific value.

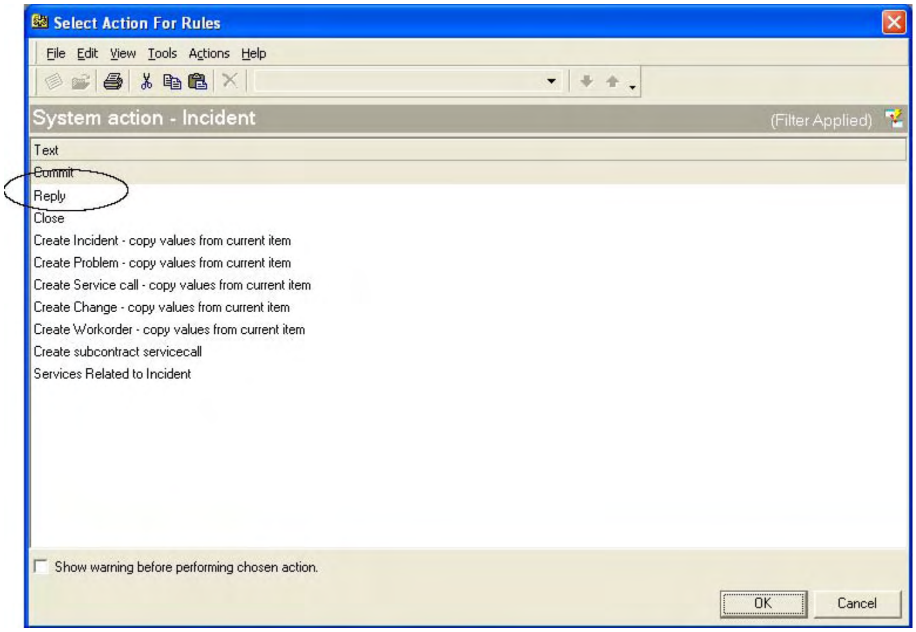
To use the Commit system action, edit a new or existing UI rule in Administrator Console. Define UI rule as described in the Service Desk 4.5 Administrator's Guide. At the page that defines the actions to perform for UI rule, press "Add" button and select "System action" from dropdown menu, as seen in the screenshot below:

Figure 2



Selecting System action opens another window, from which you can select the Commit system action:

Figure 3



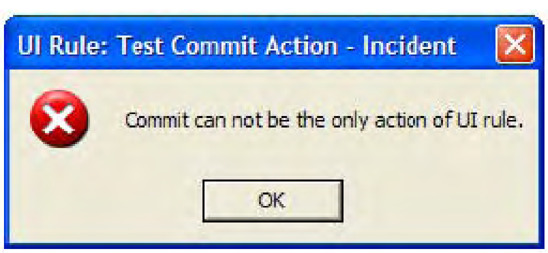
Please note that actual set of available system actions may differ from those seen in the screenshot depending on the item type of UI rule. This does not matter for Commit system action: it is available for all item types.

No further configuration of Commit system action is required. However, you must configure another action next to the Commit system action before you can continue. The Commit system action is designed to work in concert with another action. It is not allowed to use the Commit system action by itself. Typical scenarios in which the Commit system action is used include:

- An action triggers an external system that retrieves data from Service Desk using a callback.
- An action opens another form that needs to be able to reference the data in the current form.

If the administrator uses the Commit system action as the only active action performed by a rule, UI rule wizard display following error message:

Figure 4



When you add a second action to the UI rule, this error disappears.

Because the Commit system action is the same as pressing the Save button, Service Desk will not allow you to select this action on a rule that is triggered by the condition "Before the item is saved". This is because the effect of the Commit system action triggers rules that check on that condition, and therefore the rule would trigger itself. The result would be an infinite loop.

The following error message appears on the last page of UI rule wizard whenever administrator ignores this restriction: This error can only be resolved by selecting another type of triggering condition. In other words, the administrator must rethink the business logic that is implemented with the rule so it is triggered by another type of form event.

Figure 5

A UI rule is not allowed to invoke the Commit system action more than once. If you try to do this, following error message will appear:

Figure 6

The use of multiple Commit system actions in a single UI rule adds a level of complexity to the UI rules that is considered difficult to support. Therefore, administrators must rethink any UI rule that requires multiple Commit system actions.

Effect of Merging

Optimistic Locking and Merging

Service Desk uses optimistic locking to keep data consistent. This means that Service Desk checks whether anyone else has modified an item since it was opened by current user at the time that the current user tries to save that item. If it was modified in the meantime, Service Desk client will try to reconcile the modifications made by another with the current modifications.

As a result, Service Desk client will open a new form with merged result and a "merged result" dialog will popup to list the fields have been changed.

- If the field is changed by previous user only, earlier value will show in the new form.
- If the field is changed by current user only, current value will show in the new form.
- If the field is changed by both, earlier value will show in the new form and in the "merged result" dialog it will be marked as "merge impossible". If the field contains a set of items (e.g. historylines), it will also be marked as "merge impossible".

No matter whether the merge is completely successful or not, client will always popup the new form with merged result and waits for user's further action. User can either choose to save, not to save or edit and save. The merged result will not be saved automatically.

Effect of a Merge on the Commit System Action

Optimistic locking is said to be optimistic in that it is based on the assumption that in most cases, no modifications by other parties occur in the interval between opening a form and saving it. In most instances therefore, the Commit system action does not have to deal with the effect of optimistic locking, the triggering of a merge. Be advised however that when it does, administrator should be aware of the effect of a merge.

Whenever a Commit system action requests a merge, the UI rule execution aborts. All actions subsequent to the Commit system action are not performed.

The nature of merge is essentially

- Open a new form
- Load the data from database
- Compare what is changed by previous user and current user. If system decides to use the data inputted by current user, system will set the field to new value.

Consequently, UI Rule maybe triggered again in the new form.

For example, if a UI rule is "When status is changed and equals closed, save the form and popup a notification message".

Case 1

1. User A and User B opened one item at the same time.
2. User A changed description field, save.
3. User B changed status and set to closed. The UI Rule triggered to save the form and merge happened. UI Rule execution will abort.
4. Because of the merge, a new form is opened with initial data that User A saved. System decides to adopt the "status" value User B inputted. Therefore, "status" in new form is set to "closed" and UI Rule triggered again. This time, the save is successful and UI Rule execution is completed.

Case 2

1. User A and User B opened one item at the same time.
2. User A changed status field to accepted, save.
3. User B changed status and set to closed. The UI Rule triggered to save the form and merge happened. UI Rule execution will abort.
4. Because of the merge, a new form is opened with initial data that User A saved. System found merge is impossible because the "status" field is changed by both User A and User B. Therefore, system does not change the value in new form and UI Rule will not be triggered again.

It is suggested not to take advantage of the merging mechanism as the triggering of UI Rule may not behave as Administrator expected.

UI Rule Interference

It has always been the case in Service Desk that administrators should be mindful of the effects of overlapping rule conditions. With the Commit system action, extra caution is advised in this respect.

A form event may trigger more than one rule directly or indirectly. It may trigger more than one rule directly because the conditions for the rules overlap. Indirect triggering can happen when one UI rule updates a field that is checked by another UI rule, or when the Commit system action triggers rules that act on the condition "Before the item is saved". When UI rules are set up, the administrator must be aware that such a cascade of updates is possible.

The setup of UI rules allows the definition of infinite loops. For example, if one rule fills field B whenever field A is filled and another fills field A whenever field B is filled, these two rules can potentially stop a Service Desk client from responding. In practice, the scenario described above finds an exit after a number of iterations, but this is due to implementation details. There is no mechanism safe-guarding the administrator against defining an infinite loop with UI rules. It is considered the administrator's responsibility to prevent this type of rule interference from occurring.

The Commit system action introduces the possibility of creating such a loop. An infinite loop would occur when rules that respond to "before the item is saved" update fields that trigger the rule that includes the Commit system action. Therefore, the administrator should review all the UI rules based on the "before the item is saved" condition when defining a new UI rule that includes the Commit system action. To enable the audit, go to **Administrator Console** → **Security** → **Audit** → **Audit Rules** → **Service Call** and check **E-mails**.

Complex Filter Condition

Introduction

Service Desk now allows user to define complex filters using "OR" and "NOT" compounds in advanced find and view filter.

To support this feature, GUI of advanced tab page in filter is changed to tree style which is similar to Rule Condition.

What this document covers

- GUI change of advanced pane in filter definition
- How to work with "OR" and "NOT" compounds
- Backward compatibility (automatic loading of filters defined in previous Service Desk versions)
- Role based control of using this feature

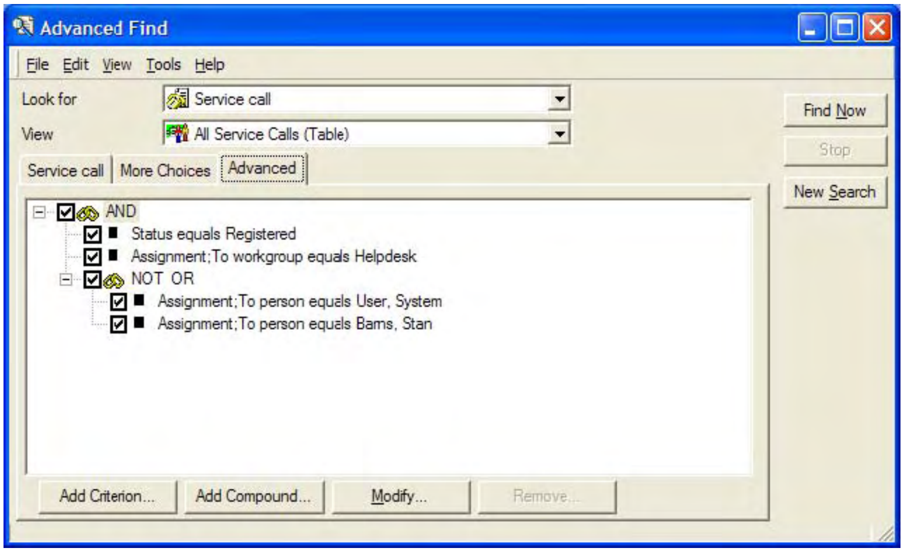
The target audience for this document is Service Desk administrators as well as normal client users.

Defining Filter

Tree Style Display of Filter Conditions

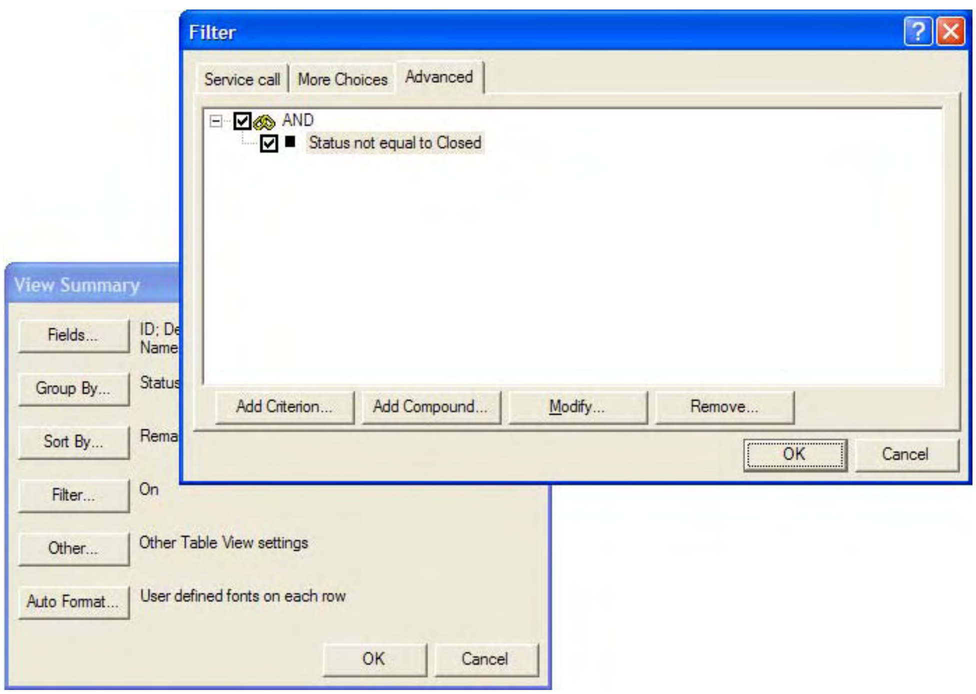
Login as system administrator (by default only System Administrator will have full access of the feature after upgrade, please refer to *Restricting the Usage of Compound*), open Advanced Find form, switch to Advanced tab, now display of filter conditions is using a tree style which is similar to Rule Condition definition.

Figure 7



The Advanced pane in filter definition of a view (including system view and user customized view) is also changed in the same way.

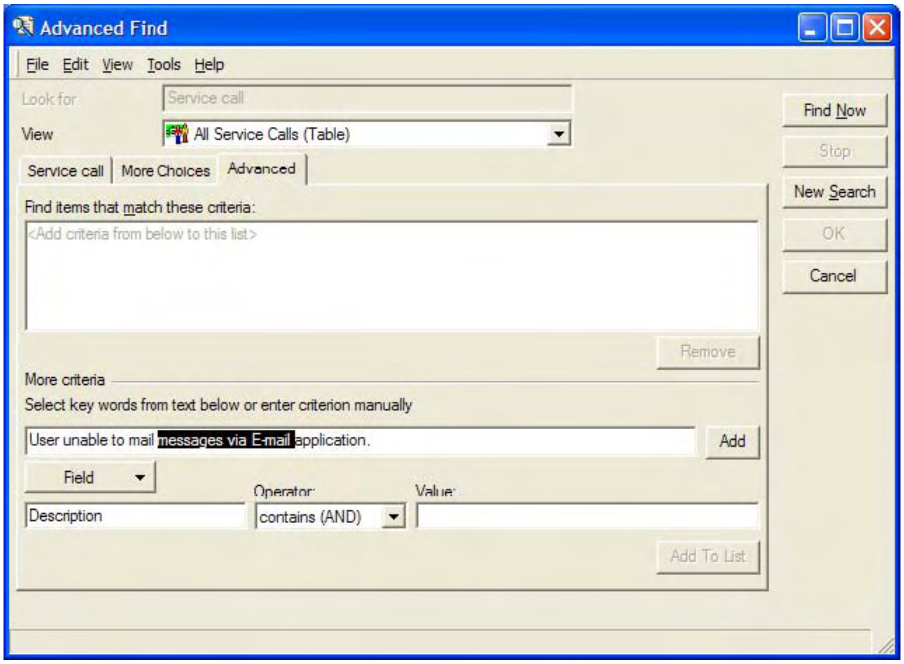
Figure 8



Unless otherwise stated, information in following chapters can be applied to Advanced Find as well as View Filter.

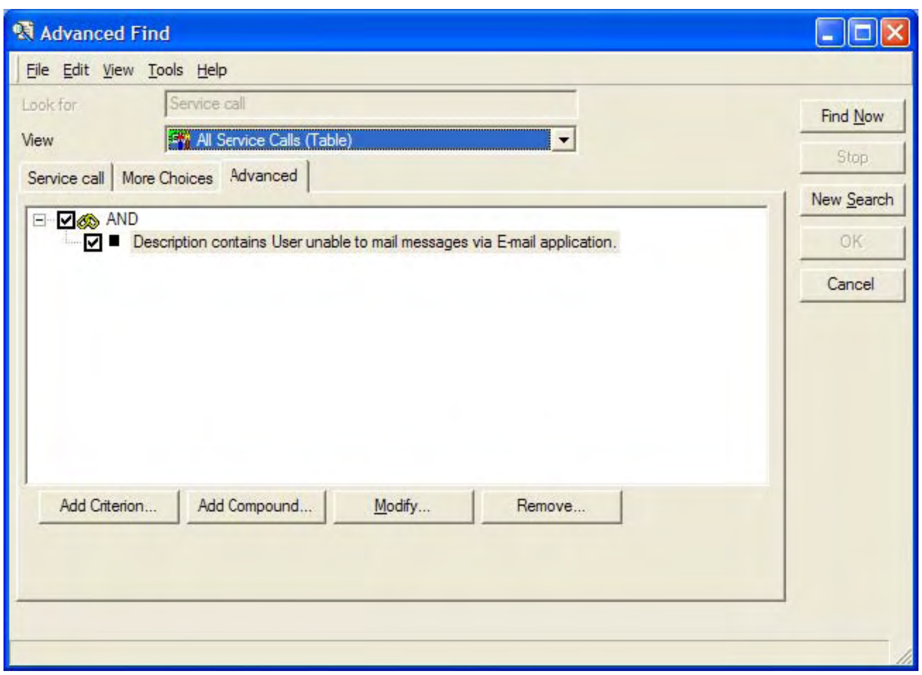
Previously, when editing a Service Desk item in a form, user can locate in one of the fields and open advanced find. The field and its data will automatically populated to advanced pane below the criterion list and wait for user's further operation.

Figure 9



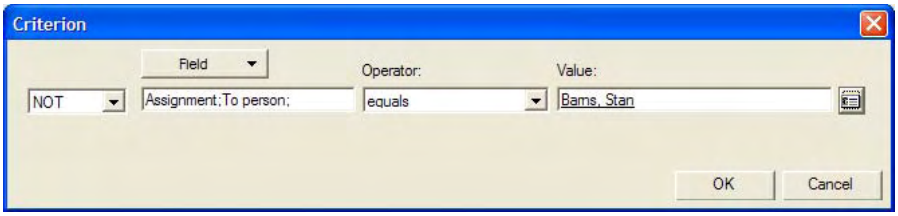
Now with introduction of tree style display, criterion will be added to the tree automatically. User can choose to edit the criterion after that.

Figure 10



Work with Criterion

Figure 11



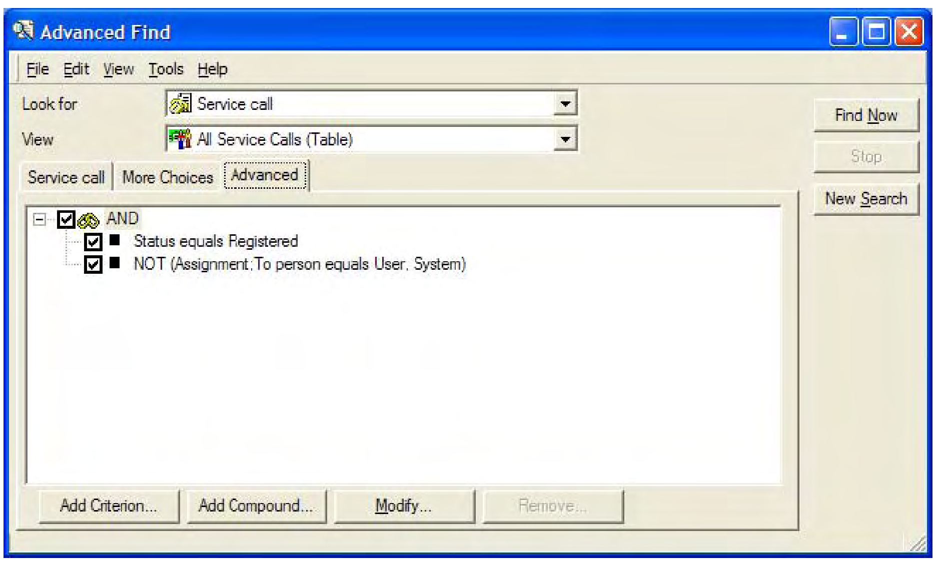
To add a criterion, click the **Add Criterion•** button and a dialog will popup. This criterion can also be negated by applying NOT prefix.

To edit a criterion, locate the criterion in condition tree and click the **Modify** button, the same dialog will popup. Or you can simply double click criterion in condition tree.

To remove a criterion, locate the criterion in the condition tree and click **Remove**.

By default, the relation between the criteria is "AND".

Figure 12



Work with Compounds

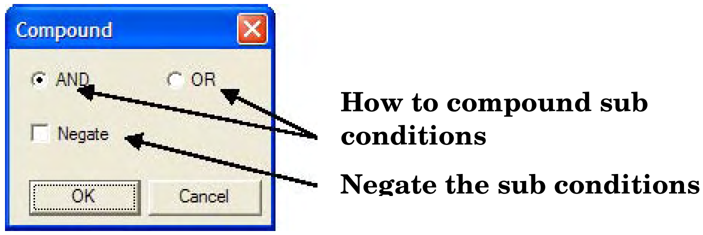
Concepts of Compound

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

There are two ways to combine sub conditions:

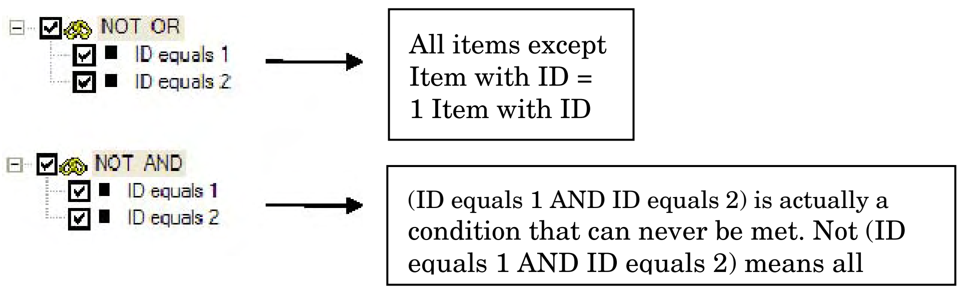
- AND means the condition is true when all sub conditions are true
- OR means the condition is true when at least one sub condition is true

Figure 13



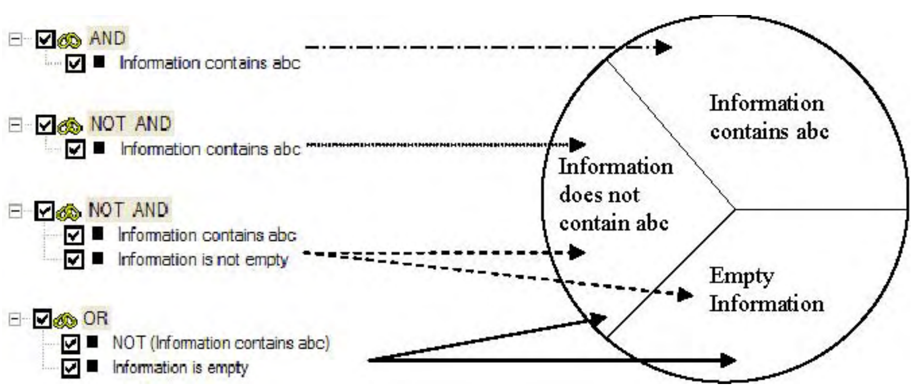
A Negate operator can be used with "AND" / "OR" to negate the whole sub conditions. For example

Figure 14



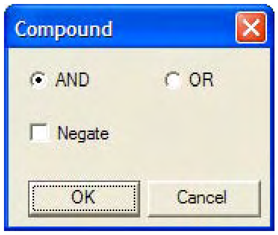
➤ Some fields can contain empty (null) values. A negated condition will not include the empty values.

Figure 15



Manipulate Compounds

Figure 16



To add a compound, click the **Add Compound**•button and a dialog will pop up. A compound operator will be added which allows adding sub conditions. To edit criterion/compound under specific compound operator, user should locate target compound operator in condition tree first before clicking the Add Criterion or Add Compound button.

To edit a compound, locate compound operator in condition tree and click the **Modify** button, the same dialog will pop up. The relation between the criteria will be changed.

By default an "AND" operator will be placed at the top of the condition tree and it can also be modified.

To remove a compound, locate compound operator in condition tree and click the **Remove** button. The compound operator will be removed together with its sub conditions.

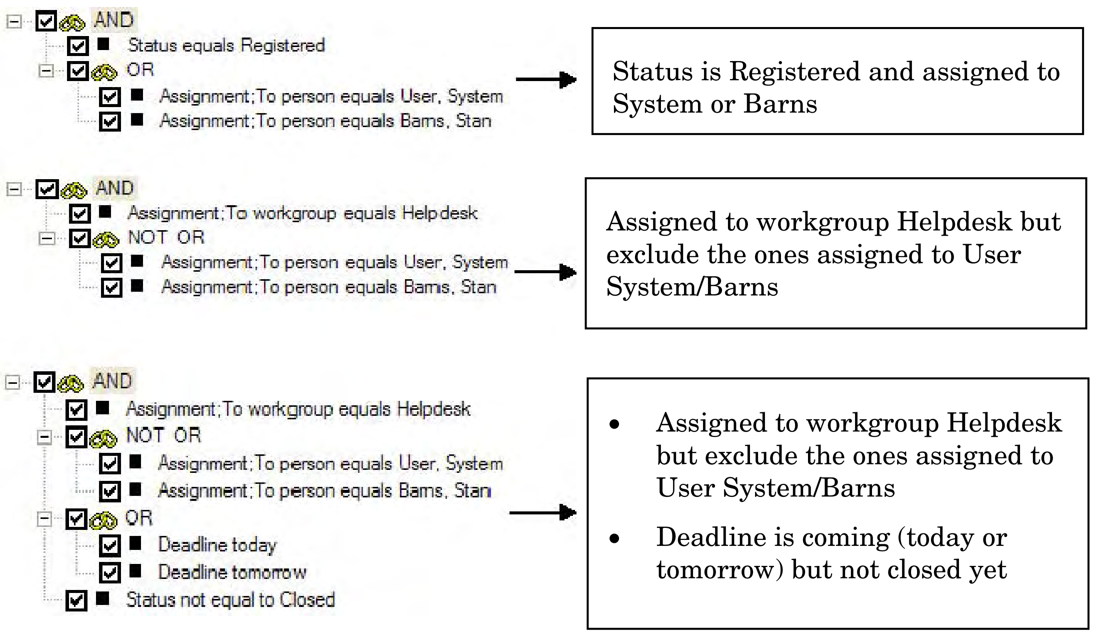
Even the top node of the condition tree can be changed to "OR", the relation between the three tab pages which consists the filter condition is still "AND". For example, user has filled "abc" in description field of standard pane and created a compound "(ID < 5) OR (Information Contains 123)", it will be finally explained as "(Description starts with abc) AND ((ID < 5) OR (Information Contains 123)).

Examples

Below are some examples showing the usage of compounds.

These examples are for demo purpose only. It is not recommended to over use "OR" and "NOT" in filter condition. Improper usage of compounds may lead to long database query which impacts the database performance. Please refer to *Restricting the Usage of Compound* for more information.

Figure 17

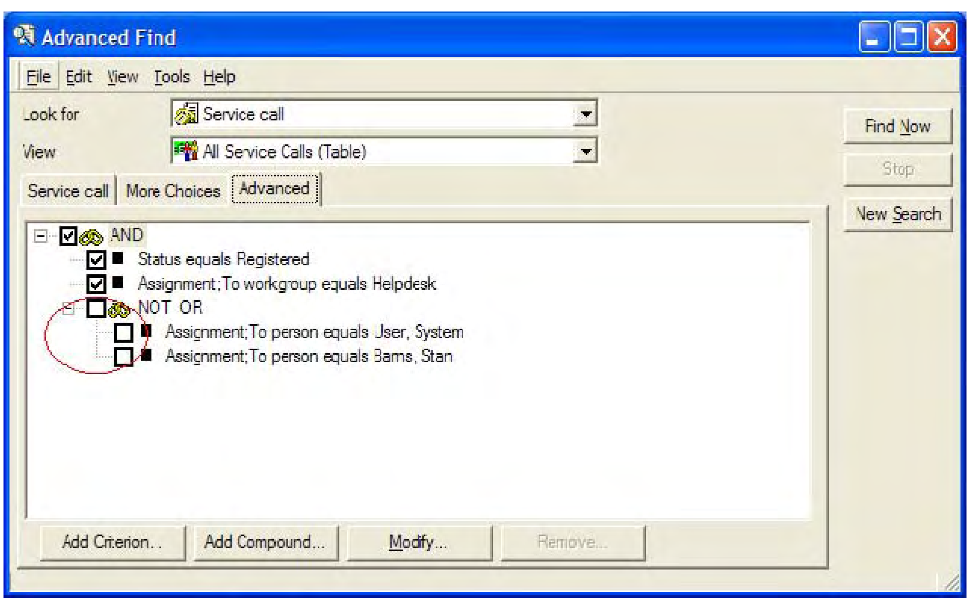


Enable/Disable Condition

User now is allowed to disable the criterion without removal of the criterion. User can click the checkbox in condition tree to disable or enable the criterion.

User can also disable the whole compound by uncheck the checkbox before compound operator, the whole sub condition will be disabled.

Figure 18



Drag Drop Support

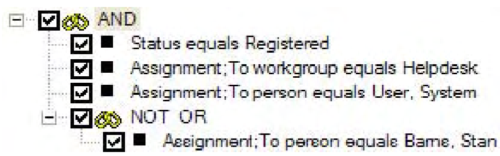
Drag and drop operation is supported in condition tree to adjust the criterion. User can drag a single criterion or a compound operator (with its sub conditions) to another compound operator.

- Click the source criterion or compound and hold the mouse button
- Move to the target compound operator
- Release the mouse button.

For example:

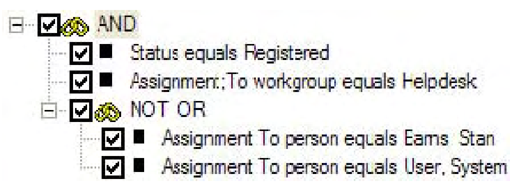
Add a compound

Figure 19



User can drag the criterion "Assignment; To person equals User, System" to compound "NOT OR".

Figure 20



Backward Compatibility

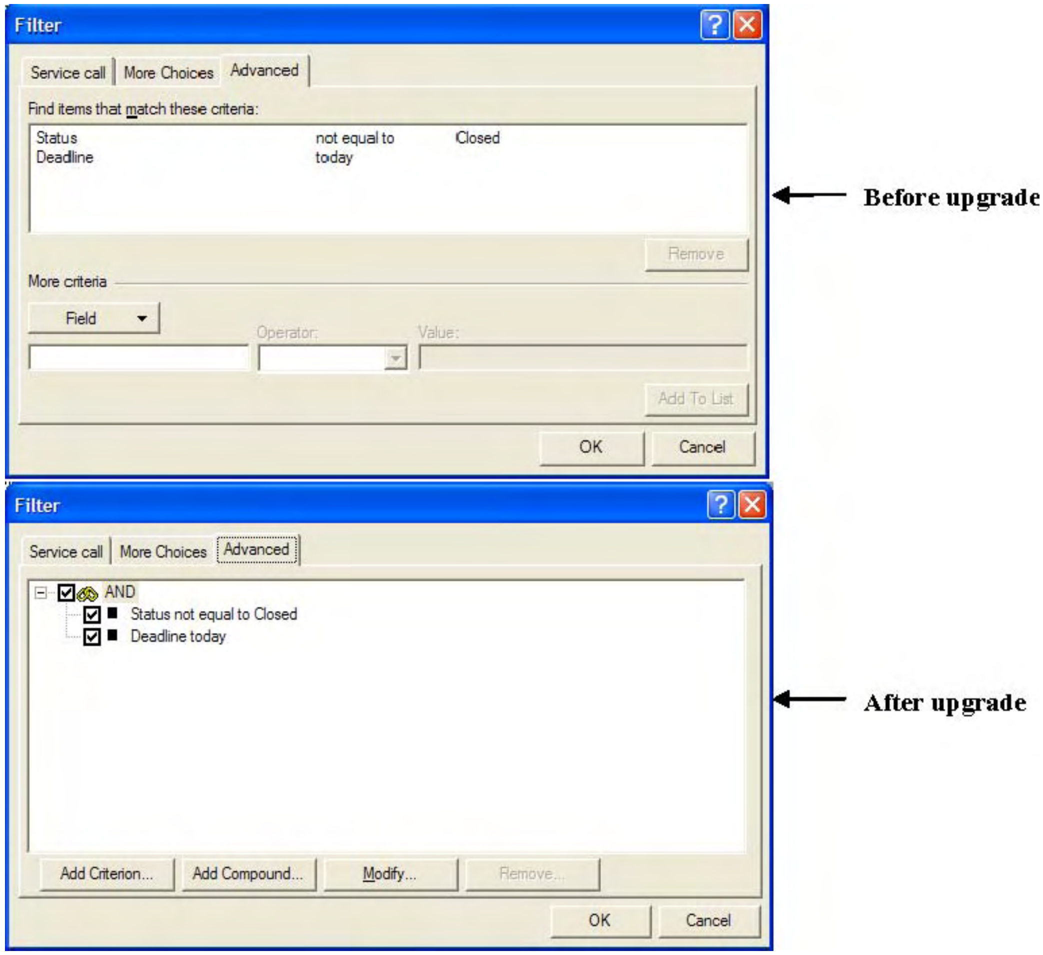
Filter Condition in View Definitions

All filter condition in existing view definition (including system view and user customized view) will be converted automatically.

The criteria in **standard pane** and **more choice pane** will be remained exactly same.

The criteria list in advanced pane will be converted to condition tree by adding an "AND" node at the top of the criteria.

Figure 21



In conclusion, the semantic of the criterion list is kept intact, and consequently search result is exactly the same.

Saved Search Condition in Advanced Find

It is also supported to load saved search condition in advanced find which is created in previous Service Desk versions. The conversion of criteria will work in the same way as view filter conversion.

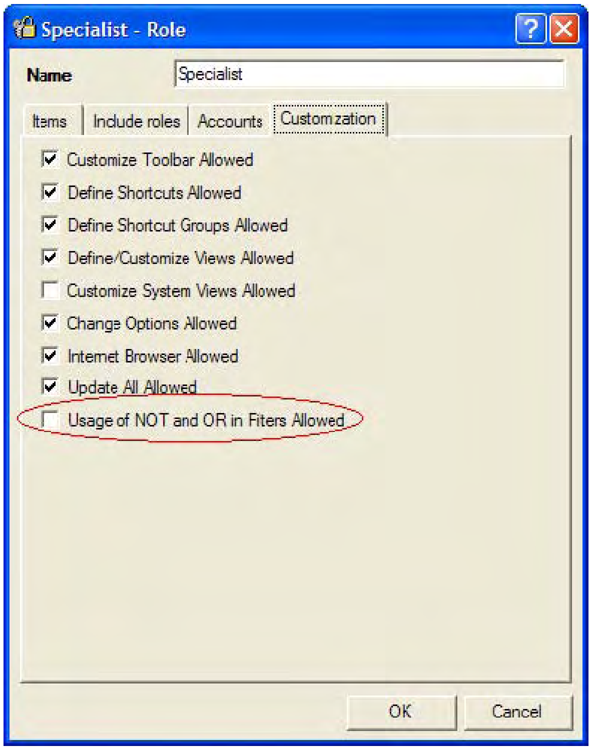
Restricting the Usage of Compound

Defining the Privilege

Improper usage of compounds may lead to long database query which impacts the database performance.

To limit the impact, a new option has been added to Role. Administrator can modify it in Role dialog, Customization tab page.

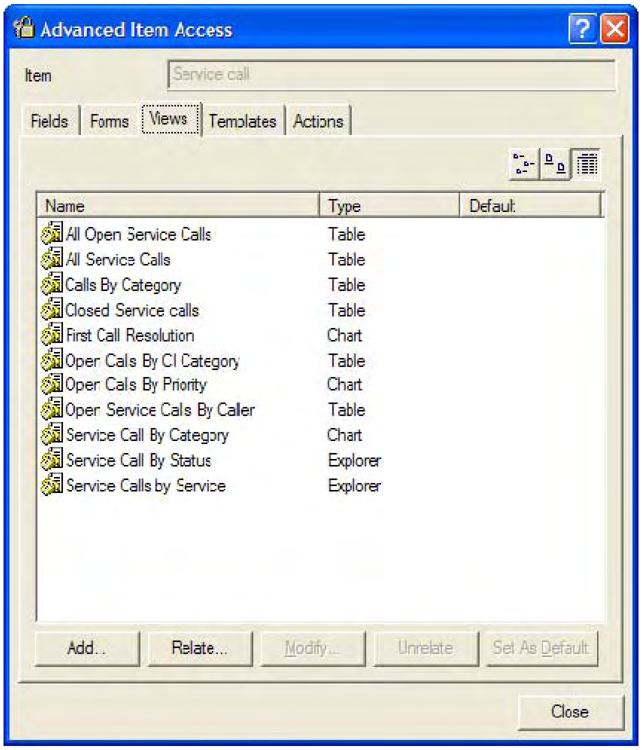
Figure 22



The intention of this setting is to control the privilege of **defining a filter with "Negate"/"OR" compound**. If a view filter contains "Negate"/"OR" compound, user can always use this view as long as he is authorized to use it.

To limit the access of certain view, Administrator should use existing Service Desk function as shown below.

Figure 23



After service pack upgrade, only System Administrator Role has this option enabled by default. For other Roles, Administrator should enable them manually.

Applying the Role Setting

Roles with "Usage of NOT and OR in Filters Allowed" Enabled

Account with roles which "Usage of NOT and OR in Filters Allowed" enabled has no restriction on defining filters with the compounds as well as applying the filters.

Roles with "Usage of NOT and OR in Filters Allowed" Disabled

Account with roles which "Usage of NOT and OR in Filters Allowed" disabled has some restrictions on defining filters but no additional restriction in using the filters to search.

- Use of Views / Load Saved Search Condition

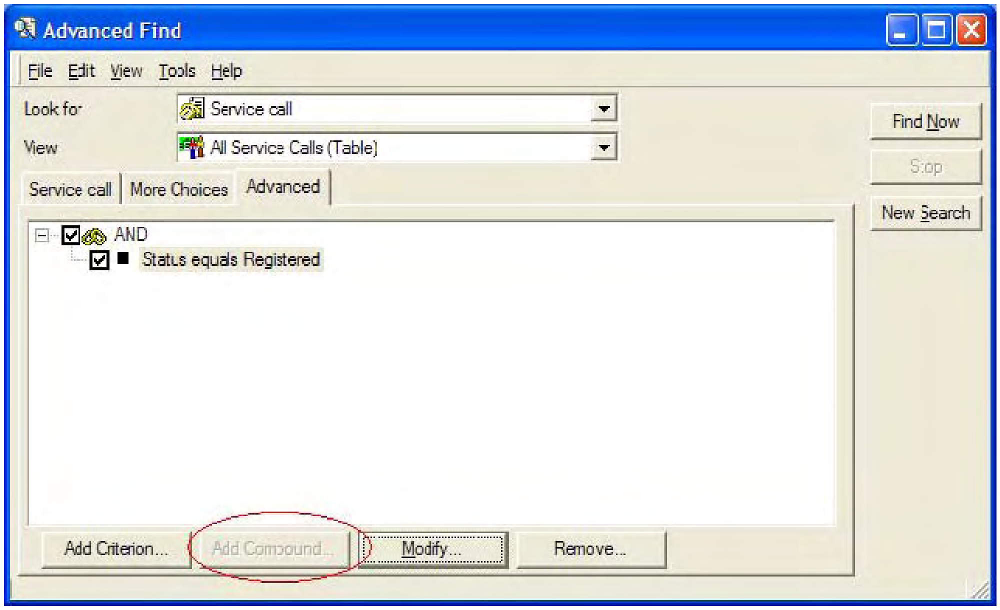
  There is no additional restriction in using the existing view to search or load saved search condition to advanced find even existing filter contains "Negate"/"OR". The only restriction is whether user has privilege to use that view or not.

- Defining New Filters

  When user defines filter condition in advanced find or view filter (if he is authorized to define/customize view), he is free to add criterion but he is not allowed to add compounds.

  As a result, there will not be any "OR"/"Negate" in the condition tree.

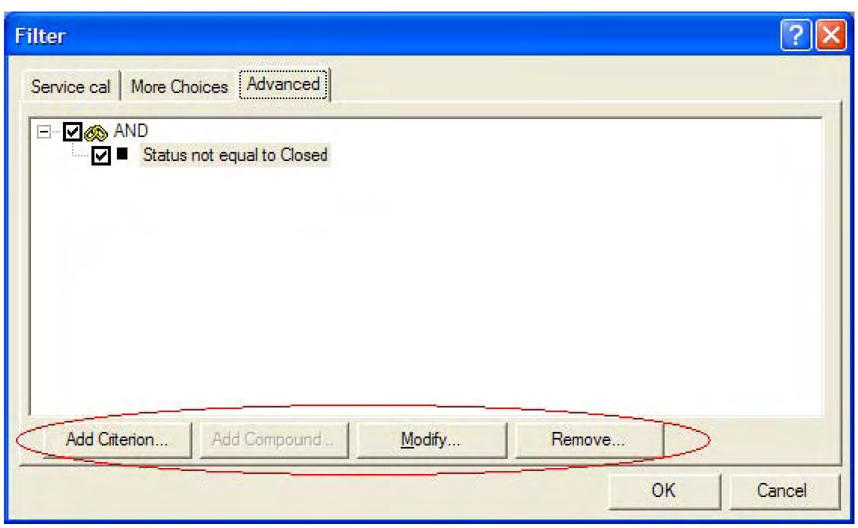
Figure 24



- Modify Existing Filters

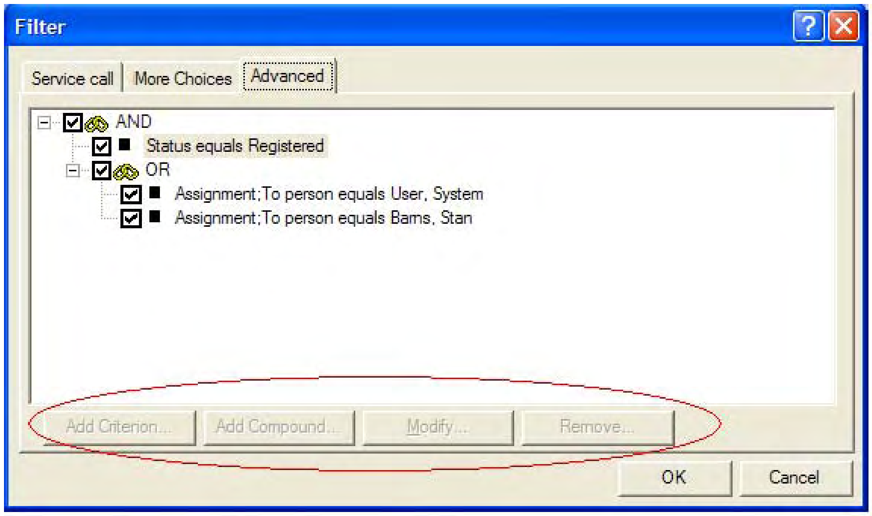
  When modifying existing filters (either in loaded search condition in advanced find or in view filters), if the filter does not contain "OR"/"Negate" compound in condition tree, user is still allowed to modify the filter by adding or removing the criteria but he cannot add compounds.

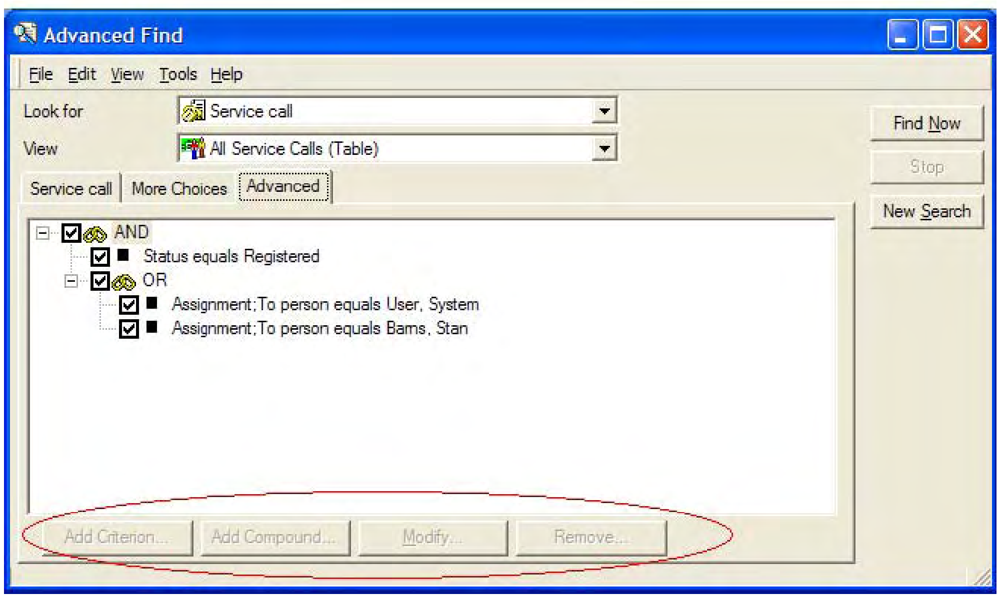
Figure 25



When existing filter contains "OR"/"Negate" compound, all edit functions of the condition tree will be disabled.

Figure 26





Email Enhancements

Introduction

This guide introduces the key concepts of following E-mail enhancements:

- Allowing user to send emails in client to handle Service Calls. By this feature, if a service call is created or updated by inbound email, user will be able to reply email with Service Desk client.
- All email histories related to Service Call can be stored and viewed.
- Some other E-mail function updates.

The target audience for this document is Service Desk administrators as well as normal client users.

E-mail Item

For the full support of E-mailing within Service Desk, a new Service Desk item called 'E-mail' has been added to Service Desk.

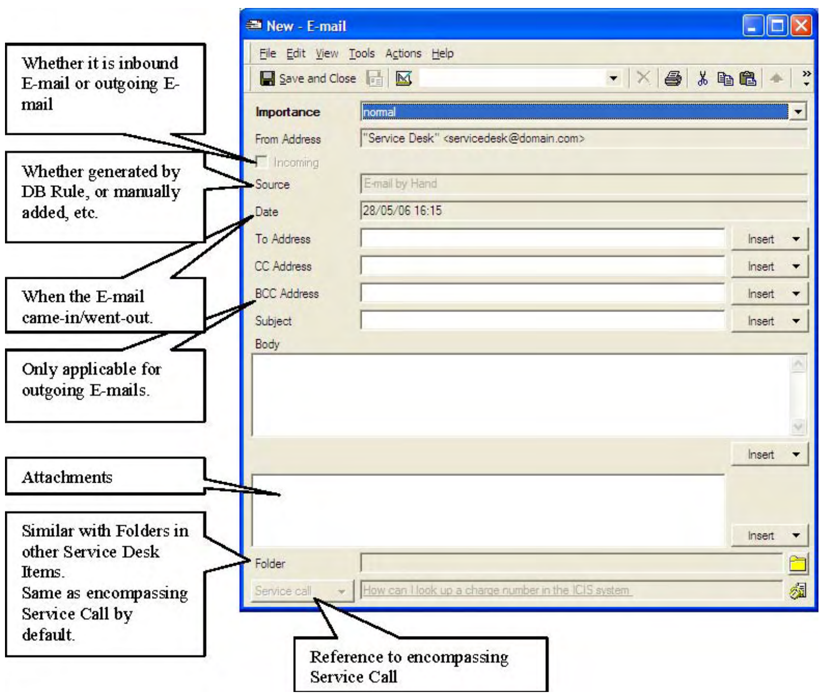
'E-mail' item works closely with Service Call item. A Service Call item can have multiple 'E-mail' items related and every 'E-mail' item must have an encompassing Service Call.

'E-mail' item can be used for both incoming and outgoing E-mails.

Fields in E-mail Item

This E-mail Item is mainly a snapshot of regular E-mail in Service Desk.

Figure 27

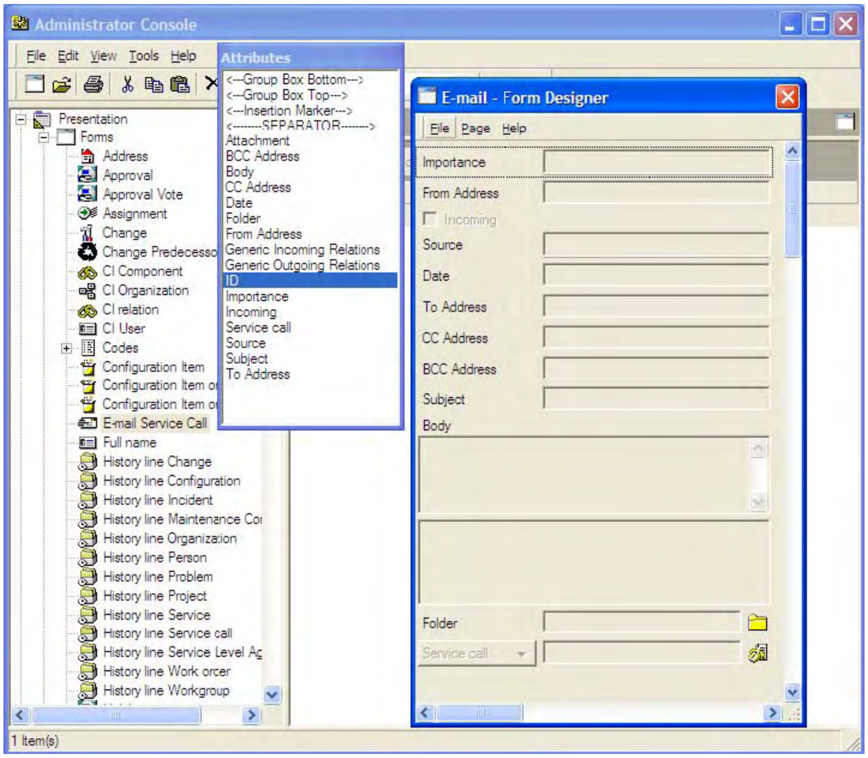


E-mail Form

A default E-mail Form has been added to Service Desk. Administrator can adjust it from **Administrator Console** → **Presentation** → **Forms** → **E-mail Service Call**.

For example, Administrator can add the field "ID" to form. ID is similar to functional IDs in other Service Desk Items which can be used to easily locate a particular Service Desk E-mail.

Figure 28



Attachment Support

Attachments for those Service Desk E-mails are stored in a way similar as the Attachment for the encompassing Service Call. However, they are stored separately.

By this, the exact history of all received E-mails can be maintained, even when Attachments from the encompassing Service Call will be removed from it.

It implies that certain Attachment might be stored twice; once in the encompassing Service Call and one in the received E-mail. This is an important notion of history preservation.

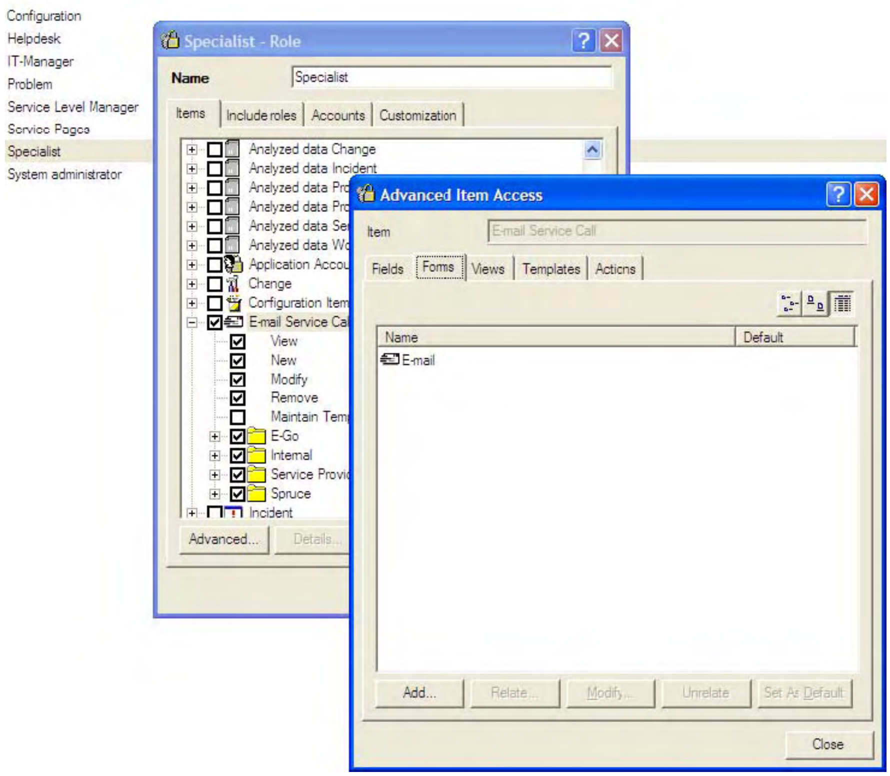
Whenever Attachments are not to be stored in a created Service Call (**Administrator Console** → **System Panel** → **E-mail** → **E-mail commands**, switched off for the 'New' command), they will still be stored within the E-mail.

E-mail Item is Authorizable

Access to E-mail can be set independently with the access to other Items. It means even a user has privilege to manipulate Service Call, he may still not have the privilege to deal with E-mail Service Call.

By default, only Administrator Role has full access to E-mail Service Call. To grant the privilege to other Role, Administrator should manually turn it on. And Administrator should also give the form / view / template access to that Role.

Figure 29



Supported Function for E-mail Item

Following functions are supported for E-mail Item

- Search in Advanced Find
- Database Rule
- UI Rule
- View
- Form
- Template
- Copy Fields
- Custom Fields

No Regular E-mail Client

E-mail Items are only Service Desk Snap-shots of all the (relevant) data found in the real (received and sent) E-mails. Whenever received or sent-out, a copy of data is made and subsequently stored in Service Desk database.

So they basically just hold the data but no heavy functional support will be provided on it.

For example:

- No Address books (or equivalent)
- No support of HTML or Rich Text format

It has not been the intention to implement a full-fledged E-mail Client within Service Desk application; this is only aimed at the proper storage, retrieval and processing of sent-in and out data is already possible within Service Desk.

E-mail Items in Service Calls (History Preservation)

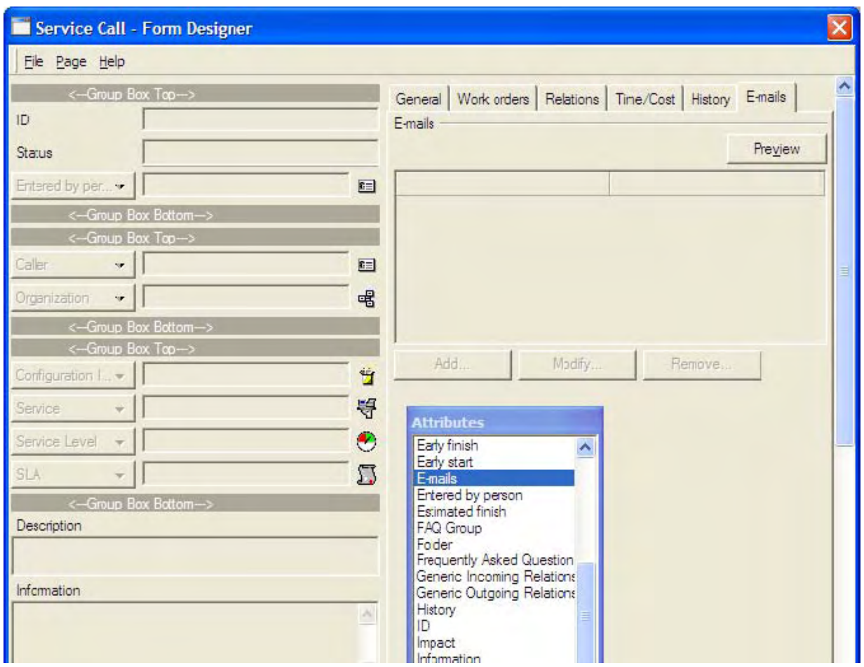
These 'E-mail representations' within Service Desk are maintained only within Service Desk items that are 'E-mail manageable', that is, Service Calls.

The whole E-mail history of related Service Call can be preserved and viewed.

Add E-mail Item to Service Call Form

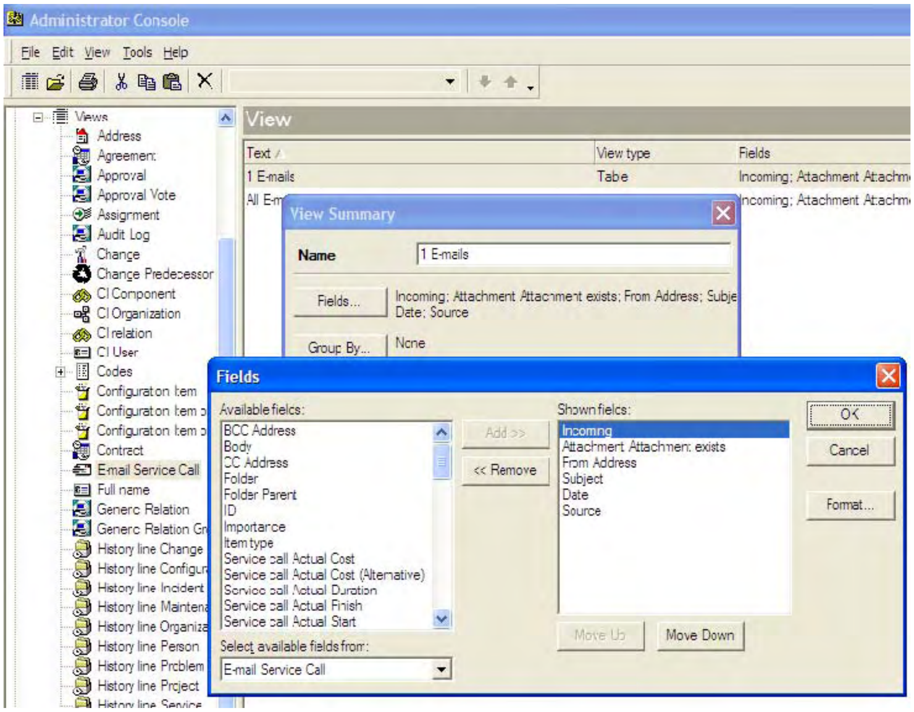
A service call can have multiple E-mail Items which represents E-mail history related to this service call. To view the E-mail items, Administrator should add the "E-mails" field to Service Call form.

Figure 30



The view that displays E-mail Items in Service Call form can be modified in **Administrator Console** → **Presentation** → **Views** → **E-mail Service Call**.

Figure 31



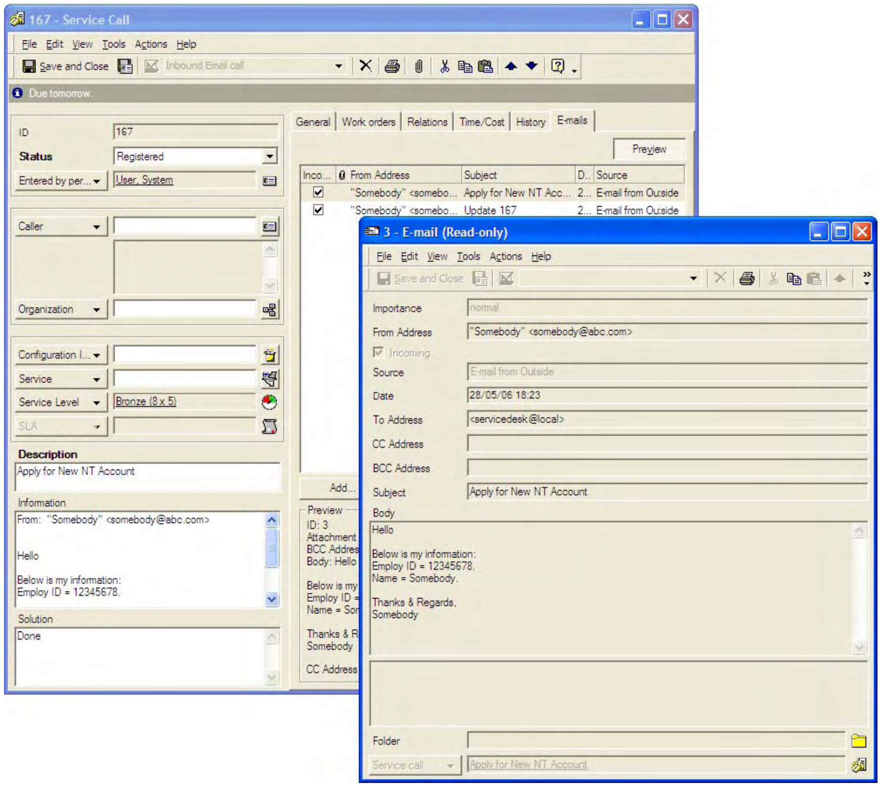
To be able to view the E-mail Item in service call and check its detailed information, user should be granted with proper privileges

- E-mails field of Service Call Item should not be hidden.
- Have access to E-mail Service Call item.
- Have access to at least one view of E-mail Service Call Item.
- Have access to at least one form of E-mail Service Call Item.

See also *Email Item is Authorizable*.

After above steps have been configured properly, user can view E-mail history of Service Call. If the user gets enough privilege, he will also be able to create or delete E-mails in Service Call (see the following sections for more information).

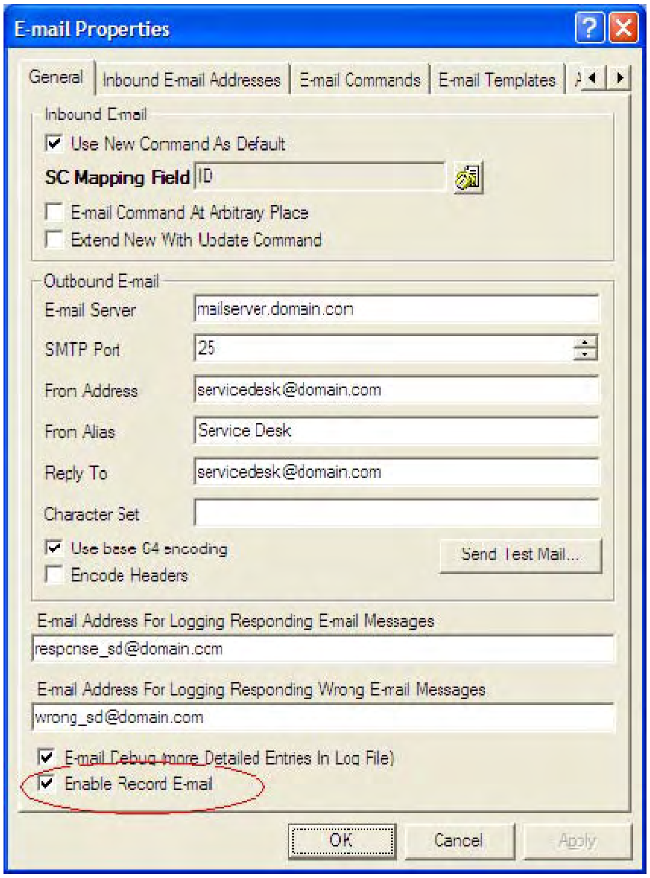
Figure 32



History Prevention

An option "Enable Record Email" has been added to Email setting. It controls behavior of logging Service Call Emails histories to Email Service Call. This option is disabled by default and Administrator can enable it from **Administrator Console** → **System Panel** → **Email**.

Figure 33



E-mails Are a Part of Service Call

E-mail Service Calls are full parts of a Service Call. It means

- E-mail Service calls can only be created within a Service Call
- When encompassing Service Call is deleted, all related Email Service Calls will be deleted.

E-mailing

Service Desk has already provided several possibilities to send outgoing E-mails:

- Database Rules
- Automatically generated Assignment/Approval Notification E-mails
- Smart Action 'New Mail Message'
- E-mail responses that are being generated as a response to the E-mail commands for Service Calls

Besides these, two new styles have been introduced:

- Send E-mail via three additional System Actions
- Manually send E-mail which is related to a Service Call

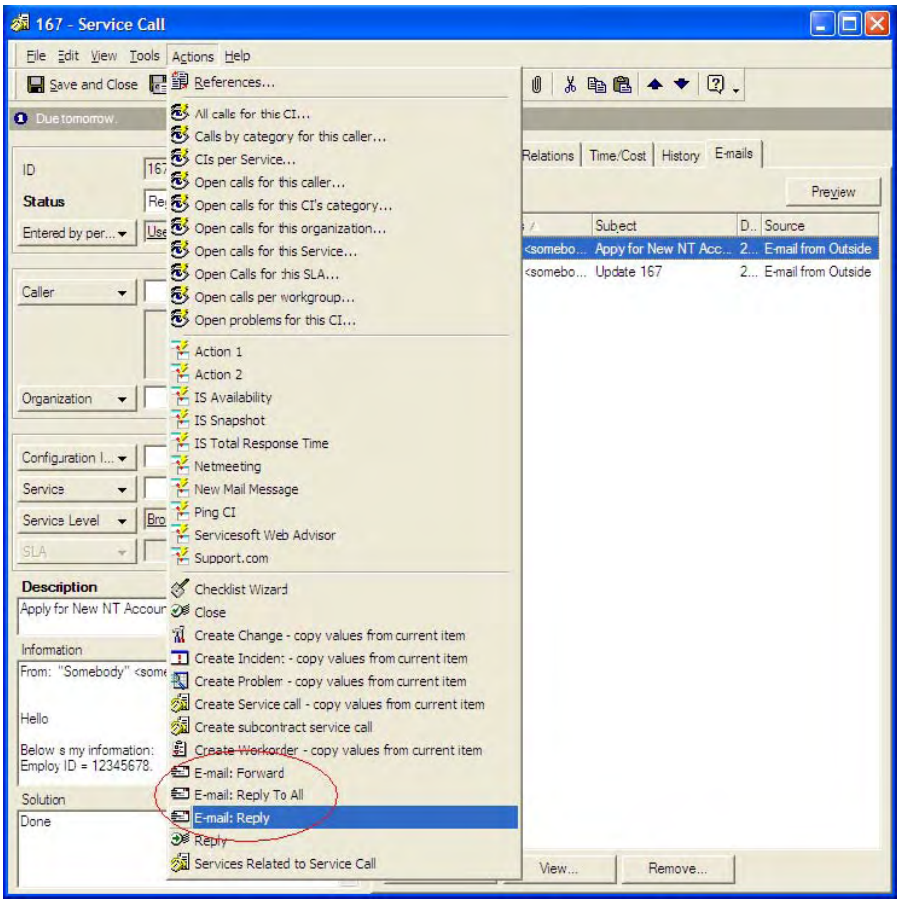
E-mailing by Service Call Action

Three System Actions have been added to Service Call which works with new Service Desk E-mail item.

To enable this feature, Administrator should enable **Administrator Console** → **System Panel** → **Email** → **Enable Record Email**.

When user opens a Service Call and select an E-mail Service Call item, clicks the "Action" menu, the new System Actions will be listed.

Figure 34



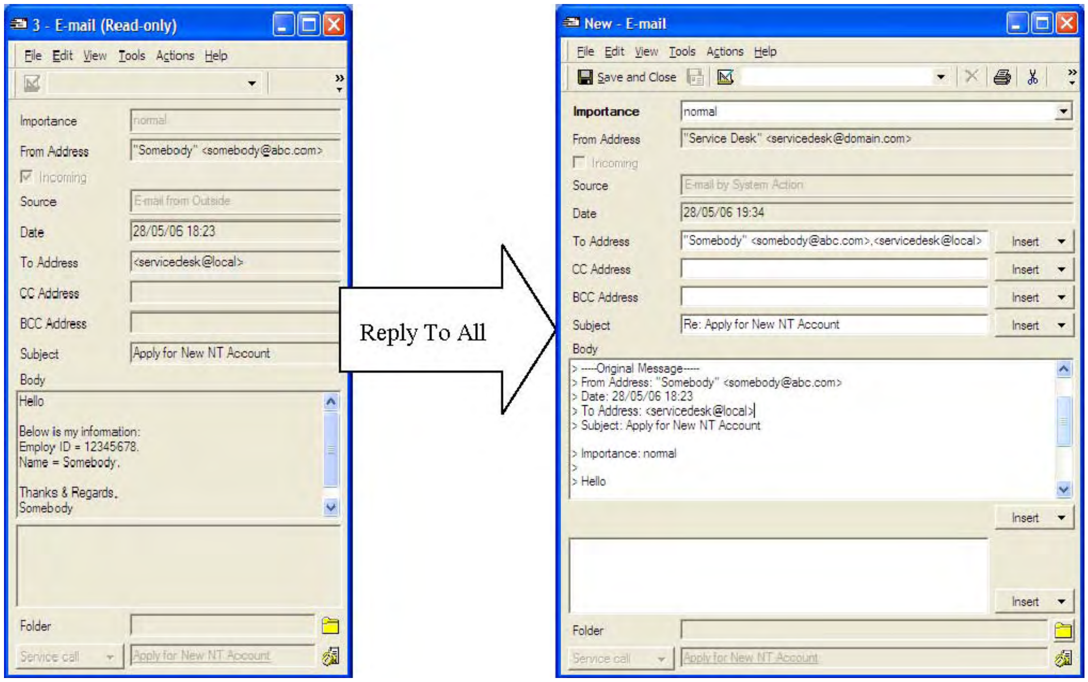
Reply to All

This will create a new E-mail item instance, populated with values from the 'source' E-mail just as commonly known. And, in particular:

- **'To Address'** consists of the **'From Address'** and **'To Address'** of the 'source' E-mail;
- The **'CC Address'** copied directly;

The 'Reply to All' Action therefore will come down to this:

Figure 35



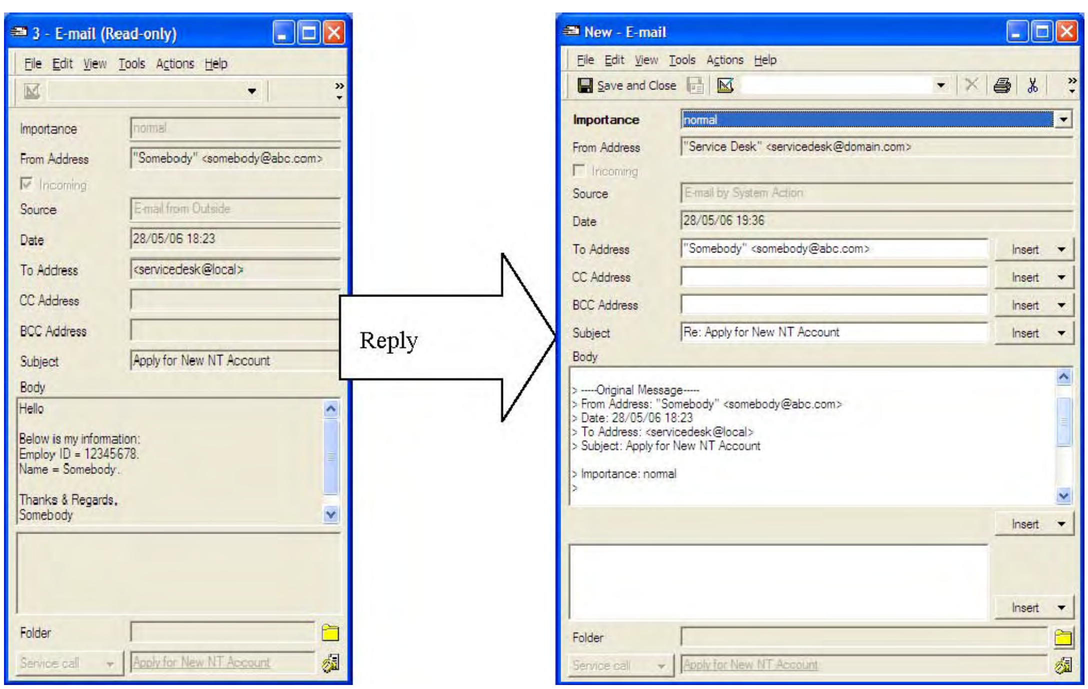
Reply

This will create a new E-mail item instance, populated with values from the 'source' E-mail just as commonly known. And, in particular

- '**To Address**' set to the '**From Address**' of the 'source' E-mail.

'Reply' comes down to something like this:

Figure 36



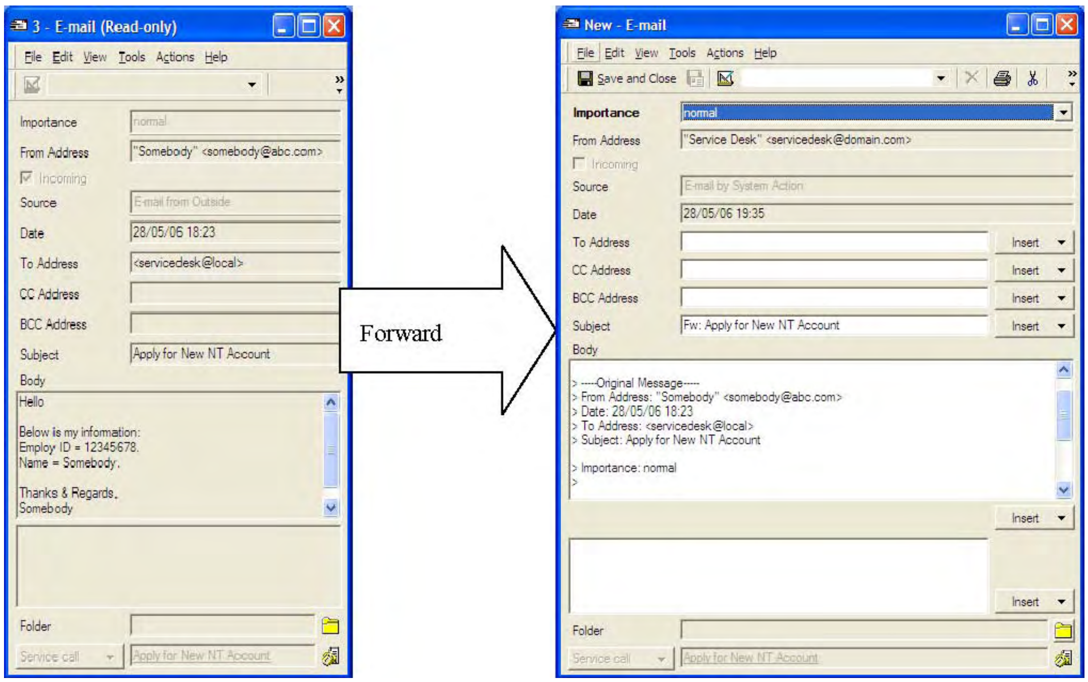
Forward

This will create a new E-mail item instance, populated with values from the 'source' E-mail just as commonly known. And, in particular

- Attachments (initially) copied to the new E-mail. Attachments can be subsequently removed as desired.

The 'Forward' Action therefore will come down to something like this:

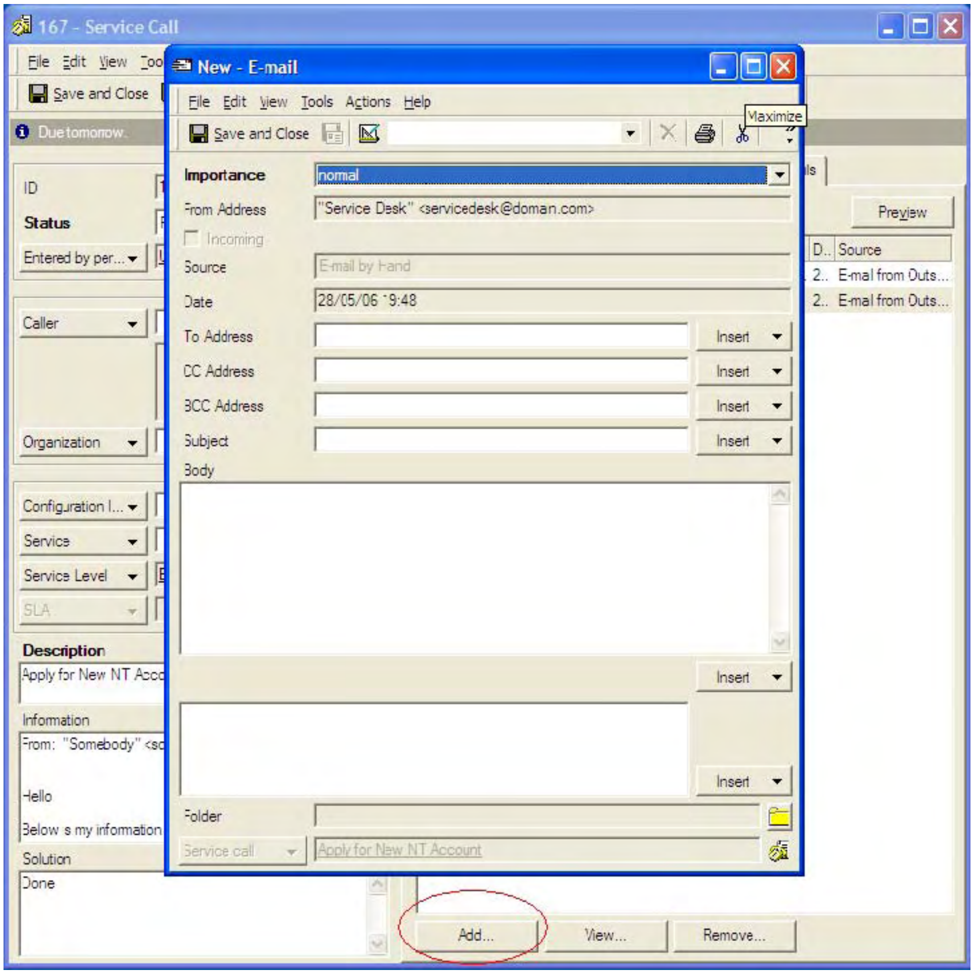
Figure 37



Manually Adding (and Sending) E-mail

Click the **Add** button on email list of Service Call form, a new E-mail form will be opened.

Figure 38



Fill in necessary fields and click **Save and Close**, an email will be sent out immediately and email history will be stored at the same time.

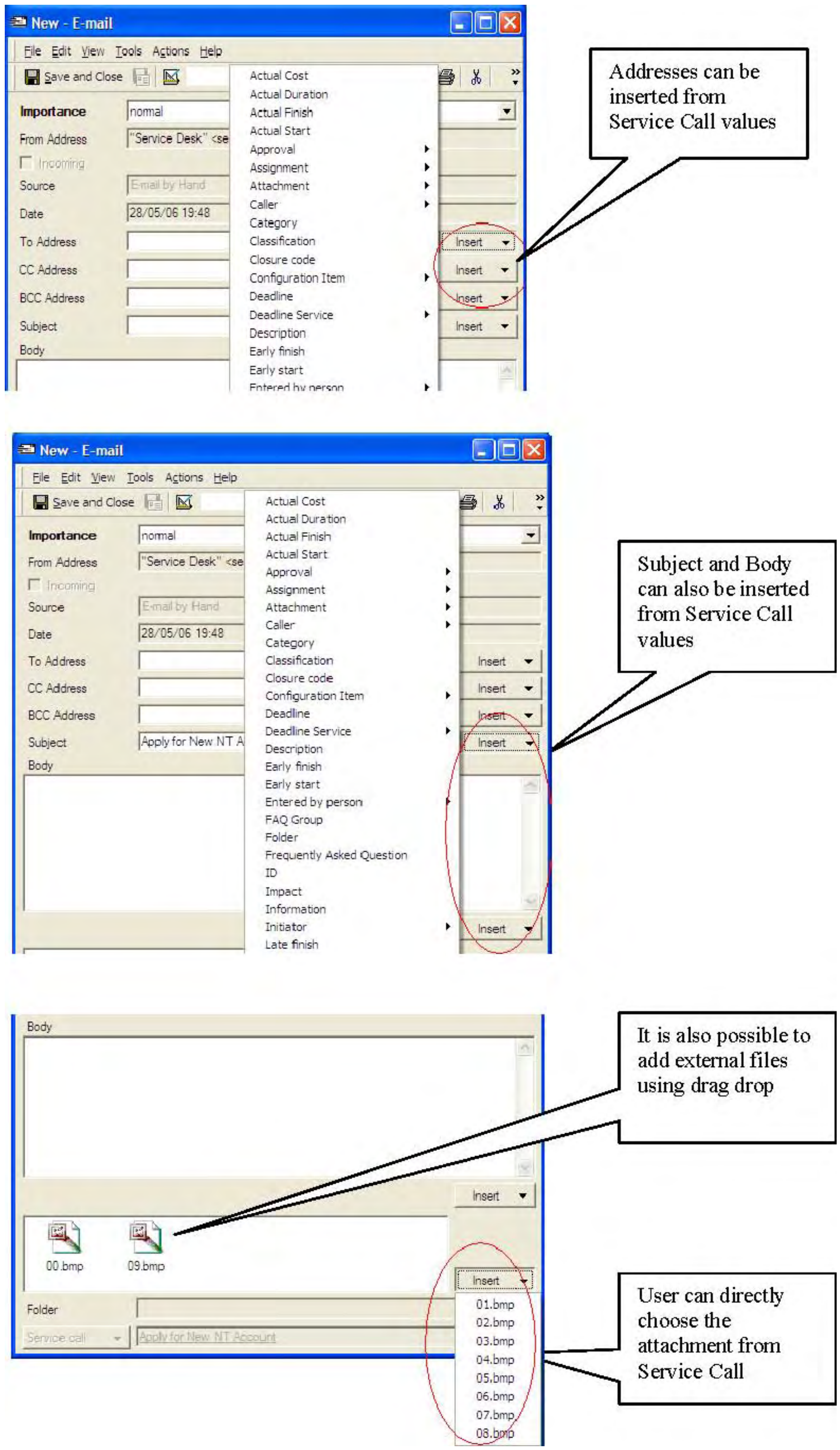
Save and send of Email is independent from the save of service call.

Inbound email is always read-only, outgoing email will be read-only from the moment it is sent.

Emails can be viewed, searched or deleted (if user has appropriate privilege).

Manipulating E-mail Form

Figure 39

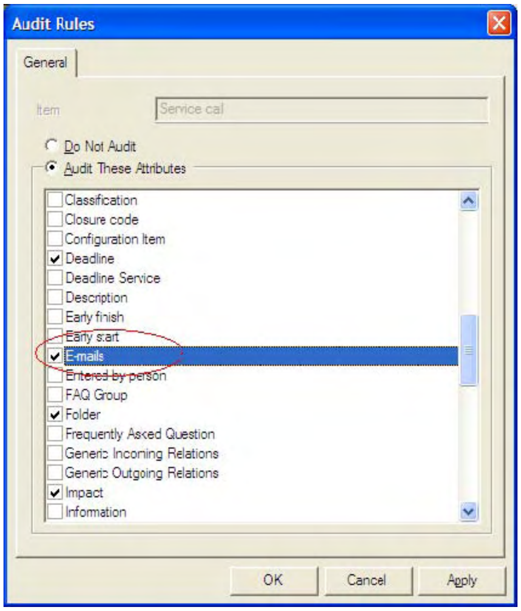


Auditing

E-mail Service Call is auditable.

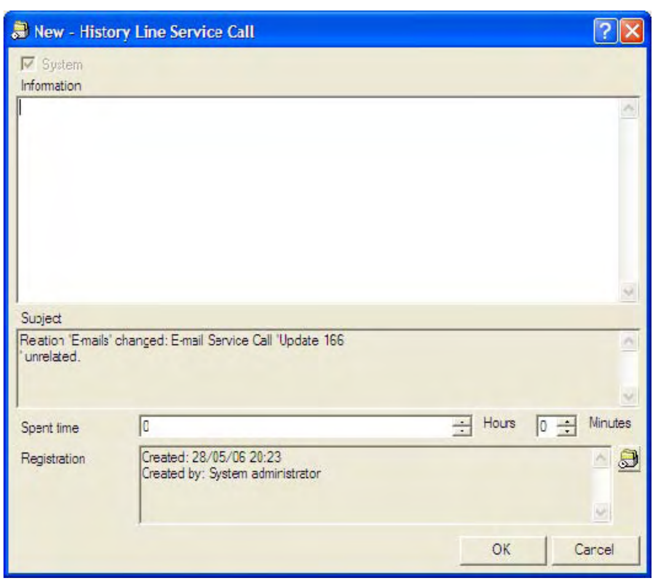
To enable the audit, go to **Administrator Console** → **Security** → **Audit** → **Audit Rules** → **Service Call** and check "E-mails".

Figure 40



The removal of Emails will be logged to Service Call historyline.

Figure 41



Additional E-mail Features

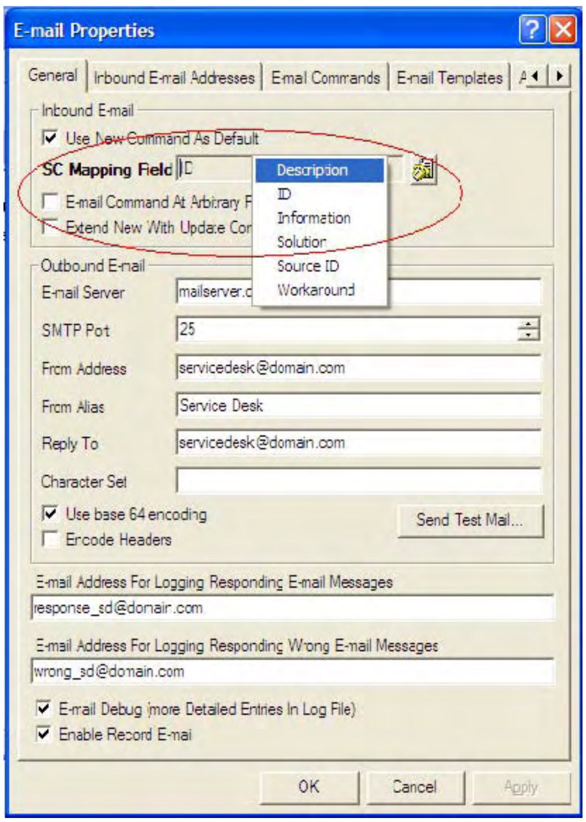
Service Call Identifier Other Than ID

For flexibility, Service Desk has been extended with possibility of specifying different Service Call Identifier Fields; before this a Service Call always was identified by its ID, so an incoming E-mail command always looked like the following:

```
'Update 123'
```

Now it is possible to specify other number or string Field from a Service Call (including custom field) to serve as identifier.

Figure 42



This means now, the E-mails like the following are possible:

```
'Update Apply for New NT Account'
```

The new Identifier may have duplicated values. If Service Desk finds multiple Service Calls match with the same inbound email, the inbound email will not be processed.

E-mail Command at Arbitrary Place

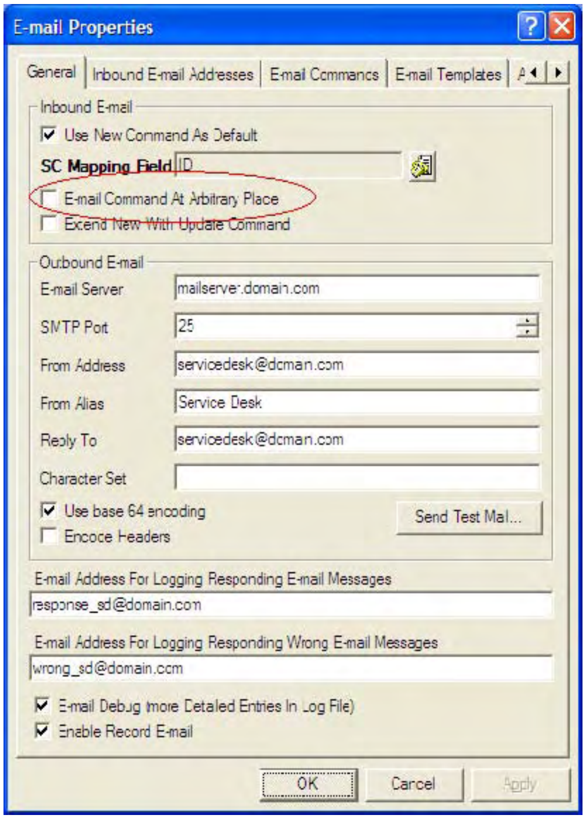
Service Desk only recognizes the entire string following the E-mail Command to be the Service Call Identifier. So, the E-mail:

```
'Re: Fw: Update Apply for New NT Account'
```

would result in Service Desk responding that it can not understand the command.

For flexibility, Service Desk has been extended with the possibility of looking for the command at arbitrary place.

Figure 43



With this option enabled, Service Desk will try to locate the command not only at the beginning of subject but also at the middle of subject. If a command is found, all preceding characters will be ignored.

In this case,

```
'Re: Fw: Update Apply for New NT Account'
```

will be treated as

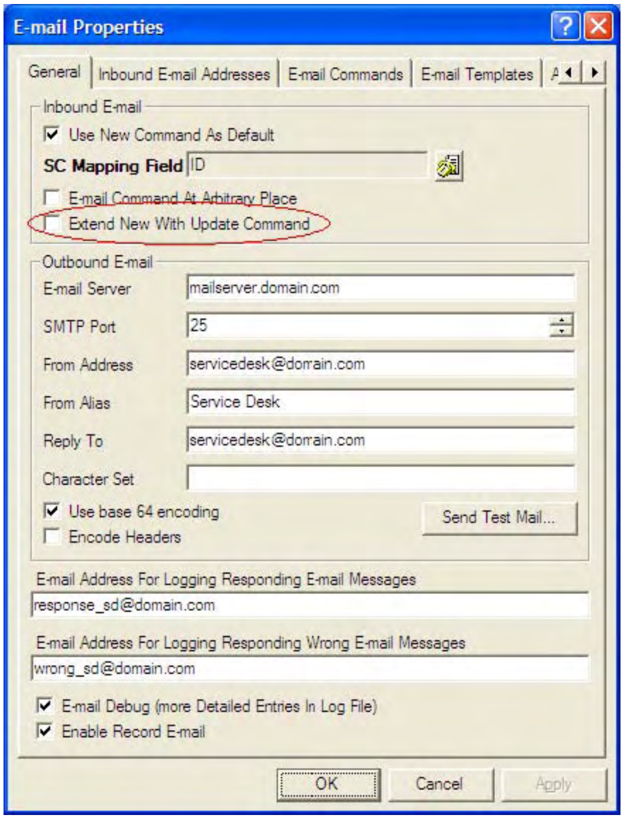
```
'Update Apply for New NT Account'
```

Extend New with Update Command

With the 'New' command only few Service Call fields can be set, others are taken from configured template.

For flexibility, Service Desk has been extended with the possibility of combining **New** command with **Update** function.

Figure 44

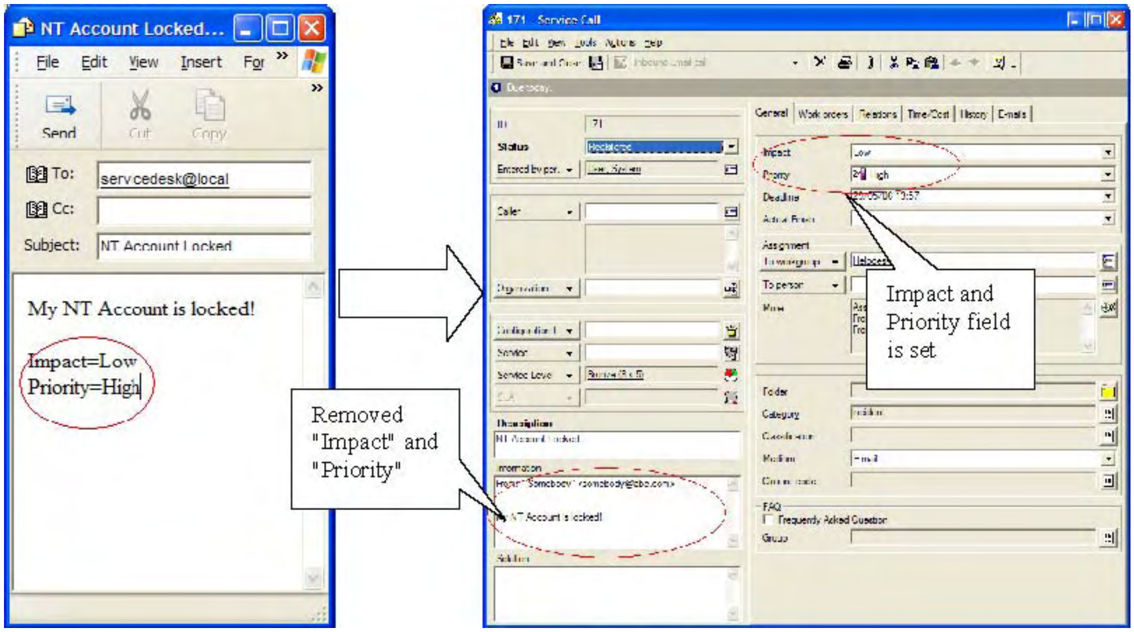


With this option enabled, if body of the E-mail which creates Service Call is formatted like a possible 'Update' command, field update is done at the same time when service call is created.

The body text will still be stored in Information Field, meanwhile additional fields will be set if it can be recognized by Service Desk.

Below is an example showing how inbound email works with the "Extend New with Update Command".

Figure 45



Known Errors and Limitations

- If the client is running on Windows with large font (120 DPI), some of the newly added Email options in Administrator Console may be invisible.

  To solve it, please switch the Windows Display properties to small font (96 DPI).

- There is a potential security policy issue with this feature. In order to use the feature, Service Desk Client will have to connect Outbound Email Server directly to send outbound email. If the security policy for Outbound Email Server allows access only to authorized users, this feature will not work properly.

10 Service Pack 20

This chapter describes new features in Service Pack 20.

Flexible Assignment Notifications

Introduction

The flexible assignment notifications feature is an extension of the notification mechanism. It enables users to distinguish between two types of assignment: when the item is assigned to the user and the user's workgroup.

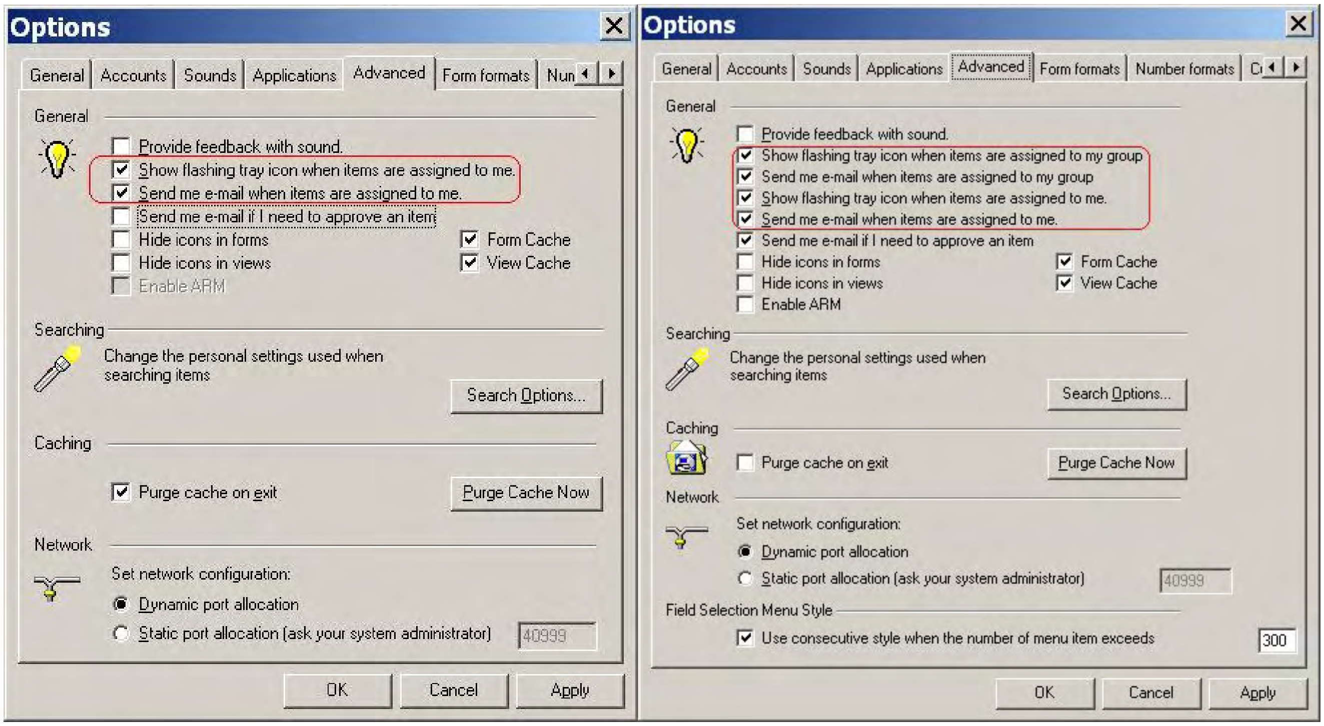
Getting Started

New Notification Options

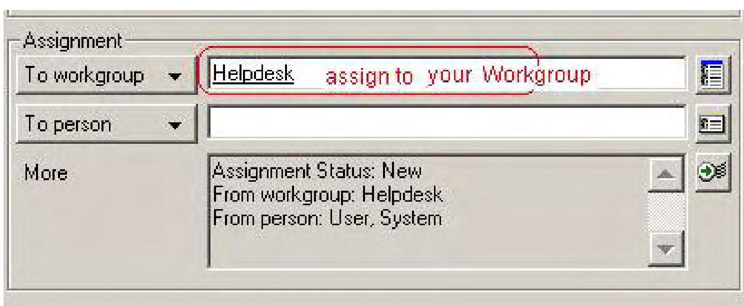
Before SP20, the notification options have only one condition: when the items are assigned to you. It means the notification is triggered when an item is assigned to you or when an item is assigned to the workgroup you belongs to without specifying to person.

From SP20, the notification options have two conditions: when the items are assigned to you and when the items are assign to your workgroup. In this way, the assignment is refined to specify the possibility whether you would like to be notified only when the items are assigned to your workgroup; only when items are assigned to you or both. As seen below:

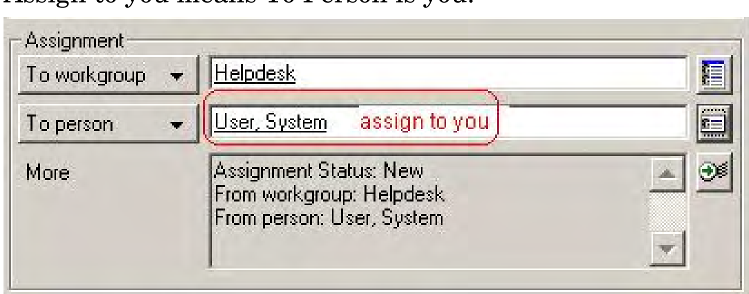
Figure 1



Assign to your workgroup means To workgroup is which you belong to and To person is blank.



Assign to you means To Person is you.



About Existing Notification Options Setting

Existing notification options setting will be translated to the new setting after upgrade.

Table 1

Notification check on Before Upgrade	Notification check on After Upgrade
Show flashing tray icon when items are assigned to me.	Show flashing tray icon when items are assigned to my group. Show flashing tray icon when items are assigned to me.
Send me e-mail when items are assigned to me.	Send me e-mail when items are assigned to my group. Send me e-mail when items are assigned to me.
Show flashing tray icon when items are assigned to me. Send me e-mail when items are assigned to me.	Show flashing tray icon when items are assigned to my group. Show flashing tray icon when items are assigned to me. Send me e-mail when items are assigned to my group. Send me e-mail when items are assigned to me.

Summary

Each user can independently set his or her assignment notification preferences. Any combination of preferences can be chosen, depending on individual needs.

Table 2

Workgroup		Person		Notification Type
Flash tray icon	Send email	Flash tray icon	Send email	
X	-	-	-	User's tray icon will flash when an item is assigned to his workgroup
-	X	-	-	An email will be sent to user's address when an item is assigned to his workgroup
-	-	X	-	User's tray icon will flash when an item is assigned to him personally
-	-	-	X	An email will be sent to user's address when an item is assigned to him personally
X	X	-	-	User's tray icon will flash and an email will be sent to his address when an item is assigned to his workgroup
X	-	X	-	User's tray icon will flash when an item is assigned either to his workgroup or to him personally
X	-	-	X	User's tray icon will flash when an item is assigned to his workgroup; an email will be sent to his address when an item is assigned to him personally

Workgroup		Person		Notification Type
Flash tray icon	Send email	Flash tray icon	Send email	
X	X	X	-	User's tray icon will flash when an item is assigned either to his workgroup or to him personally; an email will be sent to his address when an item is assigned to his workgroup
X	X	-	X	User's tray icon will flash and an email will be sent to his address when an item is assigned to his workgroup; an email will be sent to his address when an item is assigned to him personally
X	-	X	X	User's tray icon will flash when an item is assigned to his workgroup or to him personally; an email will be sent to his address when an item is assigned to him personally
X	X	X	X	User's tray icon will flash and an email will be sent to his address when an item is assigned to his workgroup or to him personally
-	X	X	-	An email will be sent to user's address when an item is assigned to his workgroup; his tray icon will flash when an item is assigned to him personally
-	X	-	X	An email will be sent to user's address when an item is assigned to his workgroup or to him personally
-	X	X	X	An email will be sent to user's address when an item is assigned to his workgroup or to him personally; his tray icon will flash when an item is assigned to him personally
-	-	X	X	User's tray icon will flash and an email will be sent to his address when an item is assigned to him personally
-	-	-	-	No notifications will be sent at all

Templates Stored in Folders

Introduction

This feature provides the means to maintain Templates in Folders. It allows Templates to be saved in Folders. User roles can be authorized to maintain templates at item level via the existing option "maintain template" and at Folder level via the new item "Template" in the Role dialog. The current general authorization for templates will be converted during database upgrade.

Getting Started

Role Setting

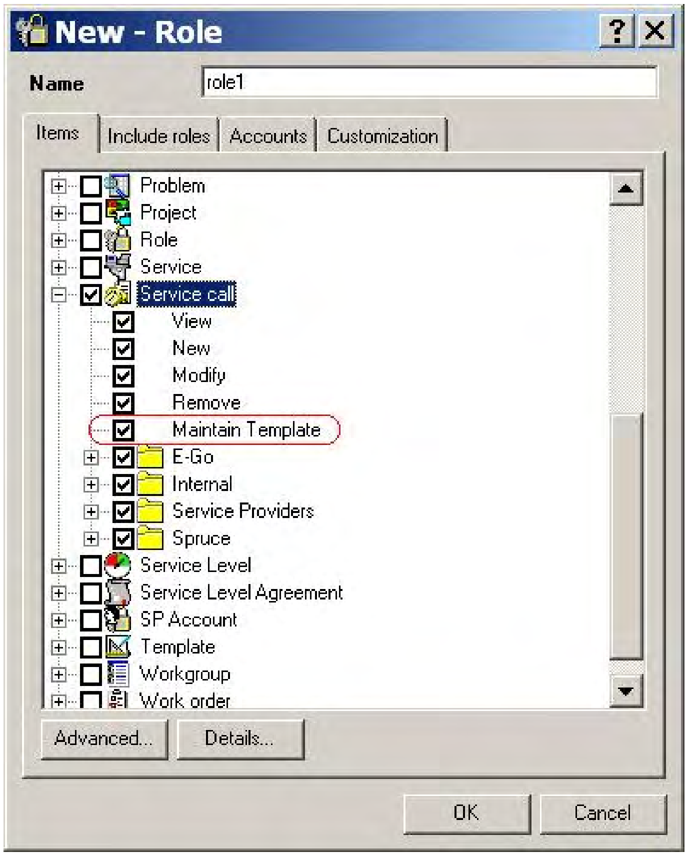
The Role dialog has been enhanced to include the new item "Template", which allows the administrator to authorize Templates in general, also called the root folder, or in specific folders.

Before SP20, if the role is allowed to use Templates, the administrator needs to relate Templates via the item's Advanced Item Access. If the role is allowed to maintain Templates, the administrator needs to check on the option "maintain templates" of the item.
From SP20, there is no change to use Templates. But to maintain Templates, the role should be authorized at both item level and Folder level. At item level, the administrator still needs to check on the option "maintain templates" of the item. At Folder level, the administrator needs to operate the new item "Template" in Role dialog. The folders authorized for Template are common across all item types.

For example, if a user can maintain Templates in folder E-go and Internal for Service Call, the role should be set as below:

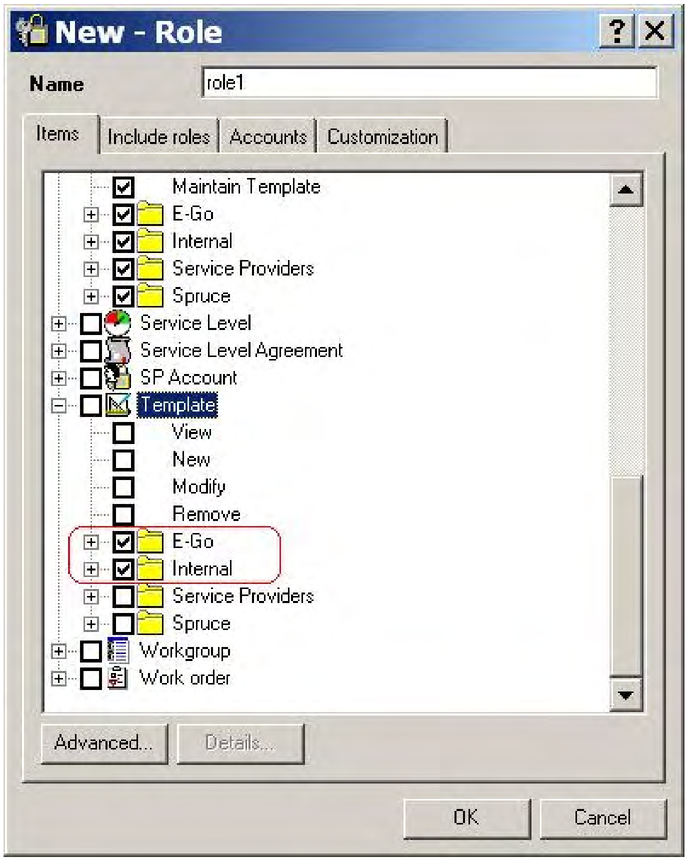
1. Check on Maintain Template for Service Call.

Figure 2



2 Authorize folders for Template.

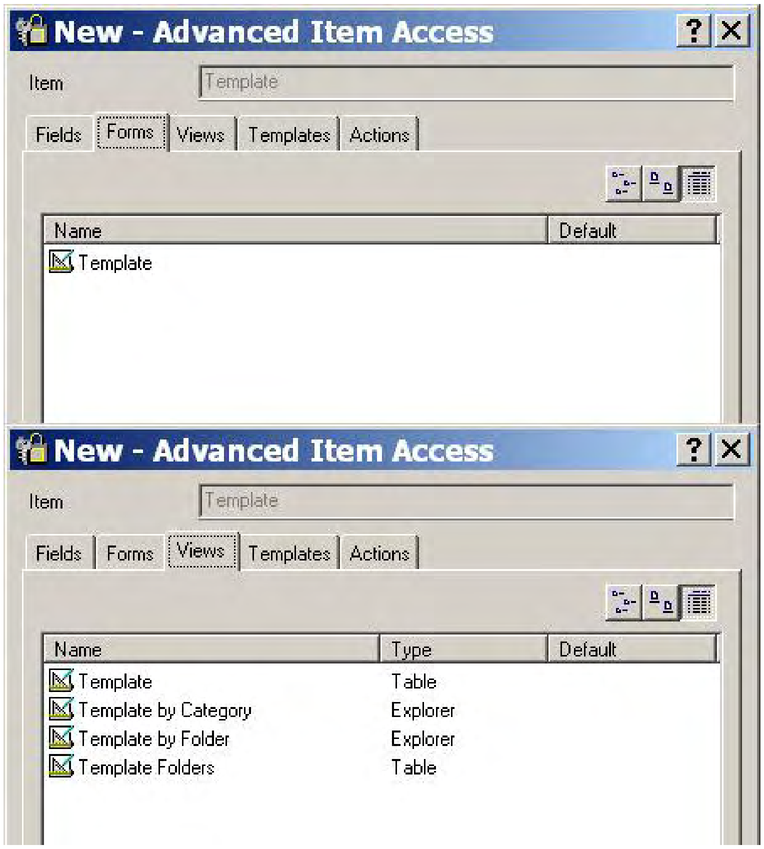
Figure 3



3 Advanced item Access for Template.

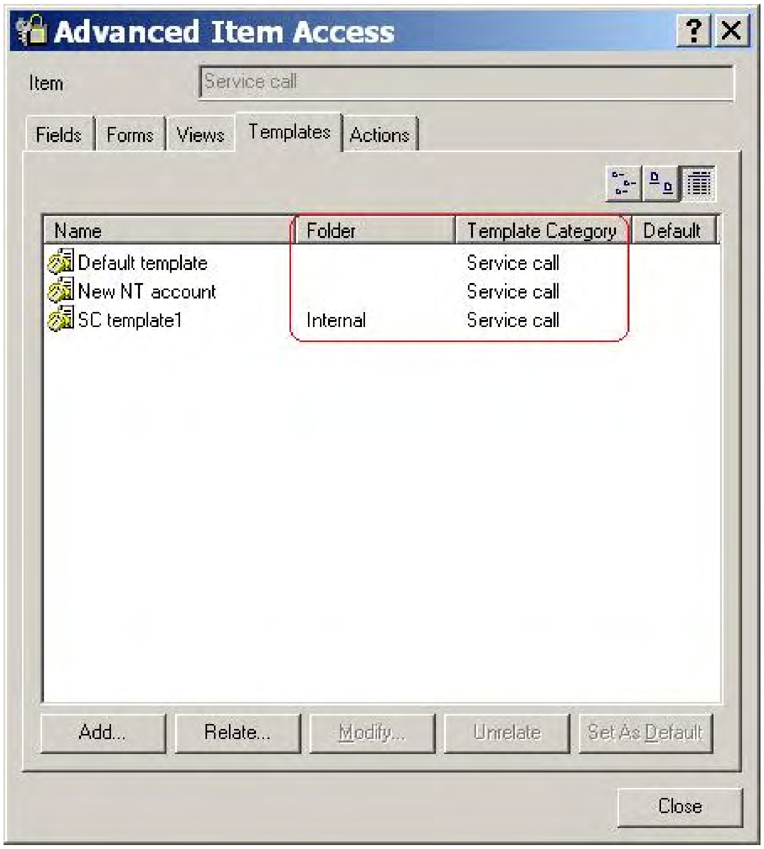
Template View and Template Form are prerequisite for maintaining template.

Figure 4



➤ Two columns, "Folder" and "Template Category", have been added to the "Template" tab of the "Advanced Item Access" dialog to help users distinguish two grouping types for Templates.

Figure 5



Maintain Template in Administrator Console

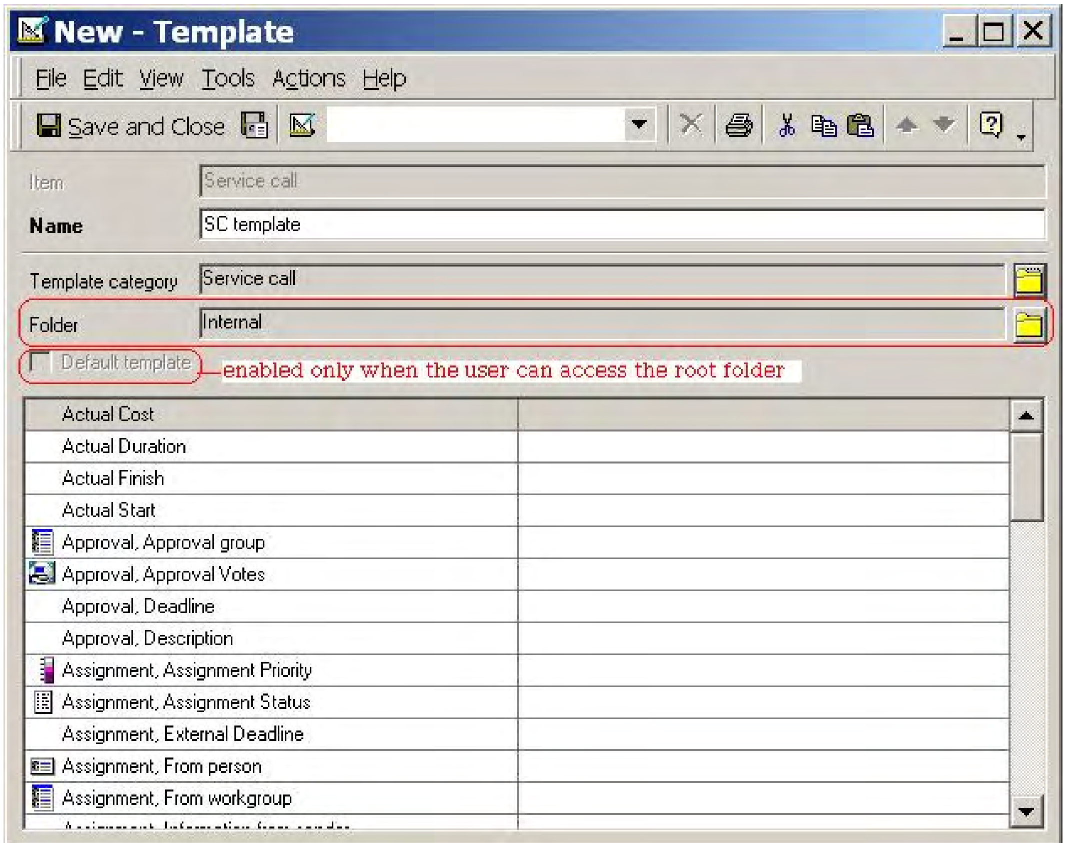
Template View

There are two new views for Template in Administrator Console. A table view, based on the current "Template (Table)" view with an additional column for Folder. And an explorer view, to show the templates by folder hierarchy.

Template Form

Two changes for Template Form: one is that the form contains a new field, the Folder. The other has to do with user authorization: for users with authorization for the template's root folder, the "Default template" checkbox will be enabled, to keep the authorization equivalent to what it was in previous versions of SD. For users who only have access to one or more folders, this field will be disabled, preventing potential problems when the current default template is in an unauthorized folder for the user.

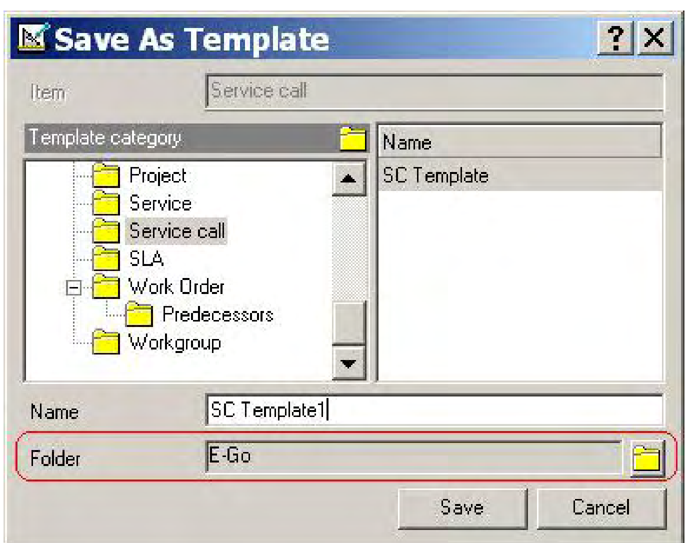
Figure 6



Save Template via Item Form

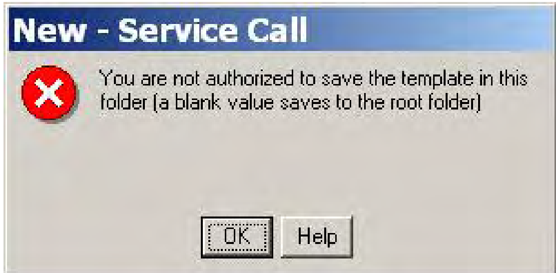
There is one change in dialog save as template, a new field "folder" to let the user save the template in folder.

Figure 7



The folder can be left blank, to save the template in the root folder, or its value can be selected from the standard find dialog, which pops up after clicking on the folder icon to the right of the Folder field. Failure to choose a folder for which the user has "New" authorization will result in an error message, as seen below:

Figure 8



Choose Template via Choose Template Dialog and Item Form

SD let the user choose a template to create items. From SP20, only the users have been authorized "View" entitlement for Template can use this function. Or the user will get error message because they cannot view/modify the attribute "template" of the item.
The roles created before SP20 will be granted "View" entitlement for template automatically during the database upgrade phase if the role has templates related to one or more items.

Upgrade Conversion

The DB configuration wizard automatically invokes a conversion utility when upgrading to the versions from SP20. This utility grants specific authorizations to each role that has Maintain Template: full authorization for Template at the root folder level, as well as access to the Template Form and the Template Views. This conversion utility will be done for only once, it means if the user updates his database from SP20 to SP21, the conversion will not be invoked again.

Known Errors and Limitations

Entitlement "Remove" for Template Ignores Template Folder

If a role is granted "Remove" entitlement for Template in any folder, SD will allow users with that role to delete any template, irrespective of the folder location. Of course, this only applies to users that also have Maintain Template authorization for one or more item types.

Workaround

Only grant "Remove" entitlement to system administrators. Users will then need to ask the administrator to delete a template. Mitigating circumstances: In practice, most templates will be in use by one or more items. As such, they cannot be deleted anyway.

Entitlement "View" for Template is Prerequisite for Using Template to Create Items

Before SP20, we relate templates to one or more items to let the user use templates when creating items. From SP20, we have to take one more action, granting "View" entitlement for Template. Or Service Desk will skip the choose template dialog to create items. And the user will get error messages when changing the templates in item form.

Auditing Attachments

Introduction

This feature introduces the capability of creating history lines which describe operations Service Desk users perform on attachments. History lines will be created when Service Desk users add or remove an attachment or modify classifications of attachments. This feature applies to 'Change', 'Configuration Item', 'Incident', 'Maintenance contract', 'Organization', 'Person', 'Problem', 'Project', 'Service', 'Service call', 'Service Level Agreement', 'Work order' and 'Workgroup'.

A new functionality, 'Remove Old Attachment Versions', is introduced by this feature, which aims to allow System Administrator to clean up the attachment server by removing all old versions of attachments of a selected set of items with same type.

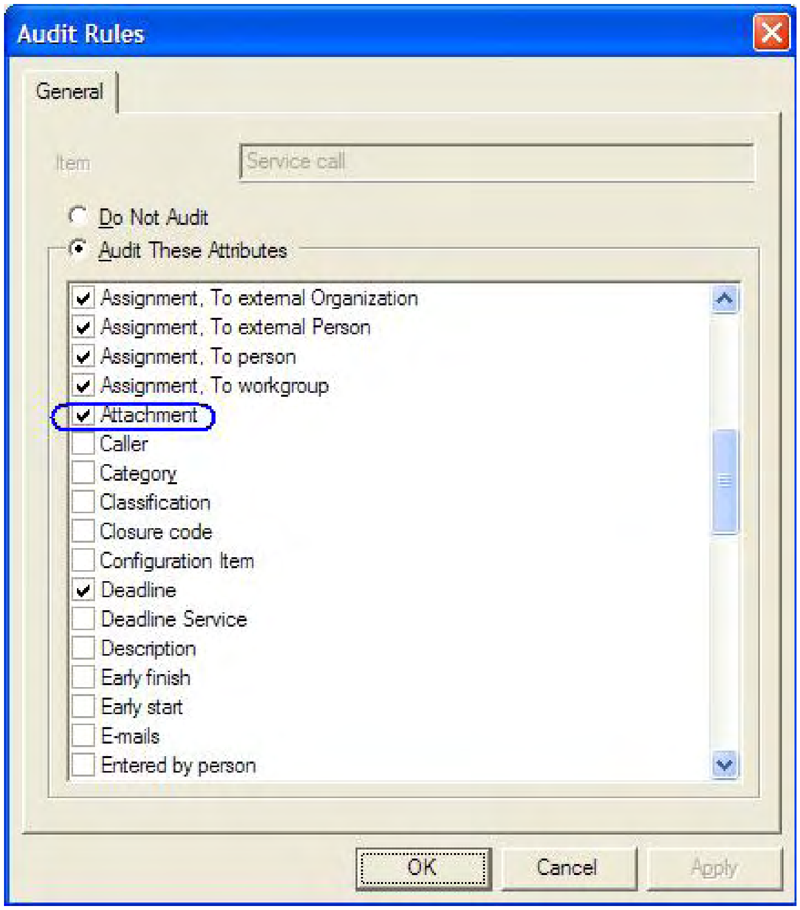
The main audience for this document is Service Desk users and administrators authorized to maintain attachments.

Getting Started

Configuration

A new option is added into configuration page of each item listed on audit rules page. These can be found in Administrator Console under **hp OpenView service desk** → **Security** → **Audit** → **Audit Rules**. To turn on the auditing attachment feature for a certain type item, just check the appropriate option box, as shown in the following illustration.

Figure 9

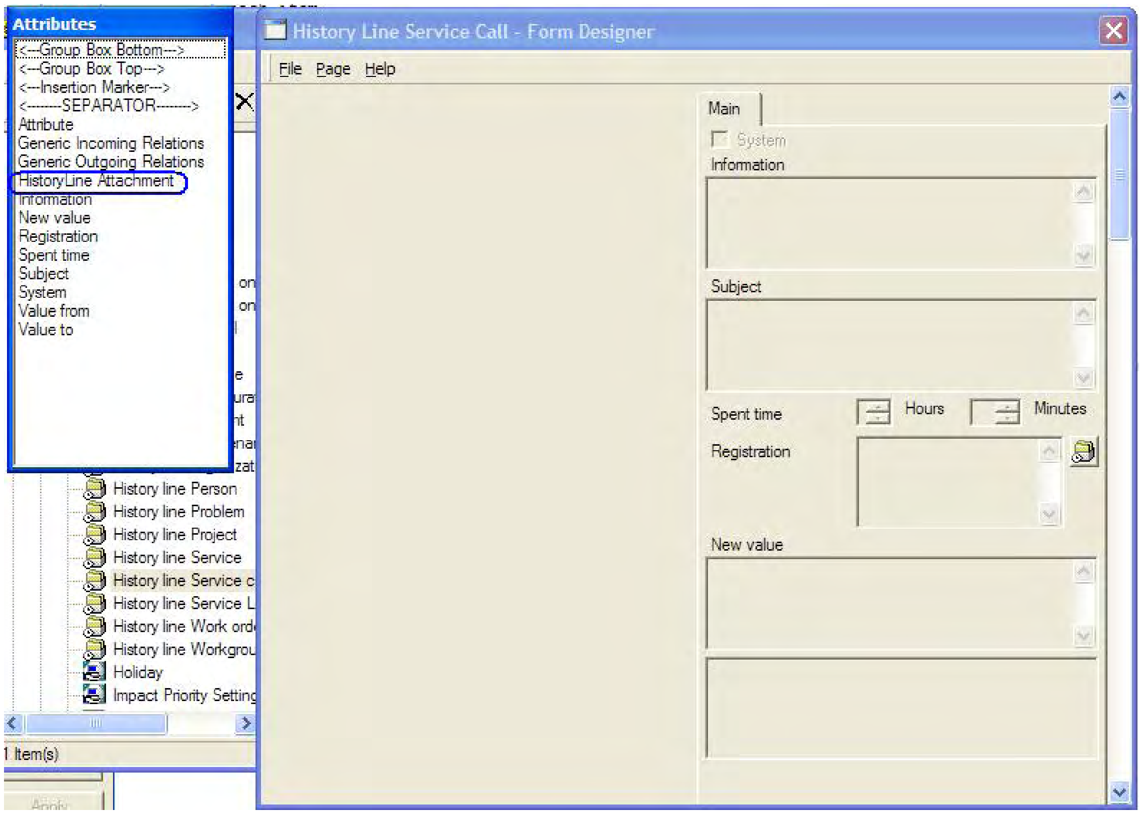


Attachments removed from records by Service Desk users are re-attached to a history line of that record by this feature as an integral part of history data. Users can then open attachments on history lines in order to audit changes made on them.

In order to add removed attachments to history lines, System Administrators need to redesign corresponding history line forms. This can be done by going to Administrator Console, **hp OpenView service desk** →**Presentation** →**Forms**. Click the appropriate history line entry on the left tree and open the appropriate form from the list on the right. As shown in following diagram, a new attribute "HistoryLine Attachment" is added onto the Attributes select dialog, to make the history line attachment field visible. Drag this attribute to the desired place on the form.

The most important information about attachment operations are recorded in the 'Subject' and 'New value' fields of history lines. "New Value" is not added by default, user also need add "New Value" attribute onto the form if he wants to know detailed information of operations.

Figure 10

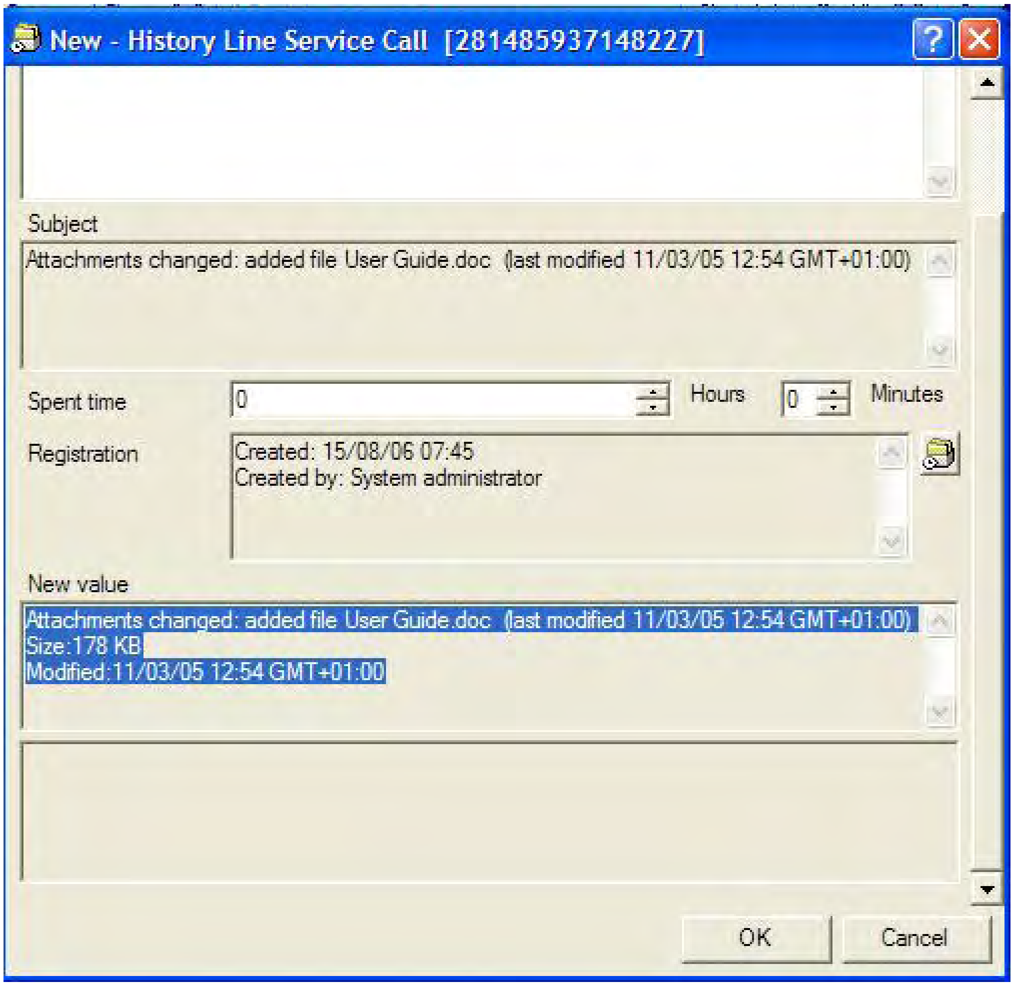


History Lines of Auditing Attachment

Service Desk users can view the content of a history line created by this feature by double clicking one of the history lines.

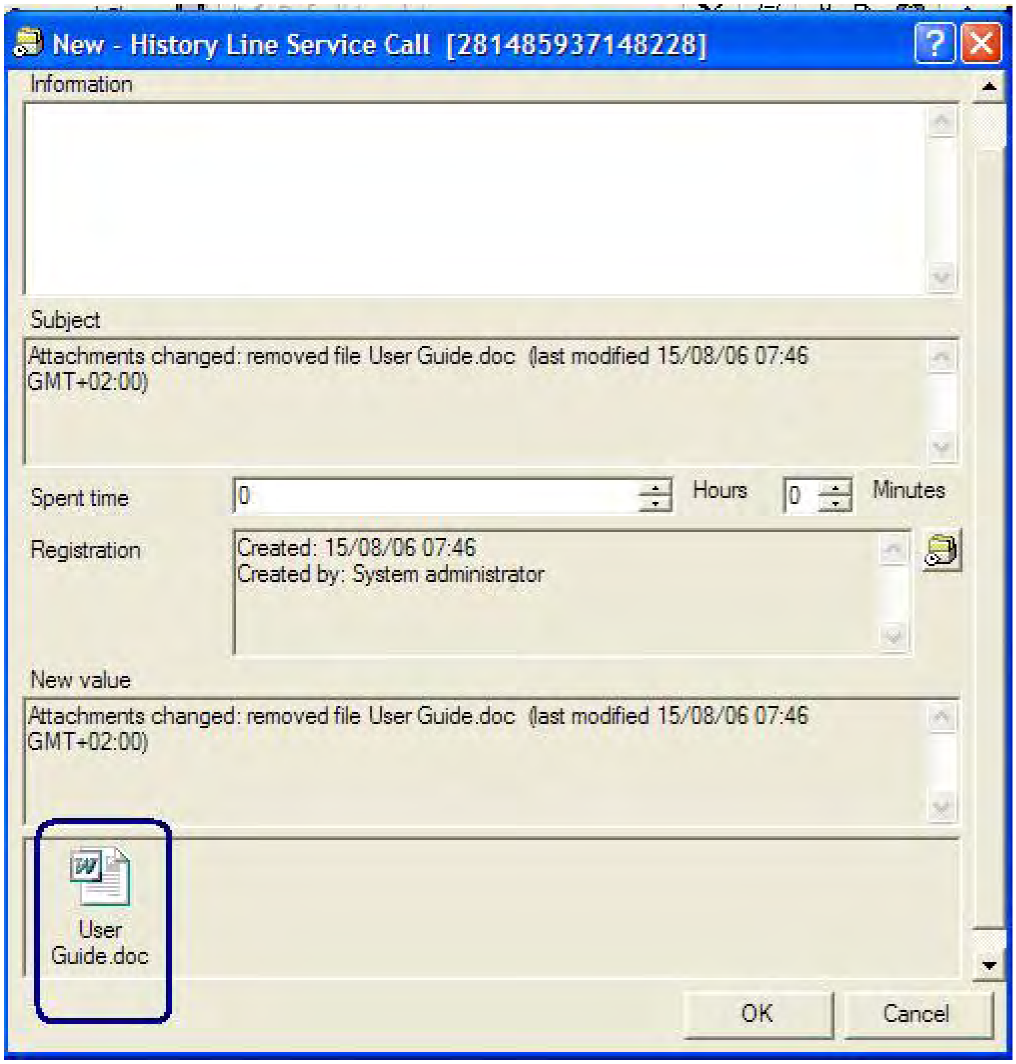
A history line about adding an attachment is shown below as an example:

Figure 11



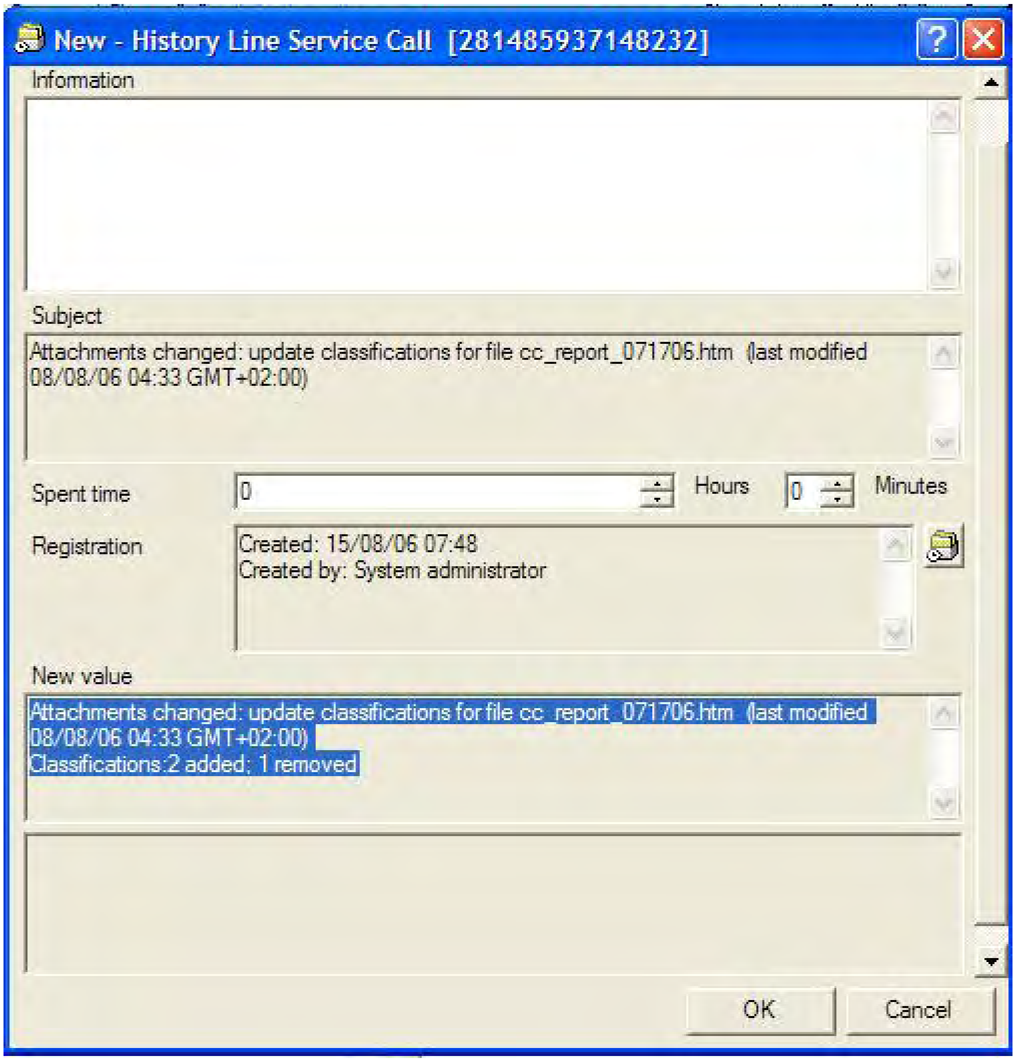
A history line about removing an attachment is shown below as an example; users can open the removed attachment by double clicking the file icon.

Figure 12



A history line about updating classifications of an attachment is shown below as an example:

Figure 13



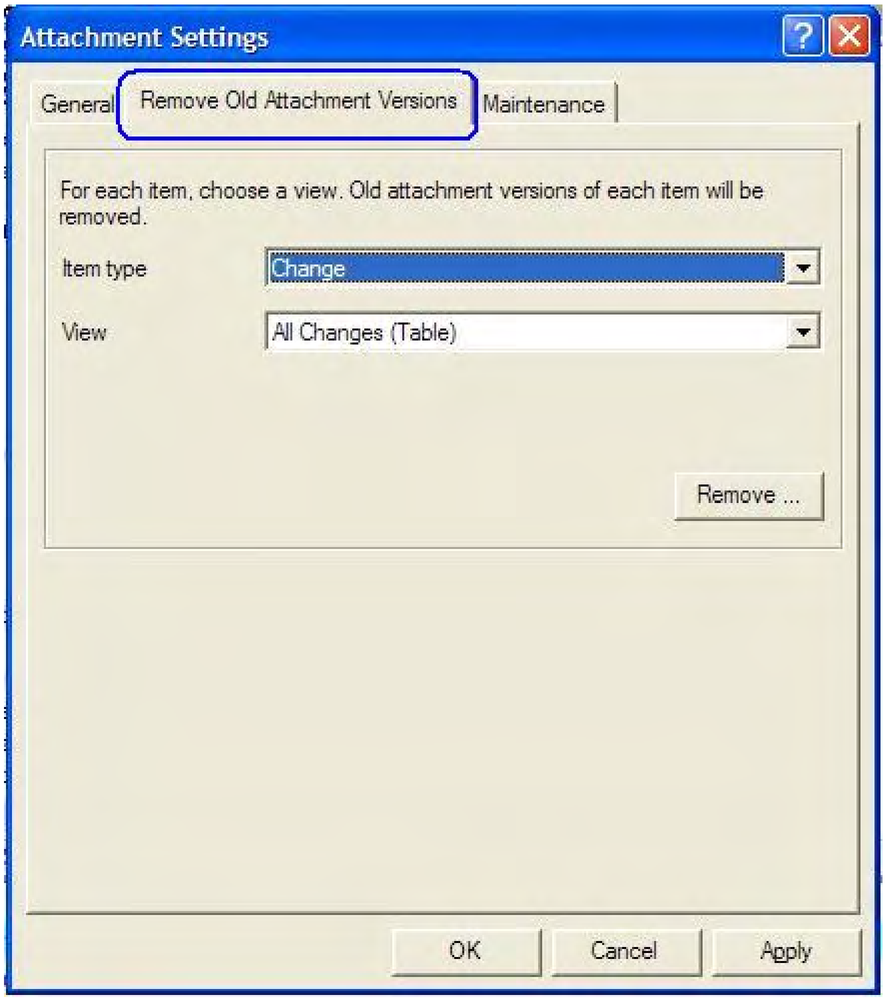
Removing Old Attachment Versions

If the auditing attachment option for a certain kind of item is switched on, Service Desk will then version each attachment attached to that kind of item. Service Desk will treat all attachments with the same display name for a single record (and its history lines) as different versions of that attachment. For example, users modify (by adding and removing) one attachment of a Service call a couple of times, different versions of the attachment with that particular display name are maintained by this Service call (and its history lines); the latest version is the one most recently added. Users then can go through the history lines and open attachments to track changes.

However, the storage space on the attachment server might be seriously impacted by Service Desk when this feature is heavily used. In some cases, old attachment versions of certain items are not relevant anymore. In that case, System Administrators can use the new functionality "Remove Old Attachment Versions" to delete attachment versions.

A new tab page "Remove Old Attachment Versions" is added to Attachment Setting dialog (as shown in following illustration) to allow System Administrators to clean up the attachment server by removing redundant attachment versions. This tab page will only be visible by System Administrators of Service Desk.

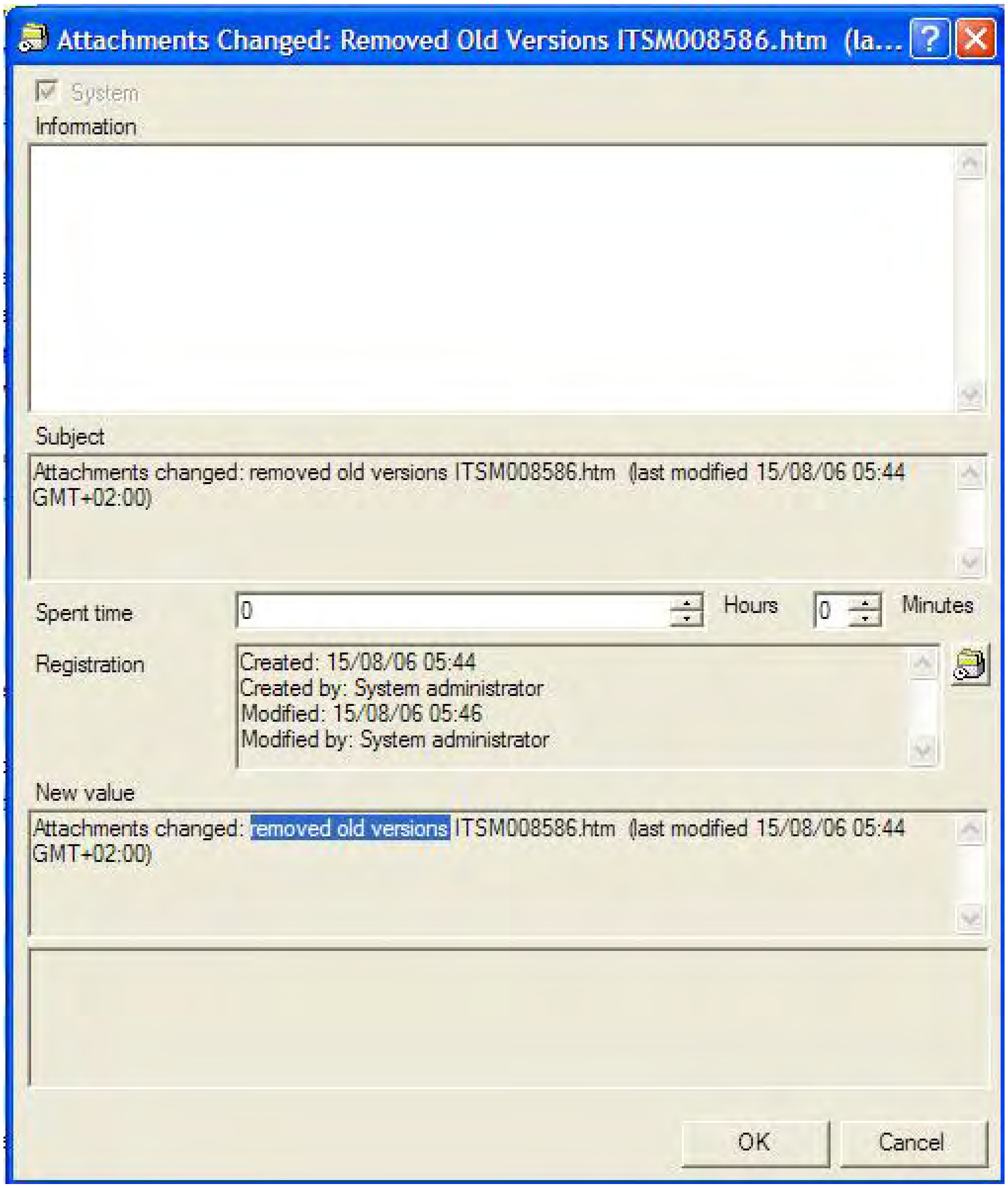
Figure 14



To remove old attachment versions, System Administrators need to first select item type which they want to perform this operation on, and then select a view from the view list. By clicking the **Remove** button, all old attachment versions selected by this view will be removed; only the most recent version will be kept. Besides, Service Desk will update history lines, which maintains these old attachment versions, in order to log this operation.

Through Web-api user can insert cognominal attachments into one record of Service Desk. It may cause those attachments get lost when performing "Remove Old Attachment Versions" function. Please use it carefully!

Figure 15



Web API

There is no change specially made in Web-API. If attachment auditing option is switched on, adding or removing an attachment and modification of classifications of attachments through the Web API will be audited.

Known Errors and Limitations

In Service Page and Web Console, history line forms only have "Information" and "Subject" attributes, which means users can not get detailed information of operations and removed attachment.

Query Restriction Enhancement

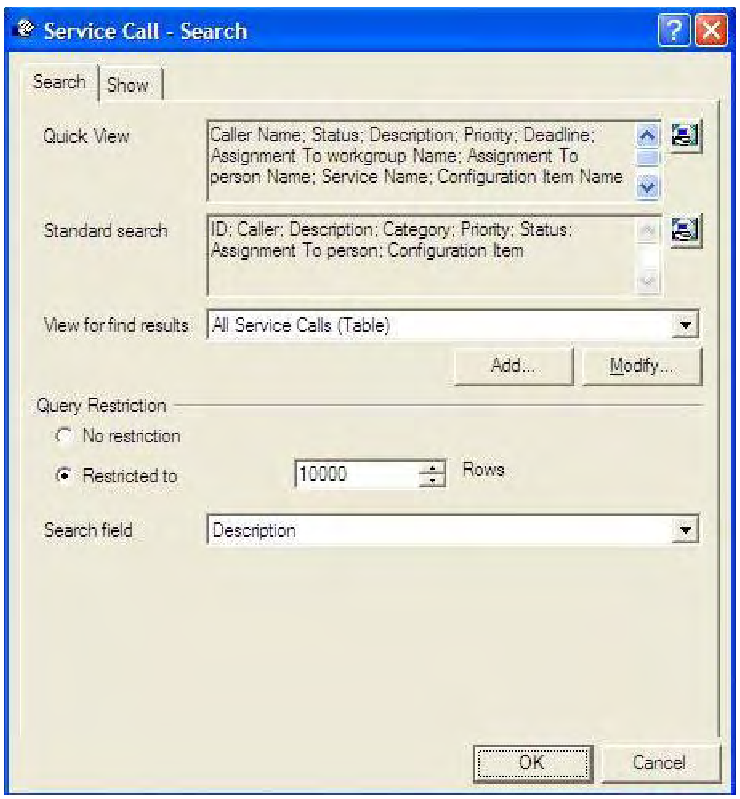
Introduction

Service Desk has provided the possibility to refine query restriction function. With this feature System Administrator can define which part of records should be included when result exceeds query restriction, 'newest records', 'oldest records' or keep original behavior which depends on database storage.

Getting Started

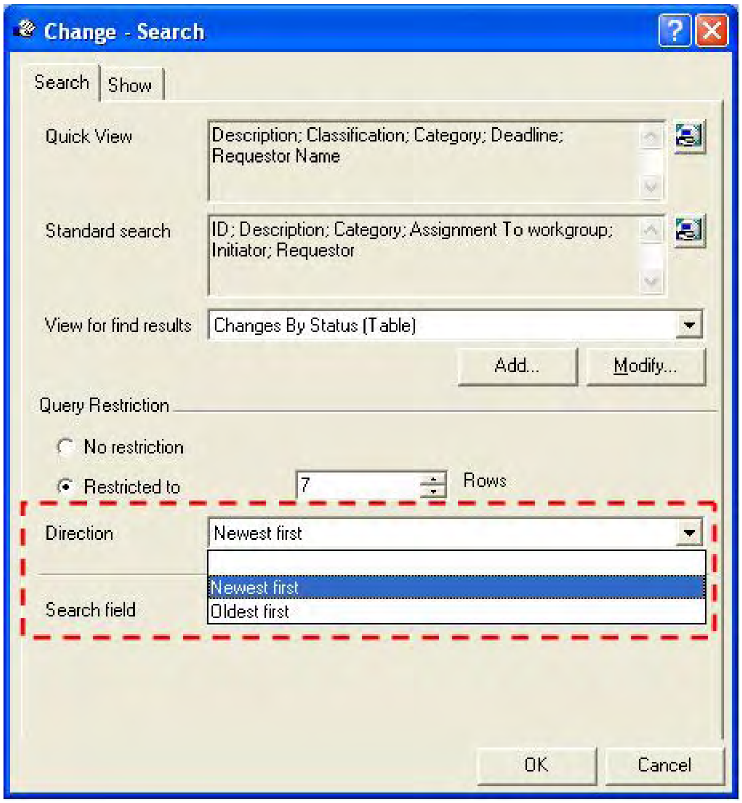
By setting the query restriction System Administrator can limit the number of records retrieved when opening a View. Before SP20, there is no sort order can be set, so the first n records are retrieved depends on database storage.

Figure 16



Starting from Service Pack 20, it is possible to specify whether the resulting ones will be the 'Oldest' ones or the 'Newest' ones; shown as below:

Figure 17



To set the order of a query restriction, navigate to **Administrator Console** → **HP OpenView Service Desk** → **Presentation** → **Search**.

Select a Service Desk Item, like 'Change' and open that item. As shown in the screen above, System Administrator can now select what the sort order *direction* should be before the restriction is applied.

The default value is **empty** one.

- When setting has been configured to be the 'Newest', the *Query Restriction* will result in only the n newest records.
- Similarly, whenever here 'Oldest' has been defined, the *Query Restriction* will result in the n oldest records.
- When nothing has been selected it will depend on database storage (normally the oldest ones will be yielded).

After setting the sort order direction, Service Desk application server should be restarted to be taken into effect.

Query Restriction with 'Newest' or 'Oldest' will only guarantee return n newest or n oldest records and it can not ensure a sorted list in the view. If you have defined "Sort By" in view settings, Service Desk will first return n newest or n oldest records according to setting in Query Restriction, then sort the view according to settings in the view.

Share Views Between Users

Introduction

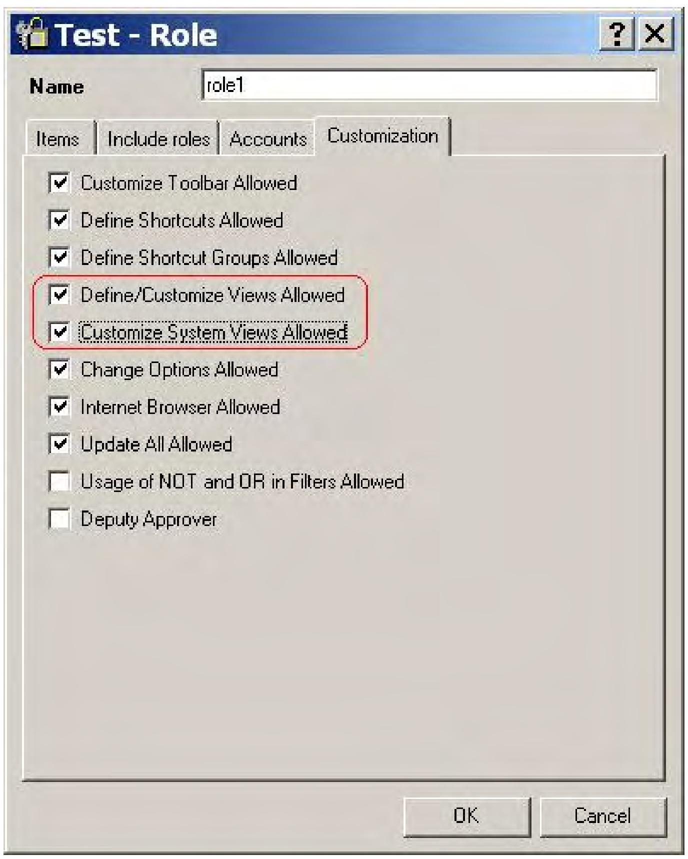
Before SP20, users can only customize the views in their own client environment. All the customized views are saved in file `views.dat`. User cannot share one certain customized view with each other. This feature provides the means to share individual user views via an export/import file mechanism.

Getting Started

Role Setting

Only those users with the role allow define/customize views can export/import views. If the user wants to export System views, he should be allowed customize System views.

Figure 18

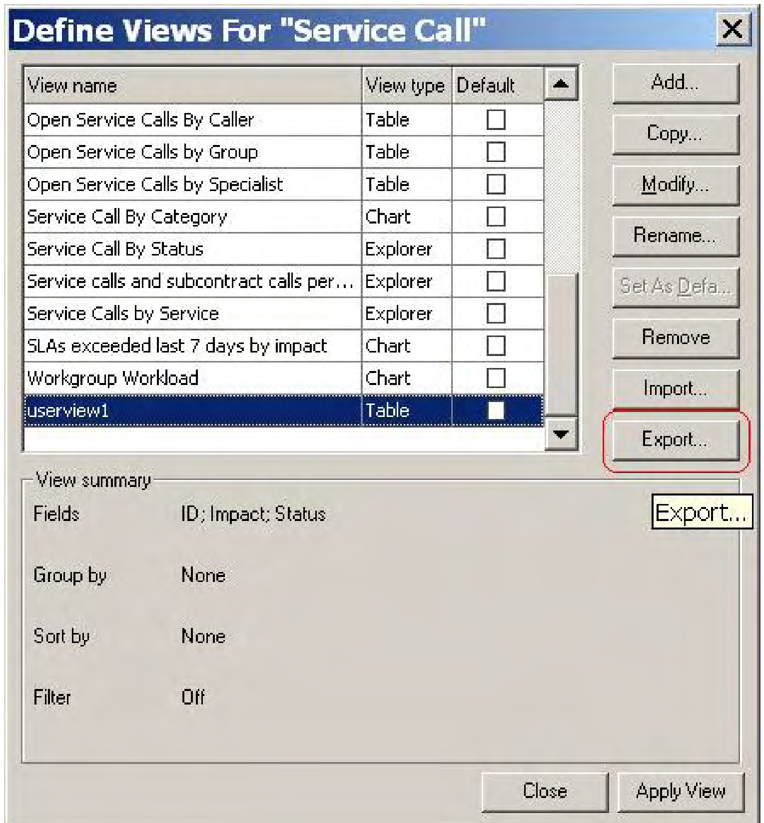


Export a User View

Login to Service Desk Client and navigate to the Define Views dialog. In the Define Views dialog, select a user view to export, and then click the new **Export** button to initiate the export procedure.

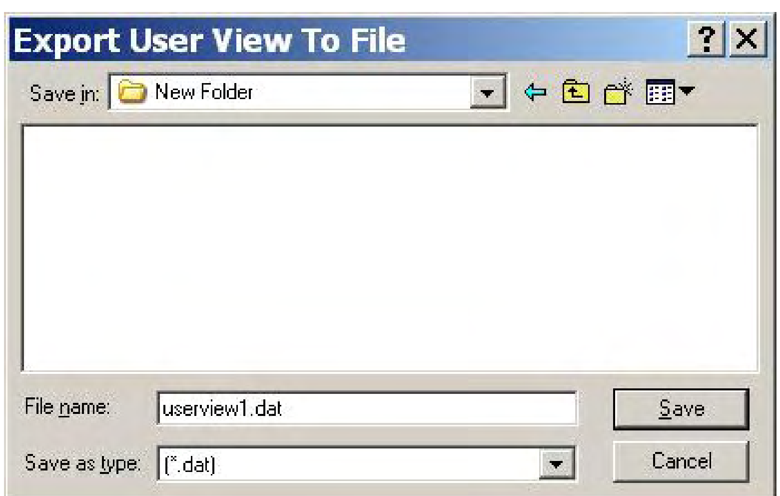
The **Export** button will be inactive when an uncustomized System View is selected.

Figure 19



Accept the default file name (the view's name) or provide a name. Click the **Save** button and then close the "Define Views" dialog box.

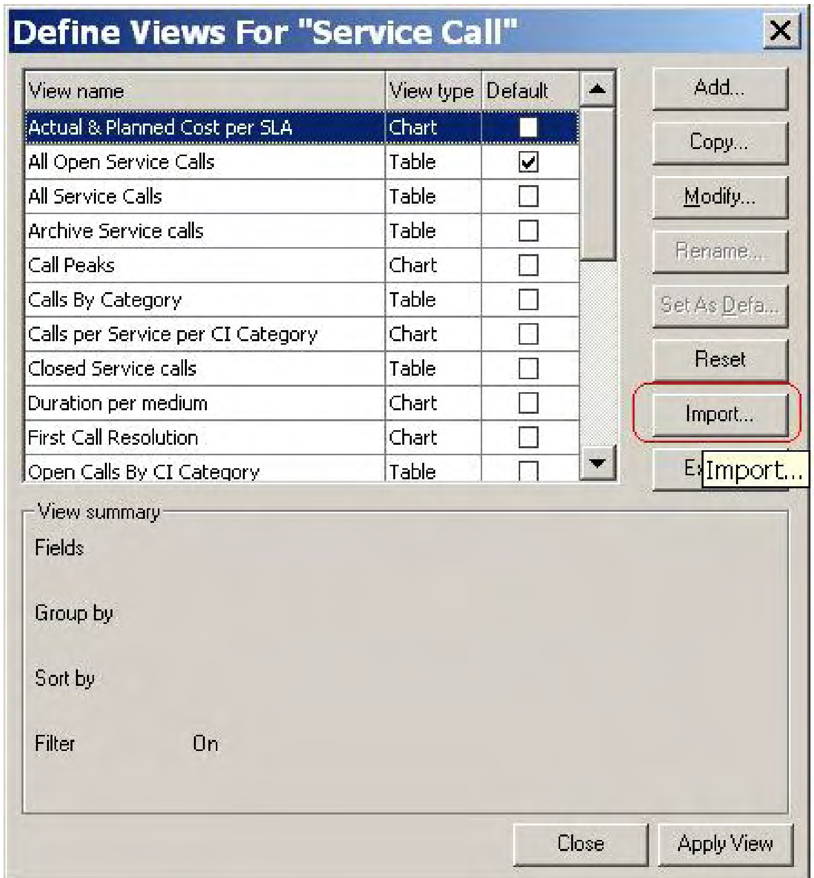
Figure 20



Import a User View

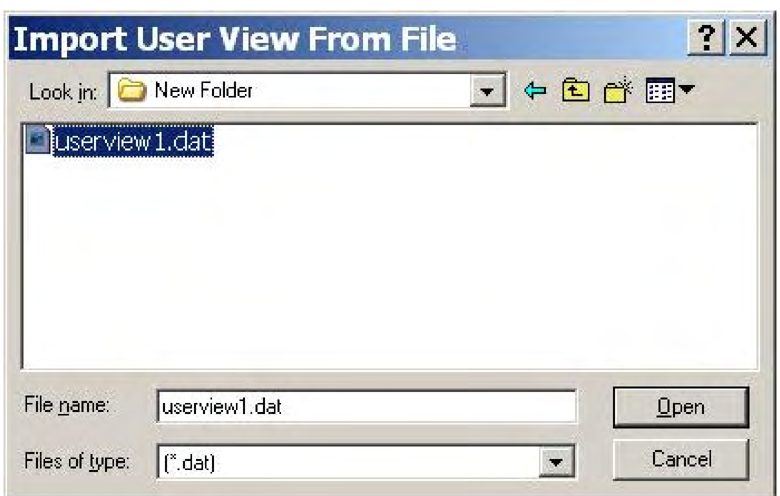
Open the Define Views dialog, and then click the new **Import** button to initiate the import procedure.

Figure 21



Select the user view file you want to import. Click the **Open** button.

Figure 22

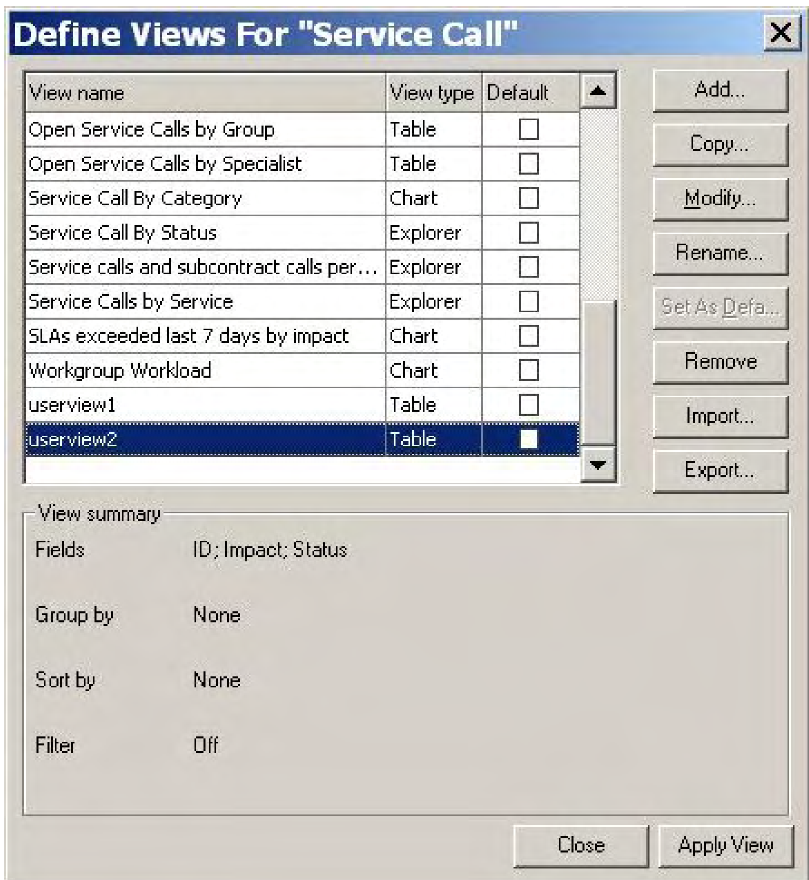


➤ If the name of the imported view already exists, a dialog will pop up to let you specify the view a unique name.

Figure 23

On return to the "Define Views" dialog, you will notice that the imported view is currently selected. To see the view in action, click the **Apply View** button.

Figure 24



Known Errors and Limitations

The feature does not support importing a view which is exported on an OS having different language setting.
Example: export from Japanese WinXP and import to English WinXP. The different language settings in OS may cause import failure.

Deputy Voting on Pending Approval Tasks

Introduction

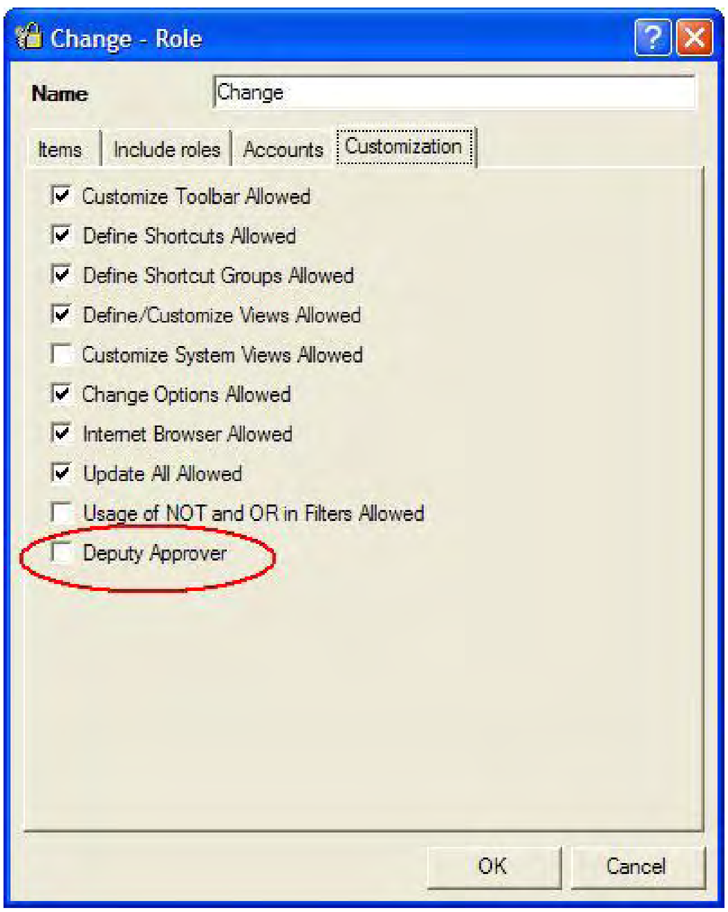
A new feature has been introduced for "deputy voting". With this feature, user is now able to view approval tasks of other people. Also some people can be authorized to vote on behalf of other people who are in long term leave.

Getting Started

Deputy Approver

To provide the possibility of "deputy voting" without having Administrator role, a new option "Deputy Approver" is added to Customization tab page of Role definition. Persons granted with this privilege are allowed to approve Approvals on behalf of any other people:

Figure 25 Deputy Approver Option



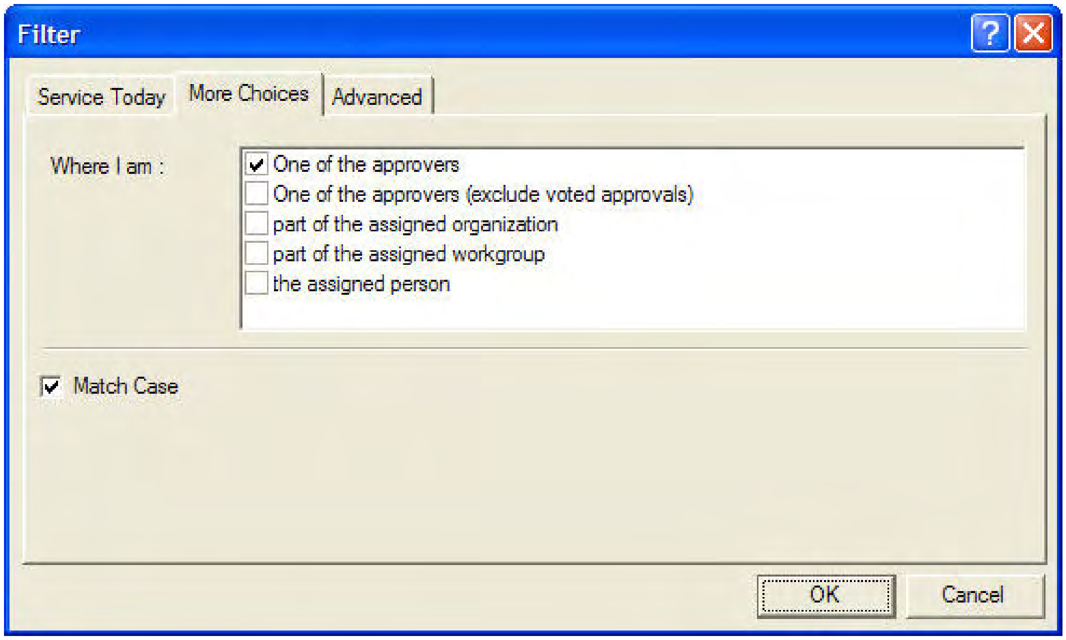
By default, this option is switched off, thus all the existing roles are not deputy approver initially except System Administrator.

Deputy Approver has very powerful privilege, he can vote for any other people, so be careful of using it.

Viewing Pending Approval Tasks of Others

Currently it is possible to list current user's *Approval* Tasks using the filter condition "*Where I am One of the approvers*":

Figure 26 View Filter Setting



It is now possible to use "Approval Approval Votes Approver" as navigation item in explorer view to list approval tasks for other person.

Figure 27 Navigation Field for Viewing Others' Approval

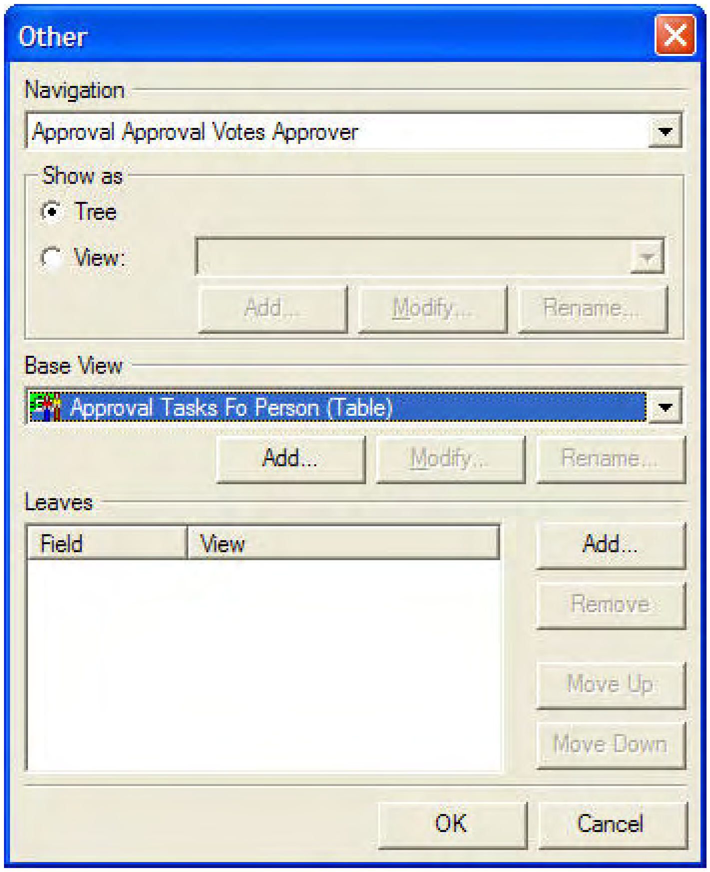
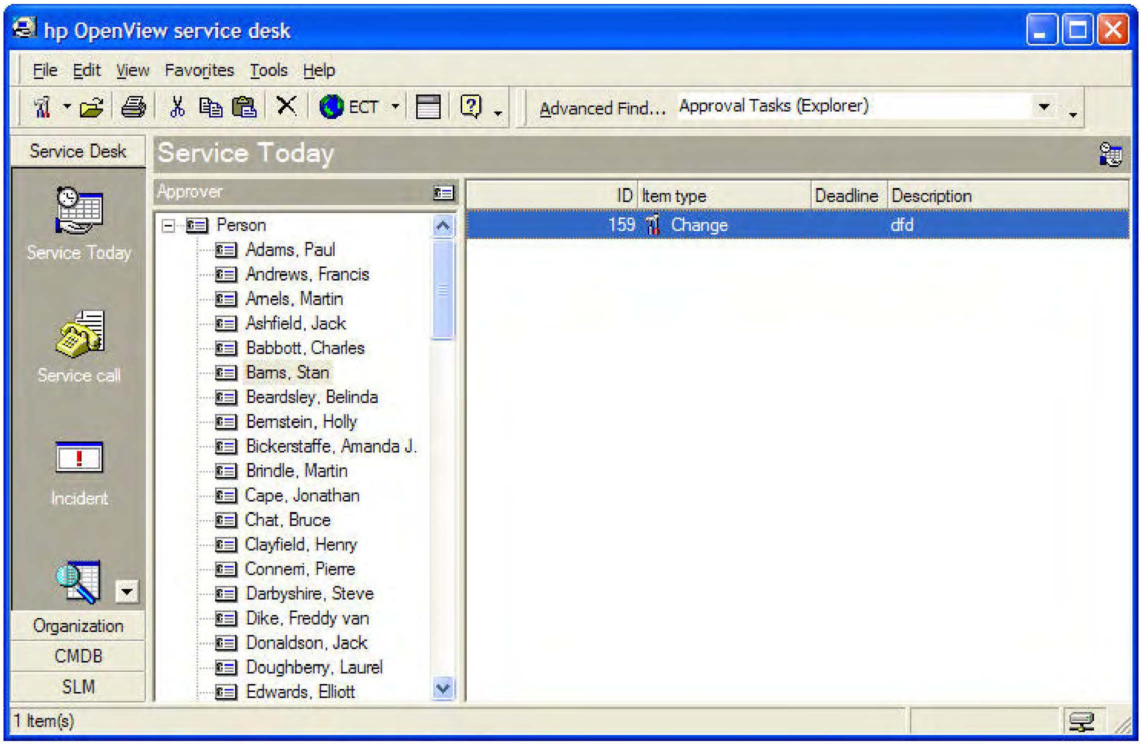


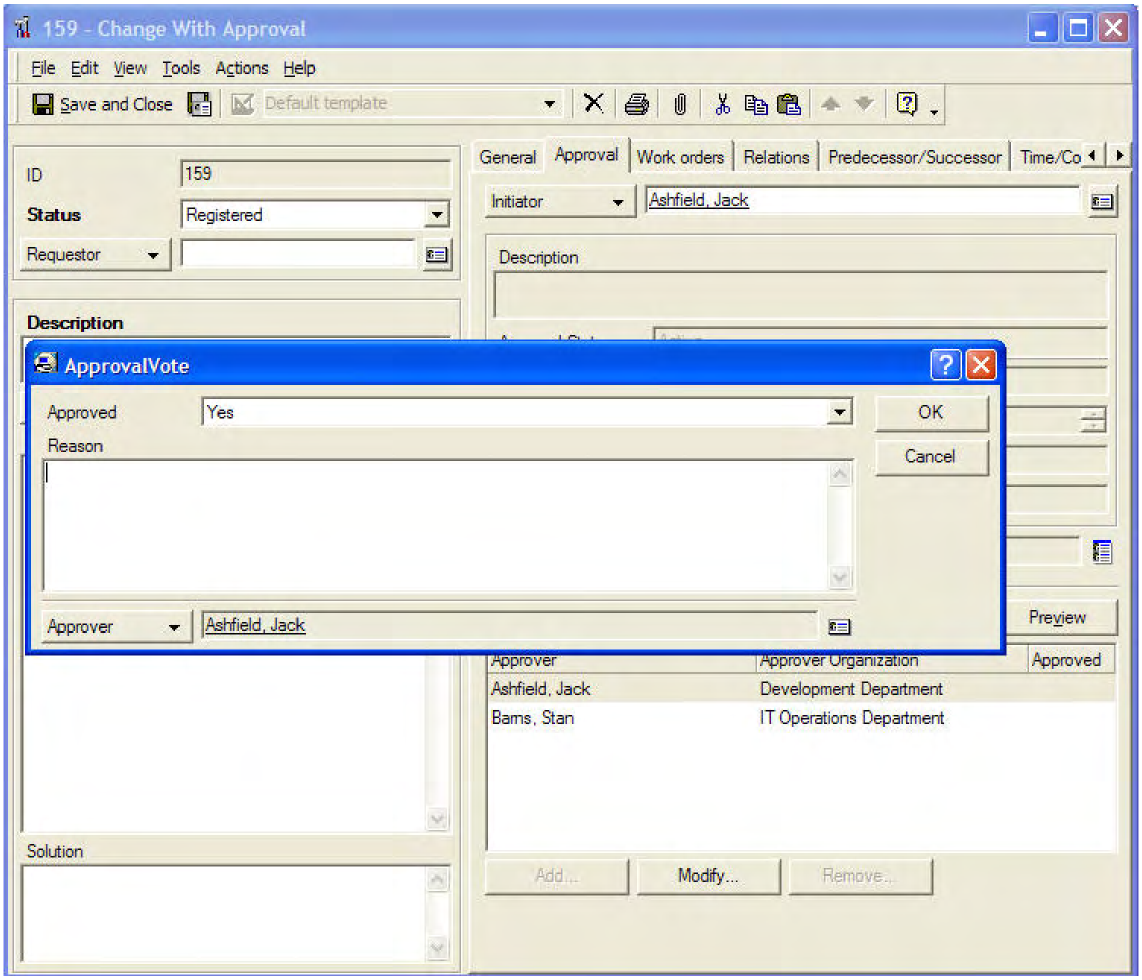
Figure 28 Explorer View for Viewing All Person's Approval Tasks



Deputy Vote Operation

Login as deputy approver, user can see the *Approval Tasks per Person as* shown in Figure 28. Then user may open an item and vote for other person.

Figure 29 Deputy Approval Voting for Others



After end user vote for others, as shown in Figure 29, any deputy voting operation is captured in history line as shown in Figure 30.

The audit function of deputy voting is done by system and can not be enabled or disabled.

Figure 30 History Line for Deputy Approval Voting

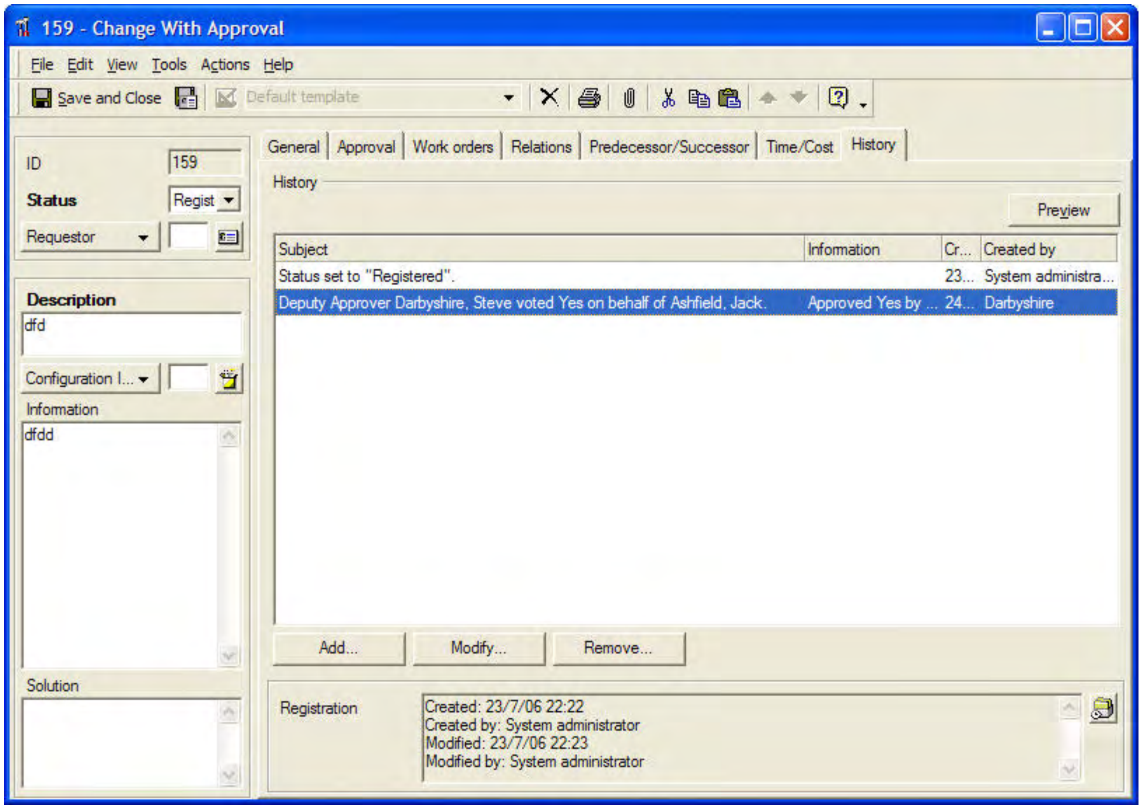
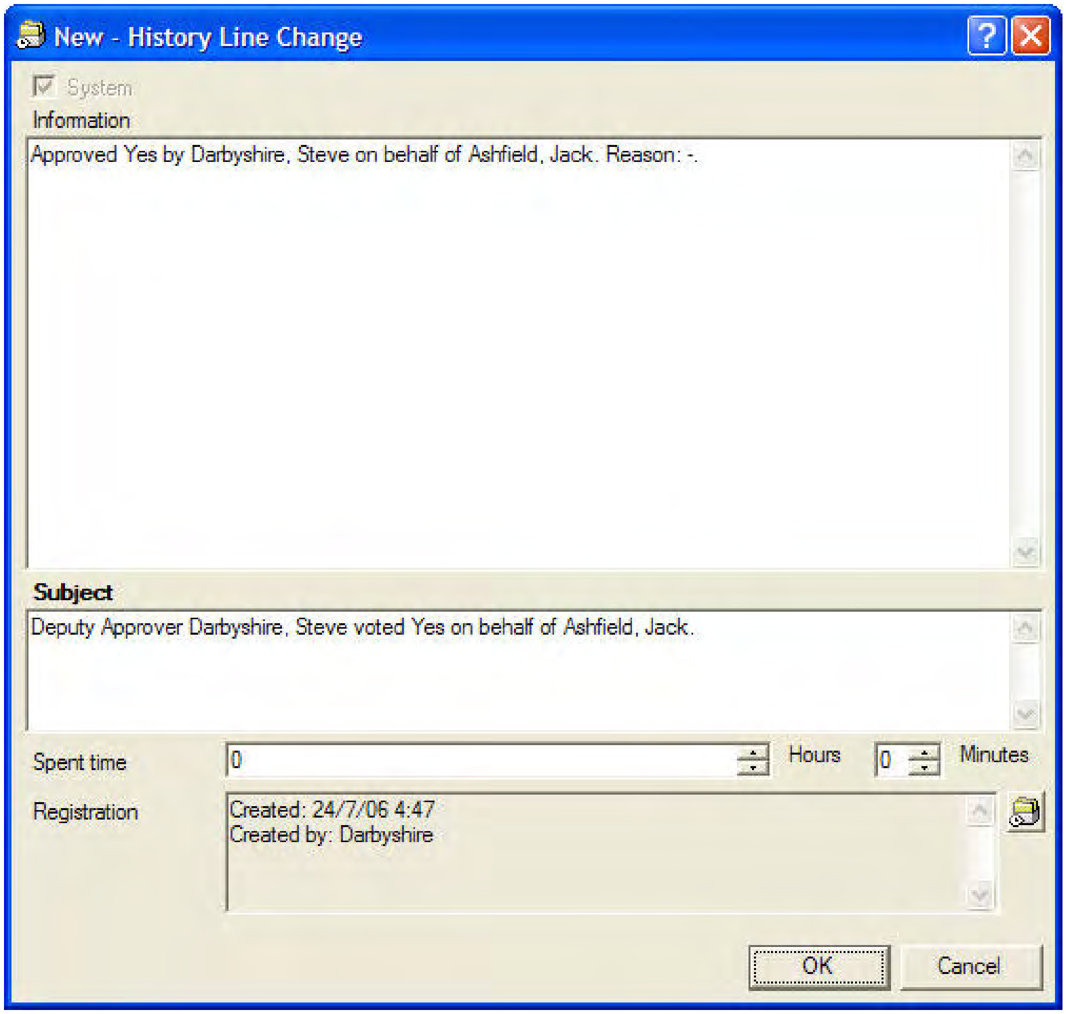


Figure 31 Deputy Approval Voting History Line Content



Pop-up Menu Enhancement

Introduction

Service Desk has offered the solution for expanding out of screen issue on field selection pop-up menus, which is mainly caused by the creation and activation of a large number of custom fields in Service Desk.

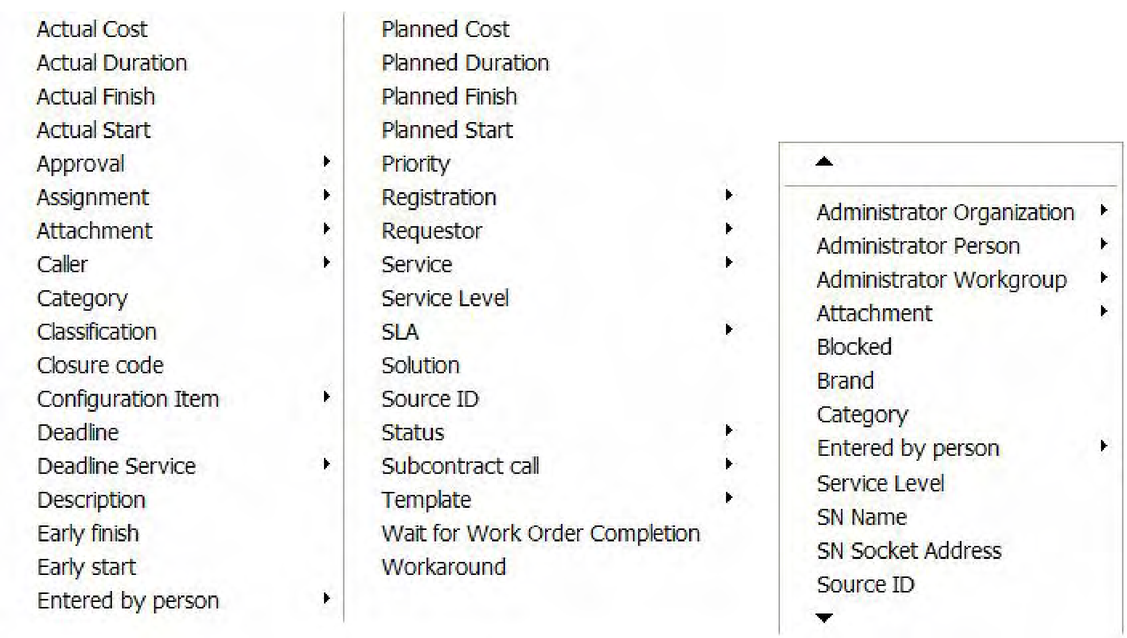
Now user can change Menu Style for field selection popup menu.

Getting Started

Bar Break Style vs Consecutive Style

Windows platform provides two distinct pop-up menu styles, bar break style and consecutive style. As shown below, bar break style (on the left) may contains multiple menu pages, and they are arranged in parallel. This style makes it possible for the end user to view and choose all the fields in a very convenient manner.

Figure 32

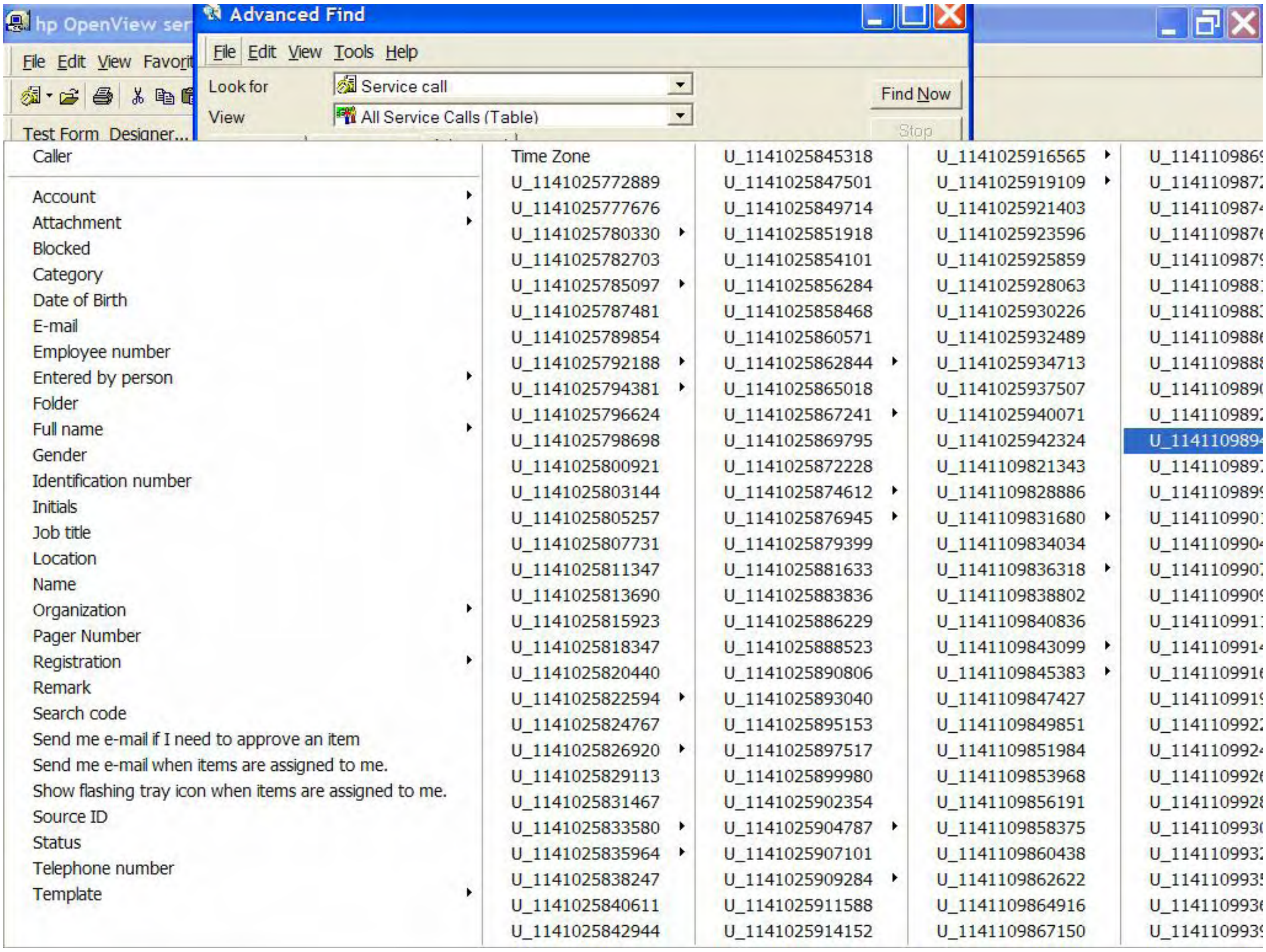


On the contrary, consecutive style (on the right) contain merely one menu page. If the amount of menu items is too large to be displayed completely within the screen, two arrows on the top and bottom will present. By scrolling the menu page up and down, users are still able to view and choose all the fields.

Situation Before SP20

At present, Service Desk 4.5 solely provides the bar break style. Thereby, in some extreme cases, for instances, after creating and activating large amount of custom fields, the bar break style menu pages will expand off the screen boundary as shown below. As a result, some fields are invisible and unselectable.

Figure 33



User Option for Switching Menu Style

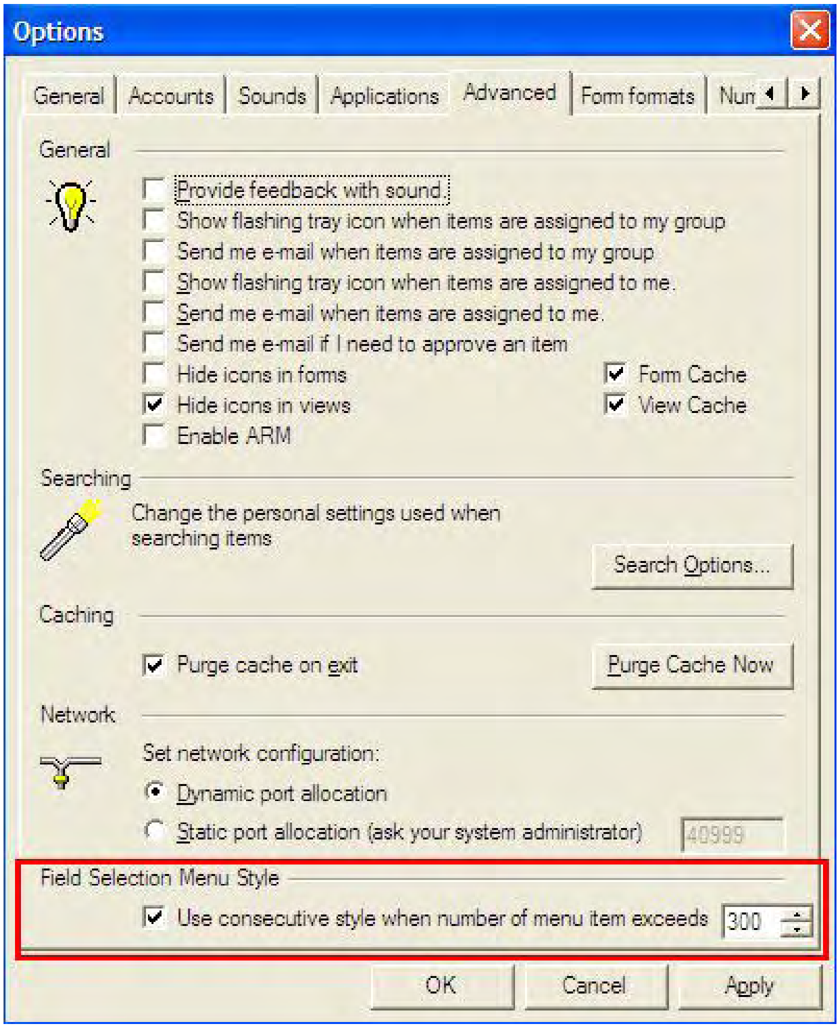
Starting from Service Pack 20, Service Desk provides a new user option "[ ] Use consecutive style when number of menu item exceeds [300]". This provides a flexible way to allow users to customize pop-up menu style according to their desktop screen resolution setting and the custom field number in their Service Desk instances.

- With this option unchecked, bar break style will always be used.
- With this option checked, bar break style will be used when the total number of menu items (exclude sub menu items) is less than the limit number. Otherwise, consecutive style will be used.

> The option will not take effect until the amount of menu items is too large to be displayed completely within the screen.

This option is checked by default, and initial limit number is set to 300.

Figure 34



This setting takes effect on all the occurrence places of the field selection pop-up menu. Field selection pop-up menus are used heavily and extensively in Service Desk. For instance, the pop-up menu on path **View Summary** → **Filter** → **Advanced Tab** → **More Criteria** → **Field**, **UI/DB Rule** → **Condition** → **Add Criterion** → **Field**, and so on.

Custom Codes

Introduction

Since Service Desk SP18, Service Desk allows user to create additional custom fields, but it is not possible to create a code field which refers to an entirely new code type. Custom Codes feature comes as an add-on feature to 'Unlimited Custom Fields' feature and aims at resolving this issue. With this feature, user can introduce new code type and make it being referred by a custom field. This new code type is called *Custom Code*. This document describes how to manage Custom Codes, what you need to take into account when managing Custom Codes and how to use a Custom Code.

The main audience for this document is Service Desk administrators authorized to configure Custom Codes and custom fields.

Managing Custom Codes

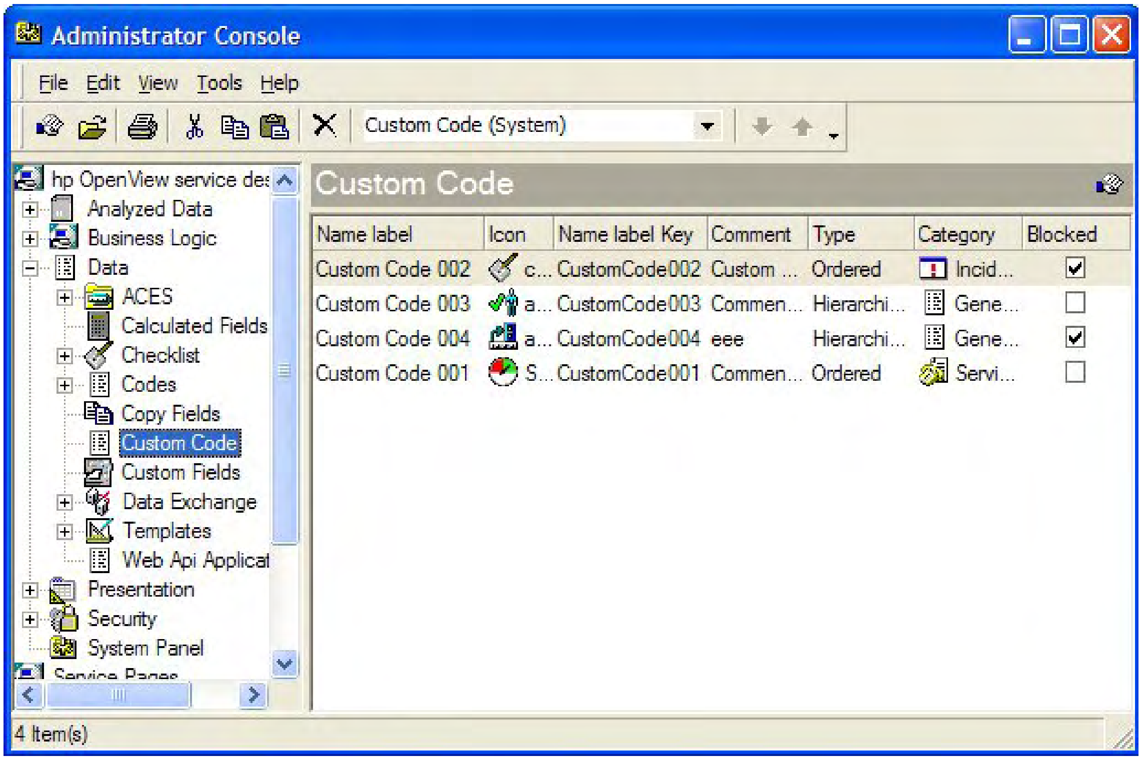
Custom Codes Definition

View of Custom Codes

In Service Desk Administrator Console, a new node 'Custom Code' has been added under *hp OpenView Service Desk* > *Data* node, it is only applicable for a Service Desk system administrator. Click this node, all Custom Codes will be shown in a table view in the right data pane. Three views are currently available for this node. they are

- Custom Code (System) - Custom Code view without grouping (See Figure 35)
- Custom Codes by Category (System) - Custom Code view grouped by Category field (See Figure 36)
- Custom Codes by Type (System) - Custom Code view grouped by Type field (See Figure 37)

Figure 35 Custom Code View 1



By default, the view 'Custom Code (System)' is used.

Table 3 shows the detailed descriptions of all the fields in Custom Code view.

Table 3 Field Descriptions

Field Name	Description
Name Label	The localized Custom Code name
Name Label Key	Key of the name label, it will be used as part of the Web API class name
Comment	Comment of the this Custom Code
Icon	The icon of this Custom Code
Type	Code type, *Ordered* or *Hierarchical*. Code values of an Ordered code can be displayed in a table view in order (e.g. Category), whereas code values of a Hierarchical code can be shown in a tree view (e.g. Priority).
Category	Calculated field. Its value will be the name of a master Item (Service call, Incident and Problem etc.) or *General*, If this Custom Code is **only** referred by an Item, the name of this Item will be the category value of this Custom Code, otherwise, the category value will be *General*.
Blocked	Calculated field, unchecked when at least one custom fields which refer to this Custom Code are activated, or otherwise

Figure 36 Custom Code View 2

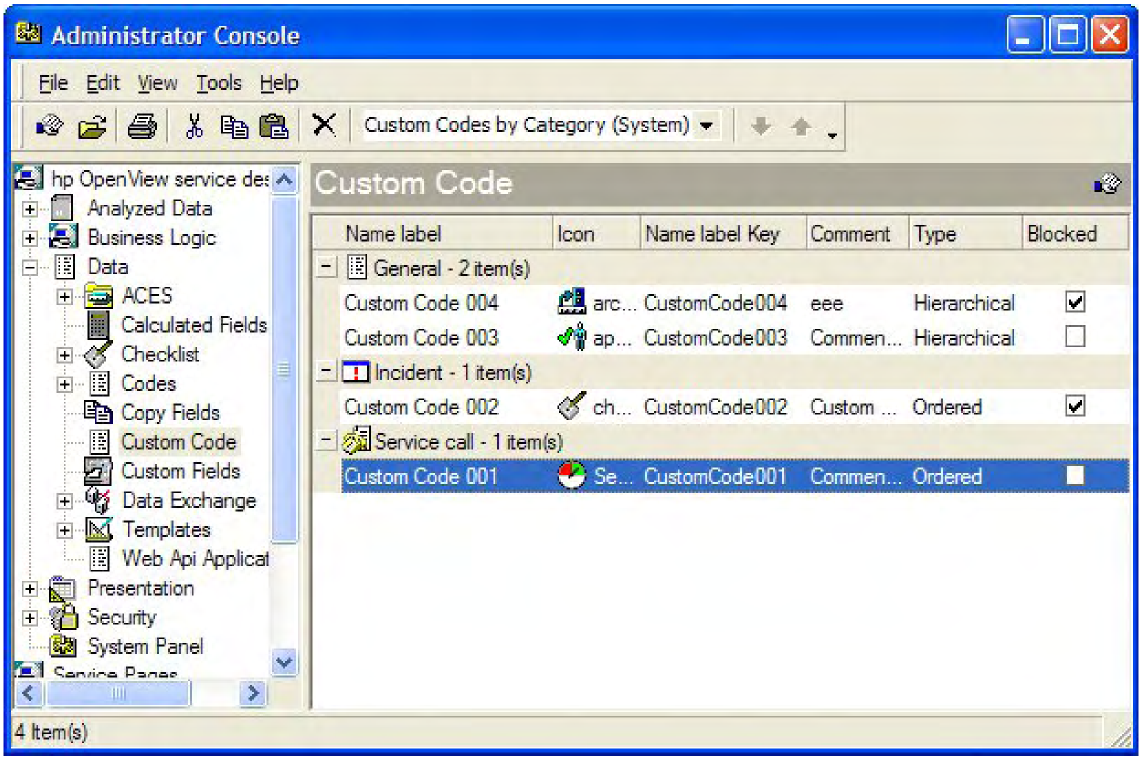
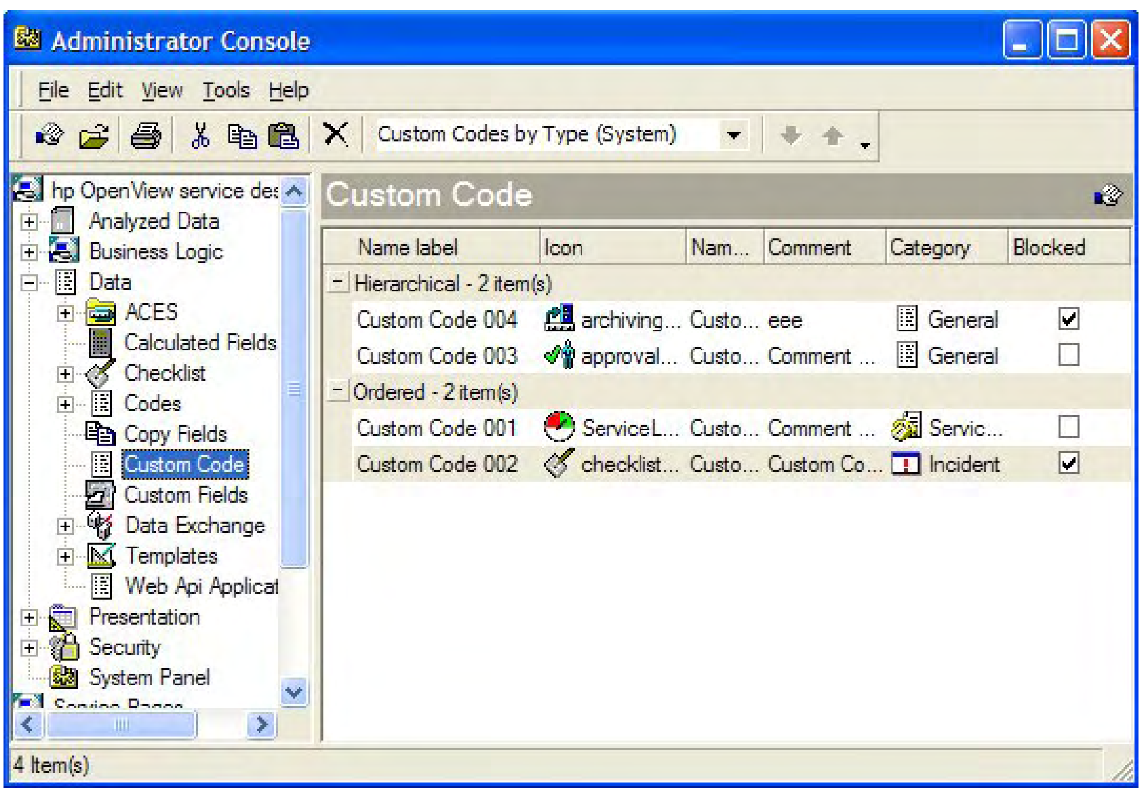


Figure 37 Custom Code View 3



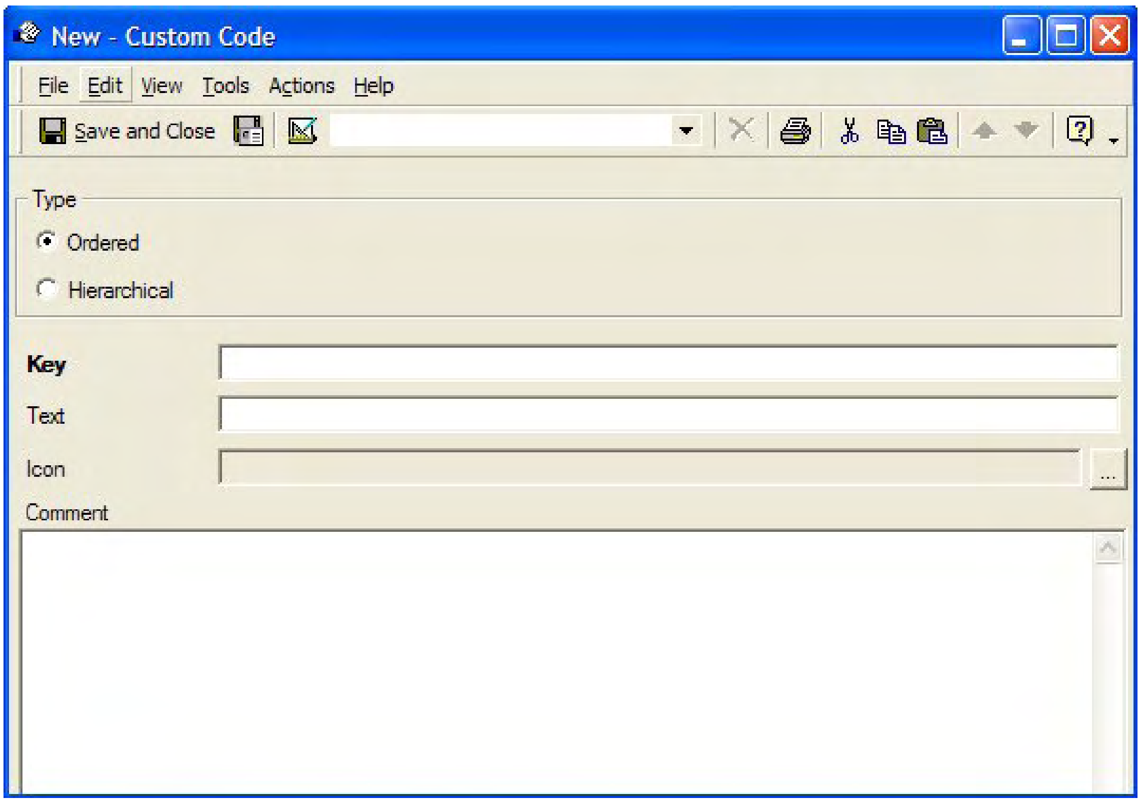
Add a New Custom Code

Open the Administrator Console, and navigate to *Data* > *Custom Code* (see Figure 35), click the **File** → **New** menu item, or select the "New Custom Code" option from the context menu (which you see when you right-click on an item in the view). This opens the New Custom Code data form. In principle, there are two main types of Custom Code

- *Ordered Custom Code* - Code values will be shown in a table view in order.
- *Hierarchical Custom Code* - Code values can be shown in a tree view.

The screenshot below shows new Custom Code data form.

Figure 38



To define a Custom Code, do the following:

1 Select the type.

2 Fill the Key field, it is recommended that user **DOES NOT** use non-Latin text as a part of the Key value. **Note:** Value of Key field should be unique with other entity.

3 Fill Text field which is actually the name of Custom Code.

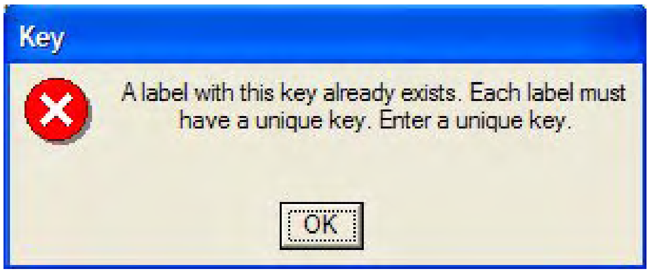
4 Select an Icon (optional), it explores icons in `<SD_INSTALL_DIR>/client/images`, it is recommended that user pick up an icon which is distributed originally with the product. Otherwise, the service desk administrator should distribute this icon to all Service Desk Clients manually, and place it in the same directory.

5 Fill Comment field (optional).

6 Click the **Save and Close** toolbar button to save the Custom Code, or click **Close** window button to cancel the operation. If the Field Key name is already in use, a dialog box (see Figure 39) informs you that the operation has failed.

7 After creating a Custom Code, default value of field "Category" will be "General" which means there is no reference with Custom Filed and the Custom Code is "Blocked" by default.

Figure 39

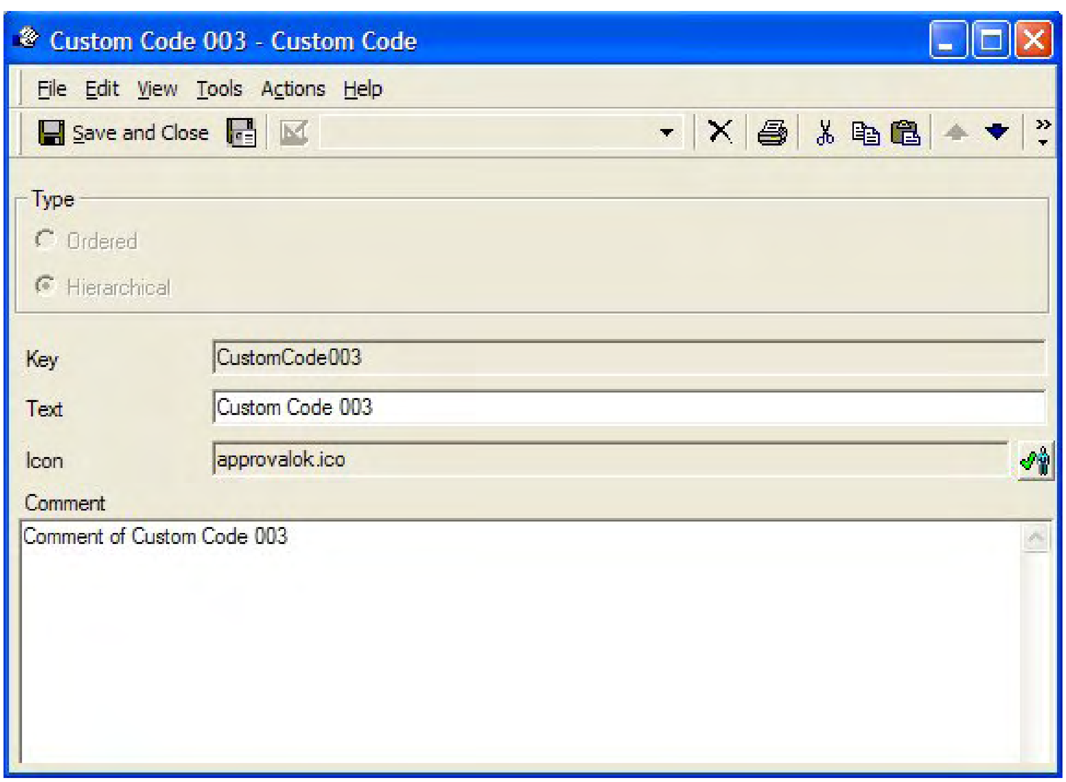


Edit a Custom Code

In Administrator Console, navigate to *Data* > *Custom Code,* select a Custom Code record, click **File** → **Open** menu item, **Open** toolbar button or select the **Open** option from the right-click pop-up context menu. This opens the selected Custom Code. In the opened form (See Figure 40), the *Type* and *Key* fields are shown as read-only, other fields are editable.

User should restart both Service Desk server and client after adding or modifying Custom Code.

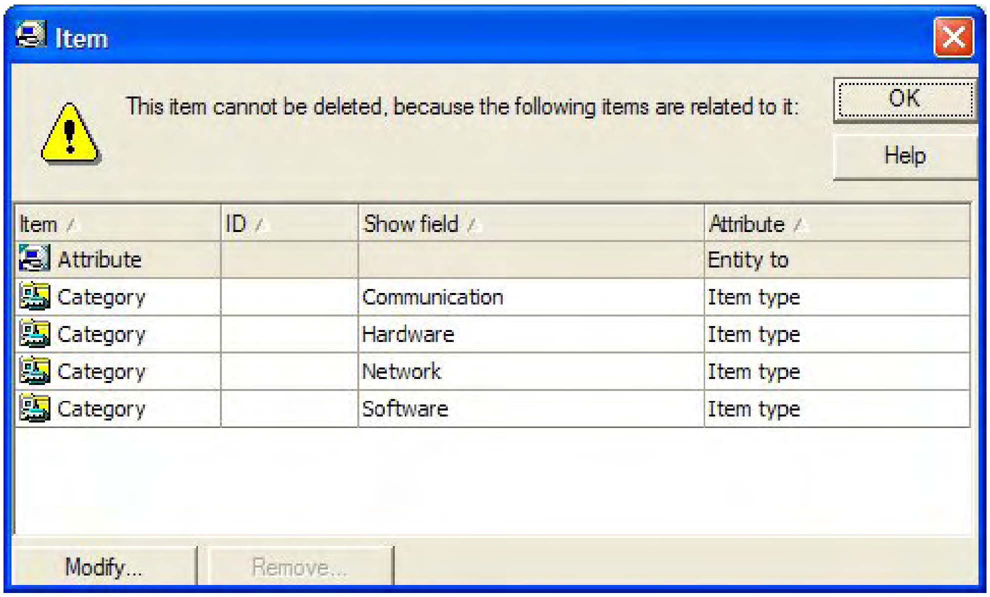
Figure 40



Delete a Custom Code

A Custom Code is removable if it has not been referred by a custom field. To remove a Custom Code, click the **Delete** toolbar button or **Delete** option from the right-click pop-up context menu. If the selected Custom Code is never referred by any custom field, deletion operation will succeed, otherwise, a dialog which reports all the references will display (See Figure 41).

Figure 41



Custom Field Refer to a Custom Code

A Custom Code can be referred by a custom field in any master item and there is no inner logic to control a Custom Code to be only referred by a specific master item. A Service Desk system administrator should be aware of that when introducing a custom field. In the view *Custom Codes by Category (*See Figure 36), if a Custom Code is not placed in *General* category, this Custom Code must has been referred by the Item which has the same name with the category name. It's highly recommended to consider circumspectly about whether this Custom Code can be referred by a custom field before referring it. A simple way to check whether this Custom Code has been referred by a custom field is as follows:

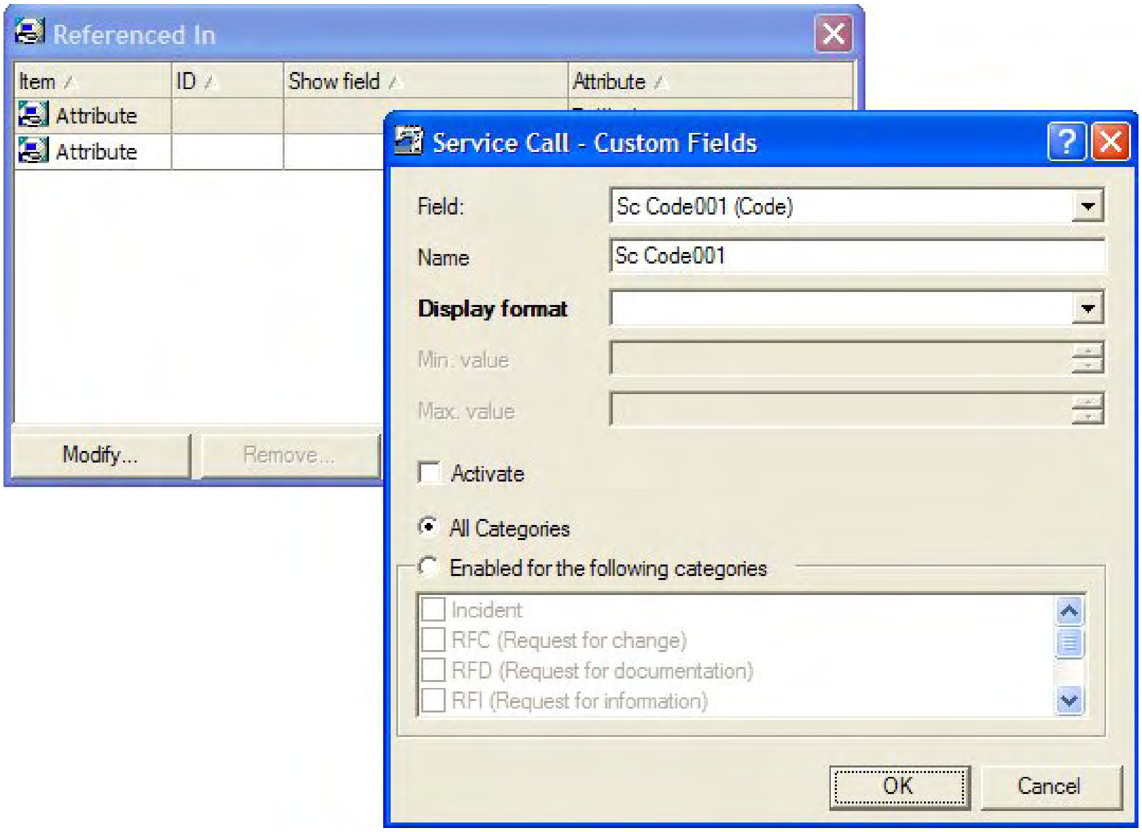
1 Open the Custom Code (See Figure 40)

2 In the opened form, click **Actions** → **References** menu item from the menu bar. This will open a dialog and report all the references (See Figure 42), Open the reference record to see the detailed information.

3 After being referred to a Custom Field, the Custom Code will not be "Blocked". The "Category" value of the Custom Code depends on the number of entity refer to it. If only one entity refers to the code, "Category" is specific entity (e.g. "Service call"); if more than one entity refer to the same code "Category" is "General".

You should refresh the Custom Code View to make update in effect.

Figure 42



Using a custom field which refers to a new Custom Code has no difference with using other custom fields. The functionality of activating or deactivating a custom field remains unchanged. The newly introduced code field can be used in Form, View, Templates, Advanced Find, DB Rule and UI Rule etc.

Manage Custom Code Values

Code values of a Custom Code become operable only after this Custom Code is referred by at least one active custom field. There is no change for functionality of managing Custom Code values, code values of a Custom Code and an existing Code are both manipulated under the node **hp OpenView service desk** → **Data** → **Codes**. To maintain Custom Code values, do the following:

1. First, navigate to **hp OpenView service desk** → **Data** → **Custom Code** node, switch to *Custom Code by category* view to see if the Custom Code is not blocked, which category it belongs to.
2. For a non-blocked Custom Code, according to which category it belongs to, a new node name by the name of this Custom Code will be shown under **hp OpenView service desk** → **Data** → **Codes** → `<Category Name>`.
3. Enter this node, user can manage code values of this Custom Code.

You should refresh the Custom Code value View to make update in effect.

Enhanced Auditing for Custom Code

In principle, Enhanced Auditing for Custom Code includes two parts

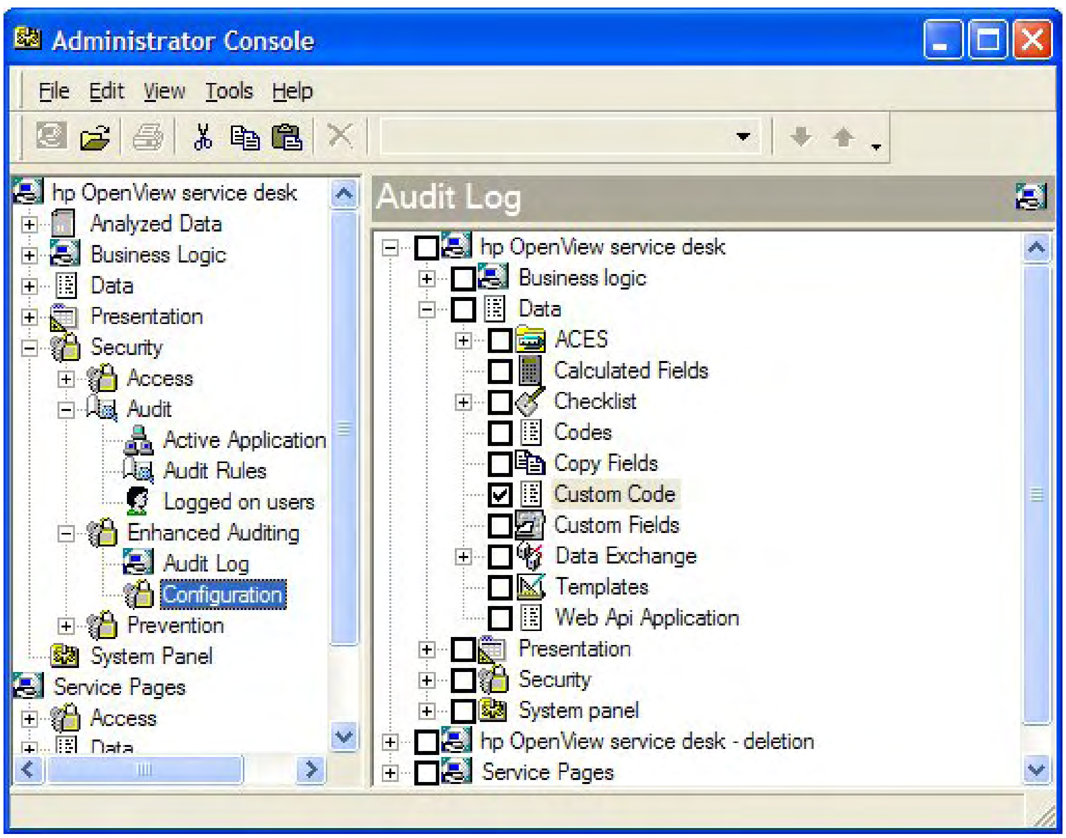
1. Audit Custom Code
2. Audit Custom Code values

Audit Custom Code

The manipulation of Custom Code can be audited with Enhanced Auditing. To audit Custom Code, do the following:

1. In Administrator Console, navigate to **Security** → **Enhanced Auditing** → **Configuration**, right click *hp OpenView service desk* node in the right tree view data pane and click **Resume** on the pop-up menu.
2. Expand the *hp OpenView service desk* node and navigate to **Data** → **Custom Code**, enable auditing (see Figure 43).

Figure 43



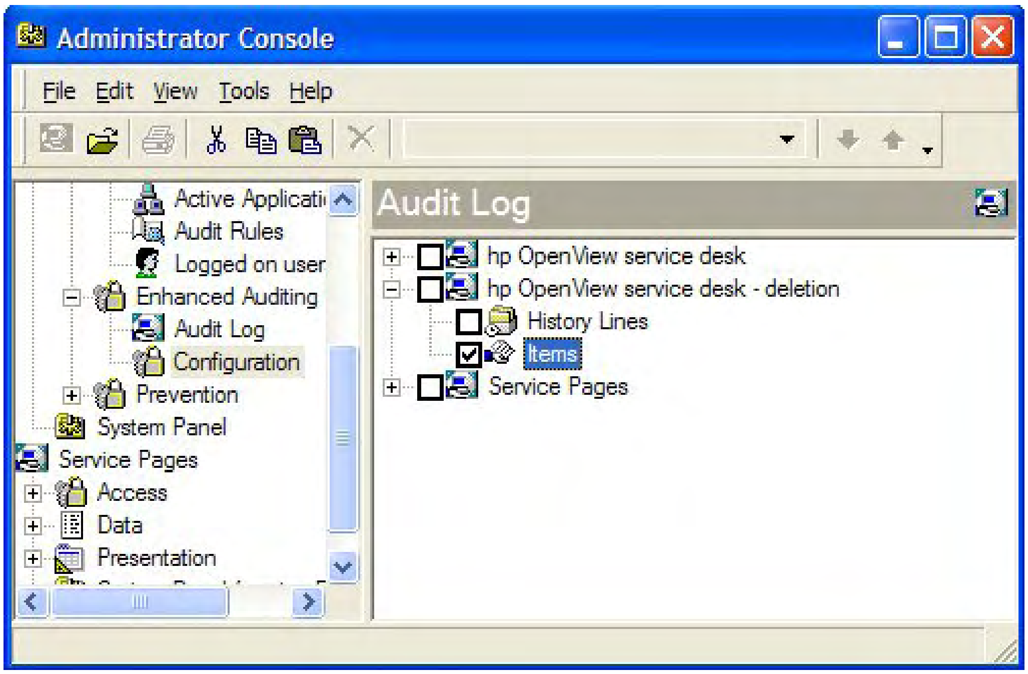
Then the operation of Custom Code creating, editing and **deleting** will be audited.

The way this feature deal with auditing delete operation is different from other parts. For other parts, auditing delete operation is managed by the audit configuration node **hp OpenView service desk - deletion** → **Items**.

Audit Custom Code Values

Custom Code values auditing is the same as the auditing of code values of existing Codes. Enable auditing for **hp OpenView service desk** → **Data** → **Codes**, then the operation of creating and modifying Custom Code values will be audited, but deleting a Custom Code value will not be audited until the audit configuration node **hp OpenView service desk - deletion** → **Items** (SeeFigure 44) is checked.

Figure 44



Custom Code ACES

Since Service Desk SP18, a Service Desk system administrator is capable of migrating custom fields from staging to a production environment. If a custom field happens to be referred a Custom Code, the ACES importing processing for this custom field must fail before the Custom Code is imported. Custom Code ACES is provided for transferring Custom Codes between two Service Desk instances with the same version.

The exported XML only includes definition of Custom Code and code values of Custom Code are not included, ACES for code values of custom codes behaves without any difference from it does for values of original codes. When performing ACES exporting, user should separate the task into three groups: Custom Code, Custom Field and Custom Code value.

When importing Custom Codes using ACES, make sure that the option "*Overwrite Existing Items*" is checked if you want these existing Custom Codes to be updated. Note that overriding existing items behaves the same as it does in custom field ACES.

To avoid problems that come from missing Custom Codes when processing custom field ACES importing, user **MUST** process Custom Code ACES importing prior to process custom field ACES importing.

Custom Code ACES importing changes the Object Model of Service Desk, this is as same as it does in custom field ACES importing.

Custom Code ACES importing, which means that restarting both Service Desk server and client is required if user want to import Custom Code values right after importing the Custom Codes by ACES.

Localization

I18N (Internationalization) of Custom Codes feature includes two parts, one is internationalization for Custom Code name and the other is for Custom Code values. They are done in **hp OpenView service desk** → **Presentation** → **Localized Text** → **Label Text** and **hp OpenView service desk** → **Presentation** → **Localized Text** → **Kernel Code Text** respectively.

Web-API

To access newly added custom code using the Web API, Java developers need to regenerate the Web API, compile it, and replace the original Web API jar file with the new Web API jar file in their Java applications. This function has been implemented in ITSM008292. Please refer to *ITSM008292* for more information.

Technical Information

Object model changed. In current Service Desk, the Code Items referred by pre-defined custom fields are all inherited from Code ordered which is defined in ITSM module, but all new custom codes are indirectly inherited from Kernel Code.

No table added. It does not need to create new data mapping after a Custom Code created. Whereas code values of new custom codes will be stored in table REP_CODES and REP_CODES_TEXT.

ESMTP Authentication for Outbound Mail

Introduction

When a mail server does not require clients to authenticate themselves, it becomes vulnerable to misuse. For this reason, many mail servers require authentication nowadays. The SMTP authentication for outbound mail feature makes it possible for Service Desk to provide authentication information to an external mail server that Service Desk uses to send mail.

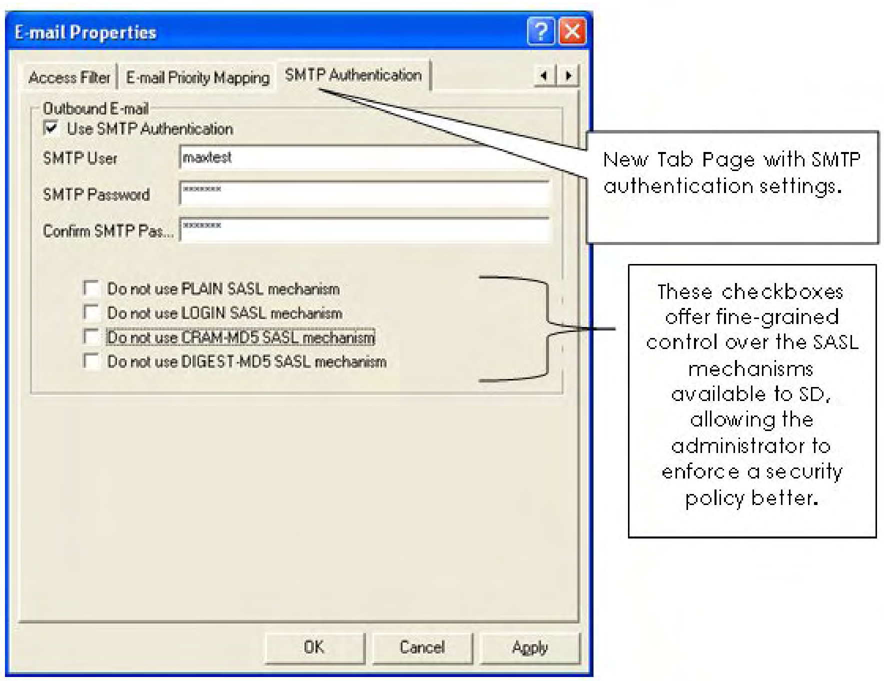
This enhancement is sometimes also referred to as ESMTP authentication. ESMTP defines a way to add extra commands to SMTP, the protocol used for sending mail. Authentication is a command extension. However, in this document, we will simply refer to this enhancement as SMTP authentication. This is also how it is called in Service Desk itself.

Getting Started

Configuring Authentication Outbound Mail

To configure authentication for outbound mail, a new tab page has been added to the e-mail configuration options. These can be found in the Administrator Console, under System Panel, E-mail. On this page, authentication for outbound mail can be turned on by checking the "Use SMTP Authentication" checkbox. An example of this new tab page is shown below:

Figure 45



When the checkbox "Use SMTP Authentication" is ticked, all attempts to send outbound mail from Service Desk will be done using the user and password specified on this page. When the checkbox "Use SMTP Authentication" is disabled, all the fields on this page will be grayed out. Any settings entered previously will be maintained but unused.

The checkboxes underneath the password field allow an administrator to turn off the use of specific SASL mechanisms. A SASL mechanism is an implementation of an authentication algorithm for a mail client. As can be deducted from the screenshot, Service Desk supports four authentication mechanisms for outbound mail: PLAIN, LOGIN, CRAM-MD5 and DIGEST-MD5.

Under normal circumstances, the administrator will not have to disable any SASL mechanism at all. In the same way as most mail clients do, Service Desk uses the first authentication mechanism that the mail server reports to support and that it also supports itself. This way, the mail client and mail server automatically negotiate which authentication mechanism will be used.

However, the administrator may choose to disable certain mechanisms to comply with company policies. For example, the authentication information sent through the PLAIN and LOGIN methods can be intercepted by network sniffing. For this reason, a company may decide to disallow use of those authentication methods. Another reason for the administrator to disable selected SASL mechanisms is to troubleshoot authentication problems.

It is not possible to disable all the SASL mechanisms by checking all of the four checkboxes for the mechanisms: at least one must be selected at any time. Using none of the SASL mechanisms is effectively the same as not using SMTP authentication. The "Use SMTP Authentication" checkbox should be cleared to configure Service Desk to stop using this enhancement.

DIGEST-MD5 authentication requires not only a username and a password but also a realm. The realm can be entered as part of the username by placing a @-sign behind the username. The part behind the @-sign is used as the realm and everything before it as username.

Testing the Outbound Mail Authentication Settings

To test the authentication settings, you can use the "Send test mail..." button, which is on the General tab page of the E-mail Properties dialog.

Places where Outbound Mail is Sent in Service Desk

Service Desk sends mail when any of the following applies:

- Service Desk executes a database rule that includes a "Send e-mail message" action.
- The current user in the Service Desk client is configured to "Send me e-mail when items are assigned to me" or "Send me e-mail when items are assigned to my group". You can turn on these settings in the client from the Tools menu, under Options, Advanced.
- The current user in the Service Desk client is configured to "Send me an e-mail if I need to approve an item." This setting is also enabled from the Tools menu, under Options, Advanced.
- The current user in the Service Desk client adds an e-mail to a Service Call.

User Notification Upon Opening or Editing the Same Record

Introduction

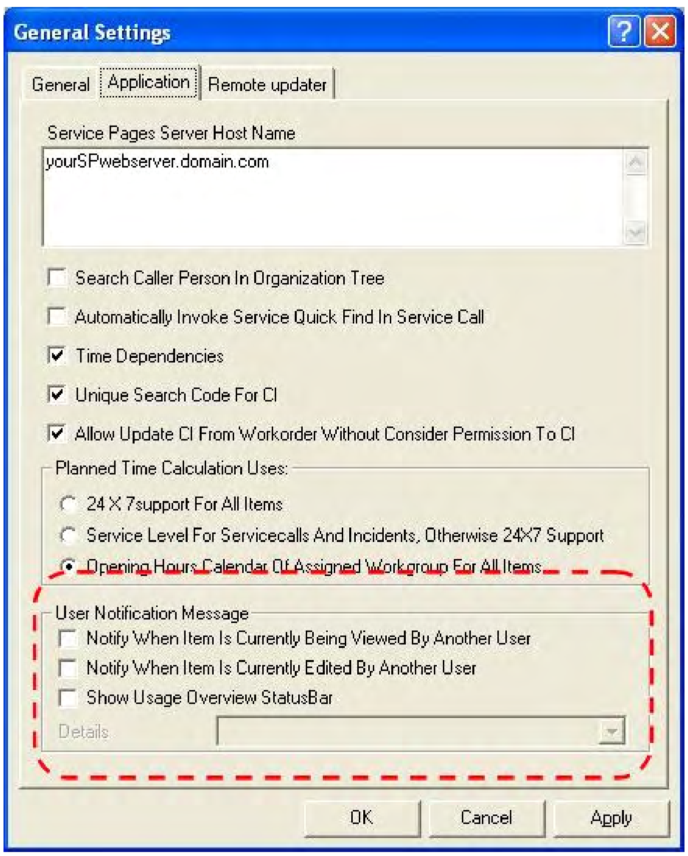
Since Service Desk SP20, Service Desk allows user to be notified whenever they open a Service Desk record that has already been opened or edited at the same moment by other Service Desk user(s).

The main audience for this document is Service Desk users and administrators authorized to configure user notification options.

New User Notification Options

In the General Settings of Administrator Console, several options have been added to configure this feature:

Figure 46 Added Options in General Settings



User Notification on Viewing

Figure 47

Whenever it is required to be notified in case another *Service Desk User* is just *viewing* the same *Service Desk record*, this checkbox needs to be selected.

When switched *on*, every time any *Service Desk User* tries to open a *Service Desk record* (e.g. a *Work order*, *Service* or a *Configuration Item*) that at that very moment is being *viewed only* by any other *Service Desk User*, *Service Desk User* will be notified.

User Notification on Editing

Figure 48

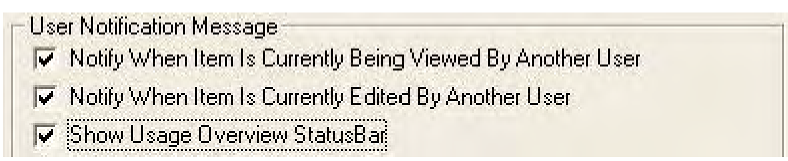
Whenever it is required to be notified in case another *Service Desk User* is *editing* the same *Service Desk record*, this checkbox needs to be selected.

When switched *on*, every time any *Service Desk User* tries to open a *Service Desk record* (e.g. a *Work order*, *Service* or a *Configuration Item*) that at that very moment is being *modified* by any other *Service Desk User*, *Service Desk User* will be notified.

In case both of these checkboxes have been switched on, **only User Notification on Editing** will be taken into effect.

Usage Overview StatusBar

Figure 49

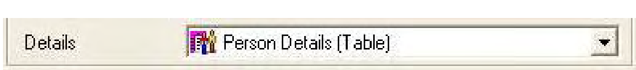


With this checkbox selected, the *Status Bar* can be added in *Service Desk Items Form* to provide information for this particular *Service Desk record*. (please refer to 4 Usage Overview StatusBar)

In case all aforementioned checkboxes are switched *off*, *Service Desk* will behave exactly the same as it did before this feature has been introduced; the default behavior therefore remains the same.

(Person) Details

Figure 50



When this is filled with a certain *Service Desk Person Table View*, every time a user notification is popped-up (either on *editing Service Desk Users*, or on *viewing Service Desk Users*), it will be accompanied by a *list* of all *Service Desk Users* that actually view/modify that *Service Desk record* at that very moment.

This list contains the information specified in the definition of selected *View*. (please refer to 3 Person Details)

When this is left empty, no detailed list will be shown.

User Notification

When either one of the notification settings has been switched on, every time an *Service Desk User* tries to open an *Service Desk record* that is being viewed or edited by an (or some) *Service Desk User(s)* he will be notified:

Figure 51 User Notification on Subsequent Opening (Viewing)

Or

Figure 52 User Notification on Subsequent Opening (Editing)

Whenever pressing 'Ok', this Service Desk User will continue working on this particular *Service Desk record*. If 'Cancel'-ed, the *Service Desk record* will not be opened.

This warning will also be presented after a refresh of the *Service Desk record* (as this is considered to be a logical re-opening of the particular *Service Desk record*).

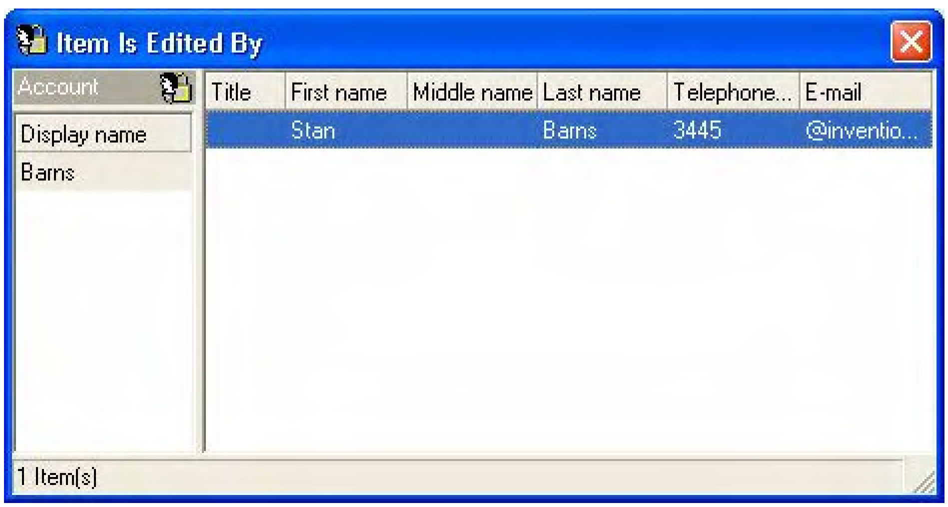
Person Details

Whenever Details has been configured (as explained in 1.4 Person Details), the *User notification* will be accompanied by a detailed *list* of all *Service Desk User* that are currently viewing or editing this *Service Desk record*:

Figure 53 **Person's Details of User Notification (Opened By)**

Or

Figure 54 **Person's Details of User Notification (Edited By)**



Like this, firstly required *highlight Person contact info details* (*Name*, *Telephone* number, and *E-mail* address e.g.) are available at a glance.

Other *Person* data can be retrieved by simply opening any of the *Persons* presented in this list. This will result in the opening of the *Service Desk Person record* referred to.

Because it is possible to have *Accounts* without *Person* details assigned to it in Service Desk, a list of *Accounts is added on the left* to overcome invisibility in such cases.

Usage Overview StatusBar

Figure 55 **Status Bar Extension**

It is possible to retrieve (upon pressing the ‘Update’ button there) overall information on how this particular *Service Desk record* is being used throughout the entire Service Desk environment; it tells:

1. ‘<n> Open’: 

How many (other) *Service Desk Users* are only *viewing* this *Service Desk record*.

2. ‘<m> Edit’: 

How many (other) *Service Desk Users* are currently *modifying* this *Service Desk record*.

> This can be used to clew the possible occurrence of merge dialogs when saving the record.

3. ‘Saved’: 

   Whether this Service Desk record has been saved.

> This can be used to clew the certain occurrence of merge dialogs when saving the record.

Double clicking either ‘<n> Open’ or ‘<m> Edit’ whenever the number is not zero (‘0’) results in the presentation of a list, similar to the one mentioned in 3 Person Details, but with corresponding list of either all *viewers* (whenever the ‘Open’ is clicked) or *modifiers* (whenever the ‘Edit’ is clicked).

Current user's actions causing the status of the Service Desk record to change (e.g. starting to edit it) will be updated in Usage Overview Status Bar automatically.

Other users' actions causing the status of the Service Desk record to change (e.g. starting to edit it) will be updated by pressing the ‘Update’ button there.

> In case the Application Server crash, the Usage Overview StatusBar will not display any information about the usage of current record, show as below:

Known Errors and Limitations

1. This feature DOES NOT support notification for actions (opening or editing) made through following ways:

- Web-Console
- Web-API
- Service page
- Inbound email
- Data Exchange (sd_event)
- Database rule

2. When opening a record twice (or more than twice) in the same client, the count of Open and Edit in status bar will be messed up.

Scenario:

1 Create a work order.

2 Create a service call.

3 Open the newly created service call.

4 In work order, relate the service call to work order and save the work order.

5 Open the service call from the work order (there will be 2 forms for the same service call, also you can open more forms for the same service call).

6 Change something in one of them. See the status bar.

The count of Open and Edit for this service call will mess up. Sometimes 1 Open 0 Edit, sometimes 0 Open 1 Edit.

3. The count of Open and Edit in status bar will be incorrect when session time out occurs.

Scenario 1:

1 Open a record. In status bar, there will be 1 Open 0 Edit.

2 Wait session timeout then edit. In status bar, there will be 0 Open 0 Edit.

Scenario 2:

1 Open a record and change something. In status bar, there will be 0 Open 1 Edit.

2 Wait for session timeout then update status bar, there will be 0 Open 0 Edit.

invent

Java™
COMPATIBLE