

# **HP OpenView Service Navigator Value Pack**

## **Integrating Service Desk with Service Configuration for Service Navigator**

**Software Version: 9.0**

**for the HP-UX, Microsoft Windows, and Sun Solaris Operating Systems**



**Manufacturing Part Number: None**

**Document Release Date: May 2006**

**Software Release Date: May 2006**

© Copyright 2006 Hewlett-Packard Development Company, L.P.

---

## Legal Notices

### **Warranty.**

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

The information contained herein is subject to change without notice.

### **Restricted Rights Legend.**

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

### **Copyright Notices.**

©Copyright 2003-2006 Hewlett-Packard Development Company, L.P.

### **Trademark Notices.**

Adobe®, Acrobat® are trademarks of Adobe Systems Incorporated.

HP-UX Release 10.20 and later and HP-UX Release 11.00 and later (in both 32 and 64-bit configurations) on all HP 9000 computers are Open Group UNIX 95 branded products.

Intel®Celeron® Intel and Celeron are US registered trademarks of Intel Corporation.

Intel Inside Logo w/ MMX The Intel Inside Logo & Pentium are US registered trademarks and MMX is a US trademark of Intel Corporation.

Intel Itanium® Logo, Intel, Intel Inside and Itanium are trademarks or registered trademarks of Intel Corporation in the U.S. and other countries and are used under license.

Intel® Itanium®Processor Family is a trademark of Intel Corporation in the US and other countries and is used under license.

Microsoft® is a U.S. registered trademark of Microsoft Corporation.

MMX™ is a US trademark of Intel Corporation.

MS-DOS® is a U.S. registered trademark of Microsoft Corporation.

OpenView® is a registered U.S. trademark of Hewlett-Packard Company.

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

OSF, OSF/1, OSF/Motif, Motif, and Open Software Foundation are trademarks of the Open Software Foundation in the U.S. and other countries.

Pentium® is a U.S. registered trademark of Intel Corporation.

SQL\*Net® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

SQL\*Plus® is a registered U.S. trademark of Oracle Corporation, Redwood City, California.

UNIX® is a registered trademark of the Open Group.

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

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

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



**1. About the Service Desk Integration**

|   |    |
|---|----|
| About Service Desk Objects in Service Hierarchies . . . . .                 | 14 |
| About Service Hierarchy Rules . . . . .                                     | 15 |
| About Child Object Rules . . . . .  | 17 |
| About Parent-and-child Object Rules . . . . .                               | 19 |
| About the Example Service Hierarchy Rules . . . . .                         | 21 |
| Example Hierarchy “Service Desk Services and Configuration Items” . . . . . | 22 |
| Example Hierarchy “Service Desk Organizations” . . . . .                    | 24 |
| Importing the Example Service Hierarchy Rules . . . . .                     | 26 |

**2. Creating Service Hierarchies with Service Desk**

|   |    |
|---|----|
| Working with Service Hierarchy Rules . . . . .                  | 28 |
| Managing Child Object Rules . . . . .                           | 30 |
| Viewing Available Child Object Rules . . . . .                  | 31 |
| Creating a New Child Object Rule . . . . .                      | 32 |
| Editing a Child Object Rule . . . . .                           | 33 |
| Deleting a Child Object Rule . . . . .                          | 34 |
| Before You Delete a Child Object Rule . . . . .                 | 34 |
| Managing Parent-and-child Object Rules . . . . .                | 35 |
| Viewing Available Parent-and-child Object Rules . . . . .       | 36 |
| Creating a New Parent-and-child Object Rule . . . . .           | 37 |
| Editing a Parent-and-child Object Rule . . . . .                | 39 |
| Deleting a Parent-and-child Object Rule . . . . .               | 40 |
| Before You Delete a Parent-and-child Object Rule . . . . .      | 40 |
| Specifying Filters for Service Hierarchy Rules . . . . .        | 41 |
| Filtering Objects by Attributes . . . . .                       | 42 |
| Filtering Objects by Associations . . . . .                     | 44 |
| Setting Default Properties . . . . .                            | 46 |
| Associating Service Hierarchy Rules . . . . .                   | 49 |
| Associating Child Object Rules . . . . .                        | 50 |
| Associating Parent-and-child Object Rules . . . . .             | 52 |
| Applying Service Hierarchy Rules . . . . .                      | 54 |
| Monitoring the Application of Service Hierarchy Rules . . . . . | 55 |
| Stopping the Application of Service Hierarchy Rules . . . . .   | 55 |
| Adding Rules Using the Service Hierarchy Wizard . . . . .       | 56 |
| Adding Parent-and-child Object Rules . . . . .                  | 57 |
| Viewing Service Hierarchy Properties . . . . .                  | 59 |
| Manually Adding Service Desk Objects as Child Objects . . . . . | 60 |

---

# Contents

|   |    |
|---|----|
| Adding the Same Service Desk Object More Than Once to the Same Service Hierarchy. . . . . | 62 |
| Adding the Same Service Desk Object to Different Service Hierarchies. . . . .             | 63 |
| Mapping Objects to Service Desk Objects . . . . .   | 64 |
| Before You Map to a Service Desk Object . . . . .   | 66 |
| Before You Change a Service Desk Mapping . . . . .  | 66 |
| Before You Remove a Service Desk Mapping . . . . .  | 67 |

## 3. Service Desk Integration for the OVO Operator

|  |    |
|--|----|
| About the Service Desk Integration for the OVO Operator. . . . . | 70 |
| Service Pages Integration for the OVO Operator. . . . .          | 72 |
| About the Service Pages Integration . . . . .                    | 73 |
| How the Service Pages Integration Works . . . . .                | 75 |
| Configuring the Service Pages Integration . . . . .              | 76 |
| Automating the Service Pages Authentication . . . . .            | 78 |
| Troubleshooting the Service Pages Integration . . . . .          | 79 |
| Service Desk Data Form Integration for the OVO Operator. . . . . | 81 |
| About the Service Desk Data Form Integration . . . . .           | 82 |
| How the Data Form Integration Works . . . . .                    | 84 |
| Configuring the Data Form Integration . . . . .                  | 85 |
| Troubleshooting the Data Form Integration. . . . .               | 86 |
| Web Console Integration for the OVO Operator. . . . .            | 87 |
| About the Web Console Integration . . . . .                      | 88 |
| How the Web Console Integration Works . . . . .                  | 91 |
| Configuring the Web Console Integration. . . . .                 | 92 |

|                           |           |
|---------------------------|-----------|
| <b>Glossary . . . . .</b> | <b>95</b> |
|---------------------------|-----------|

|                        |            |
|------------------------|------------|
| <b>Index . . . . .</b> | <b>107</b> |
|------------------------|------------|

---

## Documentation Updates

This manual's title page contains the following identifying information:

- 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/](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.

Table 1 indicates changes made to this document since the last released edition.

**Table 1**                      **Document Changes**

| <b>Chapter</b>  | <b>Changes</b>  |
|---|---|
| Chapter 2, "Creating Service Hierarchies with Service Desk," on page 27 | Added URL support for service icons and backgrounds, and graphics location directories:<br><br><install_dir>\www\images\sntp\icons<br><install_dir>\www\images\sntp\backgrounds<br><install_dir>\www\images\sntp\ui |





---

## 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 log in. Many also require a support contract.

To find more information about access levels, go to:

**[http://www.hp.com/managementsoftware/access\\_level](http://www.hp.com/managementsoftware/access_level)**

To register for an HP Passport ID, go to:

**<http://www.managementsoftware.hp.com/ \ passport-registration.html>**





Service Desk and Service Configuration work together in the following main areas:

❑ **Shared data repository—shared objects**

When running Service Configuration in a Service Desk environment, both applications store their data in a shared repository. This means that Service Desk objects can be accessed from within Service Configuration.

See “About Service Desk Objects in Service Hierarchies” on page 14 for more information.

❑ **Service Pages integration for the OVO operator**

Service Pages are secure Internet pages that enable Service Desk customers and specialists to report problems and access data over the Internet. The Service Pages application is integrated in the Service Navigator operator console so that operators can access Service Desk data in the context of a specific service or configuration item.

See “Service Pages Integration for the OVO Operator” on page 72 for more information.

❑ **Service Desk data form integration for the OVO operator**

Forms in Service Desk are used to show the detailed information of objects. The Service Desk client console is integrated in the Service Navigator operator console so that operators can open a Service Desk data form for a specific Service Desk object selected in Service Navigator.

See “Service Desk Data Form Integration for the OVO Operator” on page 81 for more information.

❑ **Web Console integration**

With the Web Console integration enabled, operators using the Service Navigator operator console can open a Web Console form of a selected Service Desk object by accessing the shortcut menu for that object. The Web Console Form displays the properties of a selected Service Desk object and offers a possibility to start a number of actions previously defined in Service Desk.

---

**NOTE**

Service Configuration (SNVP) version 9.0 is built to be used with Service Desk version 5. For more information about the SD patch level refer to the SNVP README file.

---

## About Service Desk Objects in Service Hierarchies

When Service Configuration is installed with an existing Service Desk installation, you have access to Service Desk **objects** (known as “items” in Service Desk) that are currently stored in the Service Desk database. The following Service Desk **object types** (or “item types”) can be accessed from within Service Configuration:

- configuration item
- organization
- person
- service
- SLA (service level agreement)
- workgroup

Having access to Service Desk objects of the type listed above allows you to add these objects as status-carrying objects to your service hierarchies. You can do this in two ways:

- Rule-based** (view builder)

Service hierarchy rules work like **filters**. They define conditions that must be met by Service Desk objects before the objects are added to a **service hierarchy**. For example, you could set up a filter that adds objects of the type “configuration item”, where the category equals “software”. Whenever configuration items of the category “software” are added to Service Desk, they are also added to the hierarchy that uses this rule using the command **Apply Service Hierarchy Rules**.

- Manually**

If you only want to add a few Service Desk objects to your hierarchy, add them manually. See “Manually Adding Service Desk Objects as Child Objects” on page 60 for details.

You can also map existing Service Configuration objects to Service Desk objects as described in “Mapping Objects to Service Desk Objects” on page 64.

## About Service Hierarchy Rules

Service hierarchy rules represent the core component of the Service Configuration view builder. (The view builder is only available when using Service Configuration together with Service Desk.) Service hierarchy rules are applied to objects stored in the Service Desk database. Service Desk objects that meet the criteria of the rules are added to Service Configuration service hierarchies.

For example, you could set up a rule filter that adds objects of the type “configuration item”, where the category equals “software”. After the rule has been applied, Service Desk configuration items of the category “software” are added to the hierarchy that uses the rule. When new configuration items with the category “software” are added to Service Desk, they are also added to the service hierarchy the next time the rule is applied.

Where the objects are added depends on the type of service hierarchy rule. Service Configuration distinguishes between two types of rules:

### ❑ **Child object rules**

Child object rules are associated with Service Configuration objects (objects that are not mapped to Service Desk objects). Child object rules have a child filter only, which adds Service Desk objects as child objects below the Service Configuration object. Child object rules are useful as a starting point in a hierarchy. For example, you can add child object rules to root objects in order to add a layer of Service Desk objects below the root objects.

See “About Child Object Rules” on page 17 for more information.

### ❑ **Parent-and-child object rules**

Parent-and-child object rules are associated with service hierarchies, not with individual objects. These rules have both a parent and a child filter.

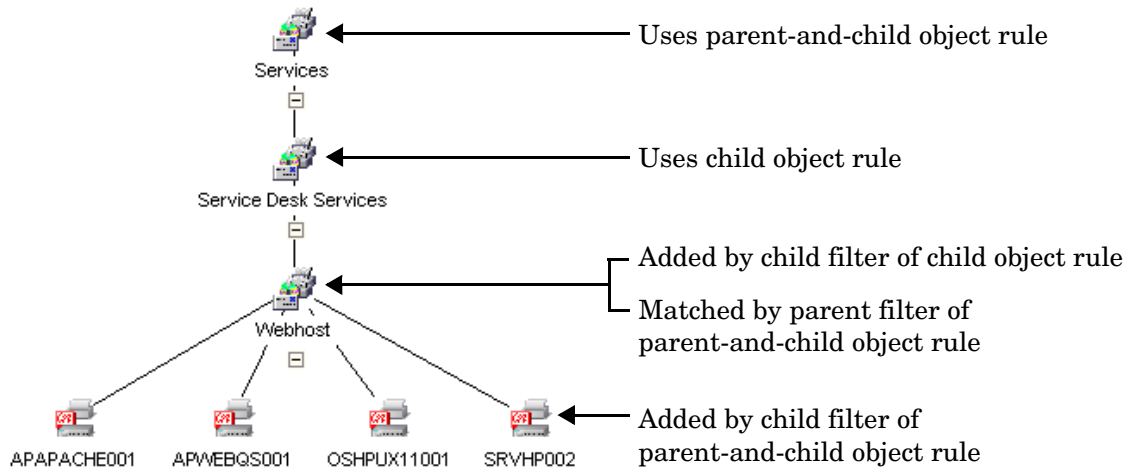
A parent-and-child object rule searches the service hierarchy for any object that matches the parent filter and that is mapped to a Service Desk object. If it finds a matching parent, the rule adds children to the service hierarchy, based on the Service Desk object’s children. The rule only adds the children that match the child filter.

See “About Parent-and-child Object Rules” on page 19 for more information about parent-and-child object rules.

Figure 1-1 shows a small service hierarchy that is built based on service hierarchy rules. The hierarchy itself is associated with a parent-and-child object rule. The root object “Service Desk Services” is associated with a child object rule. This rule adds the Service Desk object “Webhost” as child object below the parent object. “Webhost” matches the parent filter of the parent-and-child object rule. The child filter of the rule then adds a layer of Service Desk objects below the parent. The selection of Service Desk objects depends on the type of relation that is specified for the parent-and-child object rule, and on the filter criteria of the rule.

Figure 1-1

### A Rule-based Service Hierarchy





## About Child Object Rules

Child object rules select Service Desk objects that match the child filter of the rule and add them below the parent object with which the rule is associated. The parent object is always a Service Configuration object that is *not* mapped to a Service Desk object.

Child object rules have only one filter (for selecting child objects) in contrast to parent-and-child object rules, which have two filters (one for selecting parent objects and one for selecting child objects).

Figure 1-2 shows an example for a child object rule. The rule filters the Service Desk database for objects of the type “Service” that do not have a parent service and are therefore top-level services. All Service Desk objects that match this filter will be added below the parent object with which the rule is associated.

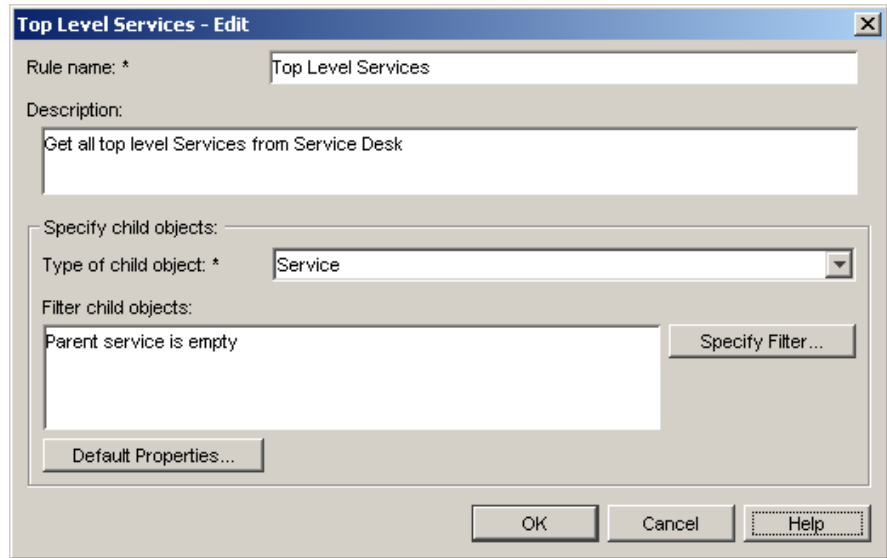
---

**TIP**

The rule assigns default properties to all matching child objects, for example default label, icon, logging options, and so on. You can change these defaults in the Default Properties dialog box.

---

**Figure 1-2**      **A Child Object Rule**



See “Creating a New Child Object Rule” on page 32 for more information about how to set up child object rules.

## About Parent-and-child Object Rules

Parent-and-child object rules are associated with service hierarchies, not with individual objects. These rules have both a parent and a child filter.

A parent-and-child object rule searches the service hierarchy for any object that matches the parent filter and that is mapped to a Service Desk object. If it finds a matching parent, the rule adds children to the service hierarchy, based on the Service Desk object's children. The rule only adds children that match the child filter.

Figure 1-3 shows an example for a parent-and-child object rule. The rule has the following filters:

❑ **Parent filter**

The parent filter matches objects of the type “Service”.

❑ **Child filter**

The child filter matches objects of the type “Configuration Item” that are related to objects of the type “Service” in Service Desk. The type of relation is “Used CIs”. The child filter therefore adds configuration items that are used by services as children below the parent.

---

**TIP**

The rule assigns default properties to all matching child objects, for example default label, icon, logging options, and so on. You can change these defaults in the Default Properties dialog box.

---

**Figure 1-3**      **A Parent-and-child Object Rule**

**Used Configuration Items - Edit**

Rule name: \*      Used Configuration Items

Description:  
Get all Configuration Items used by a Service

Specify parent objects:

Type of parent object: \*      Service

Filter parent objects:  
None      Specify Filter...

Specify child objects:

Type of child object: \*      Configuration Item

Association in Service Desk from parent to child: \*      Field...

Used CIs

Filter child objects:  
None      Specify Filter...

Default Properties...

OK      Cancel      Help

See “Creating a New Parent-and-child Object Rule” on page 37 for more information about setting up parent-child object rules.

## About the Example Service Hierarchy Rules

Service Configuration comes with two example service hierarchies that include a set of service hierarchy rules to help you get started. The following example hierarchies are available:

- ❑ Example Hierarchy “Service Desk Services and Configuration Items”
- ❑ Example Hierarchy “Service Desk Organizations”

Before the examples are available in the console, they must be imported into the Service Configuration database as described in “Importing the Example Service Hierarchy Rules” on page 26.

### **Example Hierarchy “Service Desk Services and Configuration Items”**

The example service hierarchy “Service Desk Services and Configuration Items” uses the following rules:

#### **❑ Child object rules**

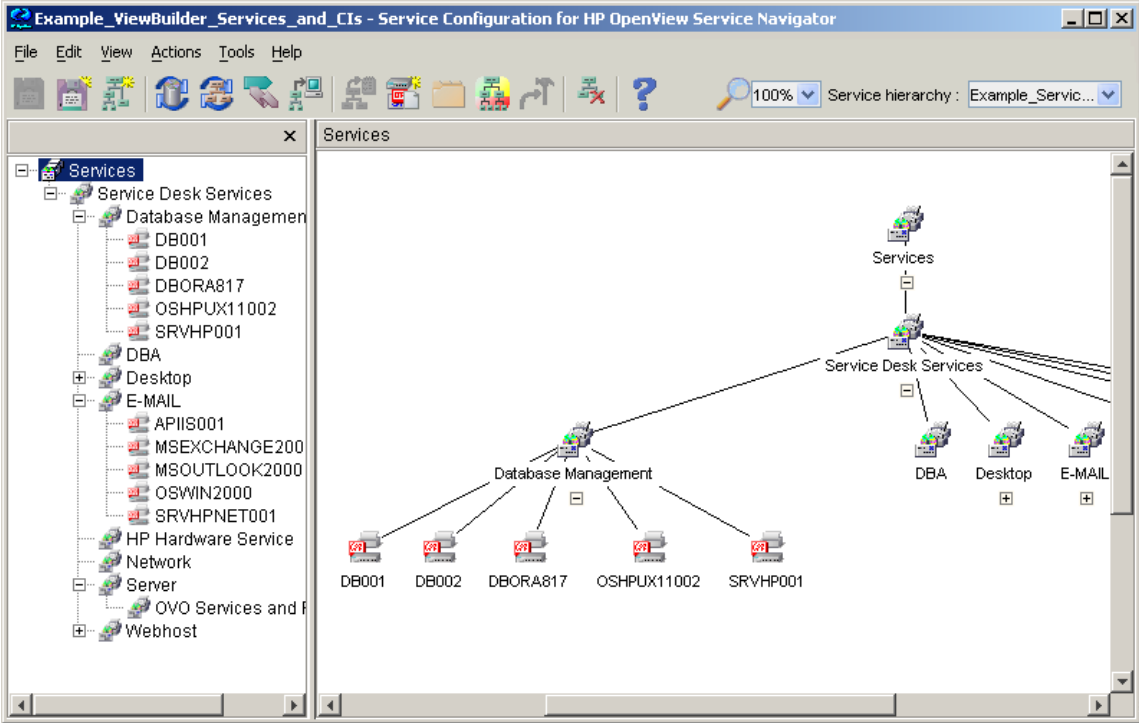
A child object rule is associated with the “Service Desk Services” root object. The rule selects all top-level services from Service Desk and adds them as child objects below the root object.

#### **❑ Parent-and-child object rules**

Two parent-and-child object rules are associated with this service hierarchy. The parent filter of the rules match the Service Desk services that are added by the child object rule described above. The child filter of the rules adds matching child objects of the type service or configuration item where the relation in Service Desk is “Child services” or “Used CIs”.

Figure 1-4 shows the Service Configuration console displaying the example service hierarchy “Service Desk Services and Configuration Items”.

Figure 1-4 Example Hierarchy “Service Desk Services and CIs”



See also “Example Hierarchy “Service Desk Organizations”” on page 24.

### **Example Hierarchy “Service Desk Organizations”**

The example service hierarchy “Service Desk Organizations” uses the following rules:

#### **❑ Child object rules**

A child object rule is associated with the “Organizations” root object. The rule selects all top-level organizations from Service Desk and adds them as child objects below the root object.

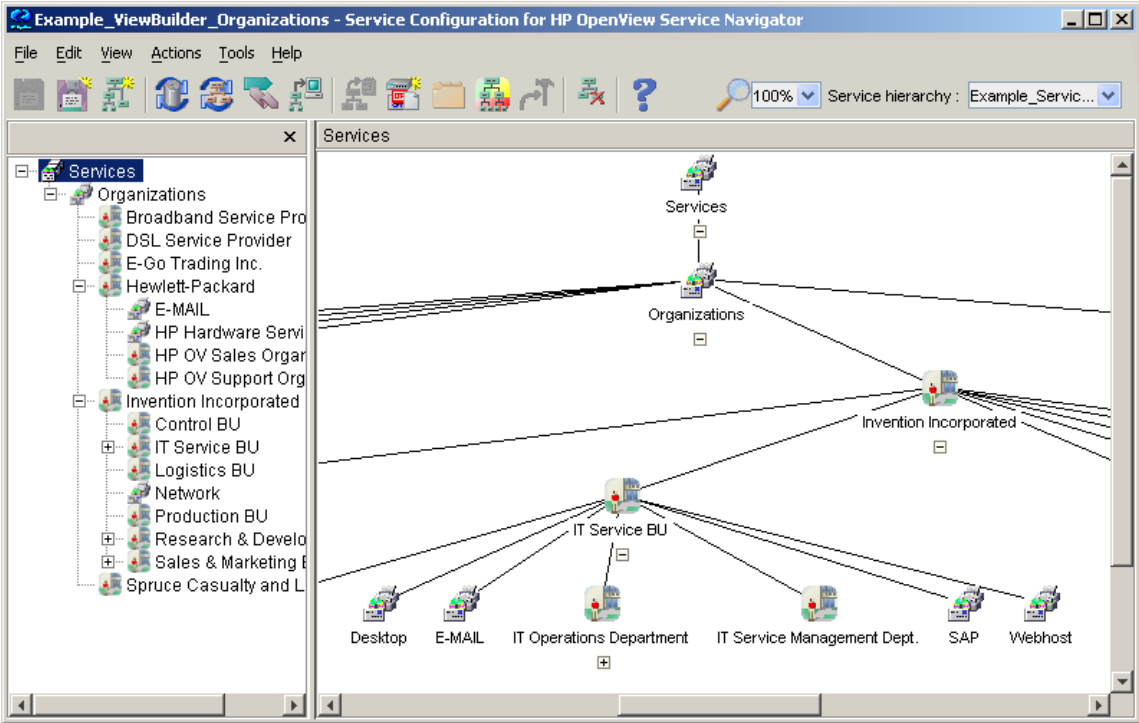
#### **❑ Parent-and-child object rules**

Two parent-and-child object rules are associated with this service hierarchy. The parent filter of the rules match the Service Desk organizations that are added by the child object rule described above. The child filter of the rules adds matching child objects of the type service or organization where the relation in Service Desk is “Service Level Agreements.Service” or “Child Organizations”.

Figure 1-5 shows the Service Configuration console displaying the example service hierarchy “Service Desk Organizations”.



Figure 1-5 Example Hierarchy “Service Desk Organizations”



See also “Example Hierarchy “Service Desk Services and Configuration Items”” on page 22.

## Importing the Example Service Hierarchy Rules

The service hierarchies that include the example rules are located in XML format in the following default installation directory on the Service Configuration client:

```
C:\Program Files\HP OpenView\examples
```

The XML files must be loaded into the Service Configuration database before the hierarchies are available in the console.

### To import the example service hierarchy rules:

1. Make sure the application server is running and the Service Configuration demonstration data is installed. The example rules are based on the demonstration data and will not work if this data is not present in the database.
2. On the Service Configuration client, execute the `cadm_LoadHierarchy` tool to load the example rules.

- a. Load the Service Desk services example:

```
cadm_LoadHierarchy "C:\Program Files\HP  
OpenView\examples\viewexample_SDServices.xml"
```

This adds the service hierarchy “Example\_ViewBuilder\_Services\_and\_CIs” to Service Configuration. See “Example Hierarchy “Service Desk Services and Configuration Items”” on page 22 for more information about this service hierarchy.

- b. Load the Service Desk organization example:

```
cadm_LoadHierarchy "C:\Program Files\HP  
OpenView\examples\viewexample_SDOrganizations.xml"
```

This adds the service hierarchy “Example\_ViewBuilder\_Organizations” to Service Configuration. See “Example Hierarchy “Service Desk Organizations”” on page 24 for more information about this service hierarchy.

3. Load the new demonstration hierarchies into the console and select **File: Apply Service Hierarchy Rules** or click the **Apply Service Hierarchy Rules** toolbar button. The associated service hierarchy rules are applied and add matching parent and child objects to the service hierarchy.



## Working with Service Hierarchy Rules

It is up to you to decide how to develop service hierarchy rules, the following top-down is recommended. Before you get started, you may want to familiarize yourself with the differences between the different types of service hierarchy rules as described in “About Service Hierarchy Rules” on page 15.

### To develop service hierarchy rules:

#### 1. Prerequisites.

Ensure the following prerequisites are met:

- *Root objects, manually added objects*

Before you can start developing service hierarchy rules, you need a layer of discovered or manually added objects, for example root objects. These objects will be the starting points for your service hierarchy rules.

- *Service Desk structure*

Parent-and-child object rules only select Service Desk objects that are related to each other. It is therefore important that your Service Desk objects are not stand-alone objects, otherwise the rules cannot work properly.

#### 2. Child object rules.

Add child object rules to the layer of manually added objects:

- a. Select a manually added object, for example a root object, and select **Add Child Object Rule** from the shortcut menu.
- b. Create a child object rule and save it. The rule is automatically associated with the selected object. See also “Creating a New Child Object Rule” on page 32.
- c. Apply the rule to test it. The child filter of the rule should add the desired objects below the manually added object. See also “Applying Service Hierarchy Rules” on page 54.

---

**NOTE**

---

You can also use the Child Object Rule Editor to create a new child object rule. You must then manually associate the rule with the manually added object before you can apply it.

### 3. Parent-and-child object rules.

Add parent-and-child object rules to the service hierarchy:

- a. Select a child object added by child object rules (or any other object that is mapped to a Service Desk object). Select **Add Parent-and-Child Object Rule** from the shortcut menu.
- b. Create a new parent-and-child object rule. The parent filter of the rule is preset with data from the selected object. Specify the child filter. The rule is automatically associated with the selected object. See also “Creating a New Parent-and-child Object Rule” on page 37.
- c. Apply the rule to test it. The parent filter of the rule determines if an object in the service hierarchy applies as parent. If so, the child filter of the rule adds Service Desk objects below the parent object. See also “Applying Service Hierarchy Rules” on page 54.

---

**NOTE**

---

You can also use the Parent-and-Child Object Rule Editor to create a new parent-and-child object rule. You must then define the parent filter yourself and must manually associate the rule with the service hierarchy.

4. Repeat step 3 until you are satisfied with the service hierarchy.

## Managing Child Object Rules

The Child Object Rule Editor allows you to manage child object rules. This is your main tool for configuring, editing, and deleting rules. You can also select rules in the editor to associate them with objects.

Child object rules are a type of service hierarchy rule. Service hierarchy rules determine the conditions that must be met before Service Desk objects are added to a service hierarchy. This builds service hierarchies dynamically: depending on which objects match the specified filter of the rule when the rules are applied, objects are added to or removed from the service hierarchy.

Child object rules are rules that are added to already existing parent objects (Service Configuration objects). They add child objects from Service Desk below the parent object.

### To open the Child Object Rule Editor:

Select **Tools: Child Object Rule Editor** from the menu bar.

The Child Object Rule Editor opens and shows all available rules. Each rule is listed with the rule name and a short description. You can sort rules by clicking the appropriate column heading. The summary section of the Child Object Rule Editor shows more detail for a selected rule.

---

### NOTE

The Child Object Rule Editor displays an additional column with check boxes when opened with the intention to associate a rule with an object. To choose a rule, double-click the desired rule or select the check box next to the rule and click **Choose**.

---

See also “Viewing Available Child Object Rules” on page 31.

## Viewing Available Child Object Rules

The Child Object Rule Editor displays a list of all available child object rules.

### To view *all* available child object rules:

Select **Tools: Child Object Rule Editor** from the menu bar.

The Child Object Rule Editor opens and shows all available child object rules. Each rule is listed with its name and a short description. You can sort rules by clicking the appropriate column heading.

### To see the *summary* of a selected child object rule:

In the Child Object Rule Editor, select the rule in the list of available rules.

A summary of the selected rule is displayed in the summary section of the Child Object Rule Editor.

See also “Creating a New Child Object Rule” on page 32.

### Creating a New Child Object Rule

You can create a new child object rule, instead of selecting from the list of existing rules.

#### To create a new child object rule:

1. Select **Tools: Child Object Rule Editor** from the menu bar. The Child Object Rule Editor opens.
2. Click **New** to open the New Child Object Rule dialog box.  

You can also select an existing rule and click **Duplicate**. A copy of the original rule is placed into the rule list for you to edit.
3. In the **Rule name** field, type a name for the rule. This information is required (max 80 characters).
4. In the **Description** field, type any comments or additional information about the rule. This information is optional (max 255 characters).
5. Specify the child objects:
  - a. Select the Service Desk object type for your child objects from the drop-down list.
  - b. Click **Specify Filter** to define the match conditions for the child objects. See “Specifying Filters for Service Hierarchy Rules” on page 41.
6. Click **Default Properties** to set default properties for the child objects selected by the rule. See “Setting Default Properties” on page 46 for details.
7. Click **OK** to save the rule and close the dialog box.
8. Associate the new rule with a parent object as described in “Associating Service Hierarchy Rules” on page 49.
9. Apply the service hierarchy rules as described in “Applying Service Hierarchy Rules” on page 54.

See also “Specifying Filters for Service Hierarchy Rules” on page 41.



## Editing a Child Object Rule

When you edit a child object rule, you can change all of the rule properties.

Child object rules are shared objects within the Service Configuration application, which means that changes to a rule affect all objects that use the rule.

When you attempt to edit a shared rule, Service Configuration asks you whether you want to edit the rule anyway, whether you want to duplicate it and edit the copy, or whether you want to cancel your edits.

### To edit a child object rule:

1. Open the Child Object Rule Editor. Choose **Tools: Child Object Rule Editor** from the menu bar. The Child Object Rule Editor opens.
2. In the Child Object Rule Editor, double-click the rule that you want to change or select it and click **Edit**.

The Edit Child Object Rule dialog box opens.

3. In the Edit Child Object Rule dialog box, make the necessary changes.
4. Click **OK** to confirm your changes.

See also “Deleting a Child Object Rule” on page 34.

### **Deleting a Child Object Rule**

Before deleting a child object rule, ensure that the rule is no longer associated with any objects because rules that are referenced by other objects cannot be deleted. After you have removed rules from objects, first save the service hierarchy, then delete the rules.

#### **To delete a child object rule:**

1. Open the Child Object Rule Editor. Choose **Tools: Child Object Rule Editor** from the menu bar. The Child Object Rule Editor opens.
2. In the Child Object Rule Editor, select the rule that you want to delete and click **Delete**.

A confirmation message asks you to confirm that you really want to delete the child object rule.

See also “Before You Delete a Child Object Rule” on page 34.

### **Before You Delete a Child Object Rule**

Before deleting a child object rule, ensure that the rule is no longer associated with any objects because rules that are referenced by objects cannot be deleted. After you have removed rules from objects, first save the service hierarchy, then delete the rules.

Consider carefully whether you really want to delete the child object rule. Deleted rules cannot be recovered.

## Managing Parent-and-child Object Rules

The Parent-and-Child Object Rule Editor allows you to manage parent-and-child object rules. This is your main tool for configuring, editing, and deleting rules. You can also select rules in the editor to associate them with service hierarchies.

Parent-and-child object rules are a type of service hierarchy rule. Service hierarchy rules determine the conditions that must be met before Service Desk objects are added to a service hierarchy. This builds service hierarchies dynamically: depending on which objects match the filter of the rule when it is applied, objects are added to or removed from the service hierarchy.

Parent-and-child object rules have both a parent and a child filter: if a service hierarchy contains an object that is mapped to a Service Desk object and that matches the parent filter of the rule, then the child filter will add all matching Service Desk objects as children below the parent.

One powerful aspect of parent-and-child object rules is that the child filter selects Service Desk objects based on the objects' parent-child relations in Service Desk. This means that for a parent-and-child object rule to find child objects in Service Desk, these objects must be related in Service Desk.

### To open the Parent-and-Child Object Rule Editor:

Select **Tools: Parent-and-Child Object Rule Editor** from the menu bar.

The Parent-and-Child Object Rule Editor opens and shows all available rules. Each rule is listed with the rule name, a short description, and the parent type. You can sort rules by clicking the appropriate column heading. The summary section of the Parent-and-Child Object Rule Editor shows more detail for a selected rule.

---

#### NOTE

The Parent-and-Child Object Rule Editor displays an additional column with check boxes when opened with the intention to associate a rule with a service hierarchy. To choose a rule, double-click the desired rule or select the check box next to the rule and click **Choose**.

---

See also “Viewing Available Parent-and-child Object Rules” on page 36.

### **Viewing Available Parent-and-child Object Rules**

The Parent-and-Child Object Rule Editor displays a list of all available parent-and-child object rules.

#### **To view *all* available parent-and-child object rules:**

Select **Tools: Parent-and-Child Object Rule Editor** from the menu bar.

The Parent-and-Child Object Rule Editor opens and shows all available parent-and-child object rules. Each rule is listed with its name, a short description, and the parent type. You can sort rules by clicking the appropriate column heading.

#### **To see the *summary* of a selected parent-and-child object rule:**

In the Parent-and-Child Object Rule Editor, select the rule in the list of available rules.

A summary of the selected rule is displayed in the summary section of the Parent-and-Child Object Rule Editor.

See also “Creating a New Parent-and-child Object Rule” on page 37.

## Creating a New Parent-and-child Object Rule

You can create a new parent-and-child object rule, instead of selecting from the list of existing rules.

### To create a new parent-and-child object rule:

1. Select **Tools: Parent-and-Child Object Rule Editor** from the menu bar. The Parent-and-Child Object Rule Editor opens.
2. Click **New** to open the New Parent-and-Child Object Rule dialog box.  

You can also select an existing rule and click **Duplicate**. A copy of the original rule is placed into the rule list for you to edit.
3. In the **Rule name** field, type a name for the rule. This information is required (max 80 characters).
4. In the **Description** field, type any comments or additional information about the rule. This information is optional (max 255 characters).
5. Specify the parent objects:
  - a. Select the Service Desk object type for your parent objects from the drop-down list.
  - b. Click **Specify Filter** to define the match conditions for the parent objects. See “Specifying Filters for Service Hierarchy Rules” on page 41.
6. Specify the child objects:
  - a. Select the Service Desk object type for your child objects from the drop-down list.
  - b. Select the Service Desk association (known as “relation” in Service Desk) between parent and child from the drop-down list. The available associations depend on the selected child object type.
  - c. Click **Specify Filter** to define the match conditions for the child objects. See “Specifying Filters for Service Hierarchy Rules” on page 41.
7. Click **Default Properties** to set default properties for the parent and child objects selected by the rule. See “Setting Default Properties” on page 46 for details.
8. Click **OK** to save the rule and close the dialog box.

9. Associate the new rule with a service hierarchy as described in “Associating Service Hierarchy Rules” on page 49.

10. Apply the service hierarchy rules as described in “Applying Service Hierarchy Rules” on page 54.

See also “Specifying Filters for Service Hierarchy Rules” on page 41.

## Editing a Parent-and-child Object Rule

When you edit a parent-and-child object rule, you can change all of the rule properties.

Parent-and-child object rules are shared objects within the Service Configuration application which means that changes to a rule affect all service hierarchies that use the rule.

When you attempt to edit a shared rule, Service Configuration asks you whether you want to edit the rule anyway, whether you want to duplicate it and edit the copy, or whether you want to cancel your edits.

### To edit a parent-and-child object rule:

1. Open the Parent-and-Child Object Rule Editor. Choose **Tools: Parent-and-Child Object Rule Editor** from the menu bar. The Parent-and-Child Object Rule Editor opens.
2. In the Parent-and-Child Object Rule Editor, double-click the rule that you want to change, or select it and click **Edit**.

The Edit Parent-and-Child Object Rule dialog box opens.

3. In the Edit Parent-and-Child Object Rule dialog box, make the necessary changes.
4. Click **OK** to confirm your changes.

See also “Deleting a Parent-and-child Object Rule” on page 40.

### **Deleting a Parent-and-child Object Rule**

Before deleting a parent-and-child object rule, ensure that the rule is no longer associated with any service hierarchies because rules that are referenced by other objects cannot be deleted. After you have removed rules from the service hierarchy, first save the hierarchy, then delete the rules.

#### **To delete a parent-and-child object rule:**

1. Open the Parent-and-Child Object Rule Editor. Choose **Tools: Parent-and-Child Object Rule Editor** from the menu bar. The Parent-and-Child Object Rule Editor opens.
2. In the Parent-and-Child Object Rule Editor, select the rule that you want to delete and click **Delete**.

A confirmation message asks you to confirm that you really want to delete the parent-and-child object rule.

See also “Before You Delete a Parent-and-child Object Rule” on page 40.

### **Before You Delete a Parent-and-child Object Rule**

Before deleting a parent-and-child object rule, ensure that the rule is no longer associated with any service hierarchies because rules that are referenced by objects cannot be deleted. After you have removed rules from service hierarchies, first save the hierarchies, then delete the rules.

Consider carefully whether you really want to delete the parent-and-child object rule. Deleted rules cannot be recovered.



## Specifying Filters for Service Hierarchy Rules

Filtering lets you determine which Service Desk objects will be included in a service hierarchy. You can choose to filter by attributes or by associations.

---

**TIP**

Specifying a filter for parent or child objects is very similar to finding objects using Advanced Find.

---

## Filtering Objects by Attributes

Filtering by attributes lets you choose from a list of attributes that describe a particular Service Desk object type or object types that are associated with that object type. In Service Desk, attributes are known as “fields”.

---

### NOTE

---

The specified filter is always case-sensitive.

### To filter objects by attributes:

1. Click **Specify Filter** in the dialog box of the rule you are currently configuring. The Specify Filter dialog box opens and displays the By attribute tab by default.

The **Filter on** field displays the Service Desk object type you have previously selected for your parent or child objects.

2. Specify the filter criteria:

- a. Select the attribute you want to use from the **Field** drop-down menu. You can choose from up to three levels of attributes.
- b. Select the operator you want to use from the **Operator** drop-down list. See “Advanced Find Operators” on page 86 for more information about available search operators.
- c. Type the value you want to match in the **Value** field or click the **Quick Find** button to open the Quick Find dialog box.

3. Click **Add Criteria** to add the criterion to the summary of filter criteria. To edit the criterion, double-click it in the summary section or select it and click **Edit**.

4. To add additional criteria, repeat step 2 and 3.

If you add more than one criterion to the list, only objects that meet *all* of the search criteria are found.

5. Click **OK** to save your filter and close the dialog box.

---

**TIP**

To remove a single criterion, select it and click **Remove**. Click **Clear** to remove all criteria at once.

---

See also “Filtering Objects by Associations” on page 44.

## Filtering Objects by Associations

Filtering by associations lets you choose from a list of association types that describe how the object types are related to each other. In Service Desk, associations are known as “relations”.

---

### NOTE

---

The specified filter is always case-sensitive.

### To filter objects by associations:

1. Click **Specify Filter** in the dialog box of the rule you are currently configuring. The Specify Filter dialog box opens and displays the By attribute tab by default.
2. Select the By association tab.  
  
The **Filter on** field on the By association tab displays the Service Desk object type you have previously selected for your parent or child objects.
3. Specify the filter criteria:
  - a. Select the attribute you want to use from the **Field** drop-down menu. You can choose from up to two levels of associated object types.
  - b. Select the operator you want to use from the **Operator** drop-down list. See “Advanced Find Operators” on page 86 for more information about available search operators.
  - c. Type the value you want to match in the **Value** field or click the **Quick Find** button to open the Quick Find dialog box.
4. Click **Add Criteria** to add the criterion to the summary of filter criteria. To edit the criterion, double-click it in the summary section or select it and click **Edit**.
5. To add additional criteria, repeat step 3 and 4.  
  
If you add more than one criterion to the list, only objects that meet *all* of the search criteria are found.
6. Click **OK** to save your filter and close the dialog box.

---

**TIP**

To remove a single criterion, select it and click **Remove**. Click **Clear** to remove all criteria at once.

---

See also “Filtering Objects by Attributes” on page 42.

## Setting Default Properties

Objects added by service hierarchy rules have the same default properties as objects that are added manually. To set different values, you can specify for each service hierarchy rule, what default values should be assigned to the generated objects. For example, you can assign a particular object icon to all child objects that are added by a particular rule. (By default, the object type determines the icon of a generated child object.)

---

### TIP

To further refine object properties, you can also change the properties of individual generated objects. These objects are then marked as manually changed so that your changes will not be overwritten the next time the service hierarchy rules are applied.

---

---

### NOTE

If your rules add multiple references to the same Service Desk object, make sure the default properties are the same in each rule.

You can add multiple references to the same Service Desk object in the same hierarchy, but the references all share the same attributes.

---

Use the following procedure to customize object properties on a rule-by-rule basis.

#### To set default properties:

1. Click **Default Properties** in the Child Object Rule dialog box or in the Parent-and-child Object dialog box. The Default Properties dialog box opens.
2. You can change the following default properties:

- a. **Label**

By default, the object's label in Service Desk will be applied to child objects. To change the default, select another Service Desk field in the list.

b. **Presentation**

- *Icon*

By default, the object type determines the icon that represents the object in the console. You can also select another icon for the object, or add an URL to a particular icon.

- *Background graphic*

By default, the background color is determined by the default Windows setting for the desktop. You can also select another background for the object, or add an URL to a particular icon.

- *Presentation depth*

By default, the presentation depth is set to 2. This includes the object itself and one more level below the object.

c. **Status logging**

By default, no status information will be logged. You can choose to log the status of the object itself and all of its child objects, or you can specify the level of child objects to log. (Select one (1) to log only the status of the object itself.)

d. **Object status**

- *Propagation from child to parent*

Choose a propagation rule to determine how child objects propagate their status to parent objects. (The default rule is Unchanged.)

- *Calculation rule*

Choose a calculation rule to determine how child objects calculate their status. (The default rule is Most Critical.)

- *Propagation from messages to child*

Select a propagation rule to determine how messages propagate their status to the child object. (The default rule is Unchanged.)

- *Service names in messages*

Choose whether you want to use a particular field in Service Desk to create service names in messages. Select the field from the list box. (By default, no service names in messages are generated.)

---

**NOTE**

Service Configuration interprets carriage returns as separators for service names in messages. Therefore, if the value of the selected Service Desk field contains a carriage return, the string will be considered as two service names.

---

3. Click **OK** to save your changes and close the dialog box.



## Associating Service Hierarchy Rules

Depending on the type of rule, service hierarchy rules are either associated with objects or with service hierarchies:

❑ **Child object rules**

Child object rules can be associated with any Service Configuration object that is not mapped to a Service Desk object. See “Associating Child Object Rules” on page 50.

❑ **Parent-and-child object rules**

Parent-and-child object rules are associated with service hierarchies. See “Associating Parent-and-child Object Rules” on page 52.

When you remove a service hierarchy rule from an object or from a service hierarchy, only the association is removed. The rule itself continues to exist.

If rules no longer apply, objects that were added by service hierarchy rules will be removed the next time the service hierarchy rules are applied. Manually added or discovered objects will also be removed from the hierarchy if the parent objects are no longer added by a rule.

See also “Associating Child Object Rules” on page 50.

## Associating Child Object Rules

Child object rules can be associated with any Service Configuration object that is not mapped to a Service Desk object. Child object rules provide a starting point for parent-and-child object rules because child objects added by child object rules can be matched by the parent filter of parent-and-child object rules.

There are several ways of adding child object rules to Service Configuration. The quickest way is to use the shortcut menu to add a rule to a selected object. You can also manually associate a rule with an object using the object's properties dialog box.

### To associate a child object rule with an object:

1. Select the object to which you want to add the child object rule.
2. Double-click the object to display its properties. You can also select **Object Properties** from the shortcut menu or select **Edit: Object Properties** from the menu bar. Alternatively, click the **Object Properties** toolbar button. A property summary dialog box for the selected object opens, displaying the Attributes tab by default.
3. Select the **Associations** tab.  
  
The Associations tab displays folders for the different types of associations. Click the + (plus) sign next to an association group or double-click the association group to expand the folder.
4. Select **Actions: Add Child Object Rules** from the dialog menu. The Child Object Rule Editor opens.
5. In the Child Object Rule Editor, choose a rule from the list by selecting the appropriate check box. You can select more than one rule if desired.

---

### TIP

---

Sort the rules by name to help you locate the rule you are looking for.

6. Double-click the selected rule or click **Choose** to choose the rule(s) and close the Child Object Rule Editor.

The associated rule is now listed on the Associations tab. If necessary, expand the list of associated child object rules by clicking the + (plus) sign or double-clicking the child object rule grouping.

---

**NOTE**

To remove a child object rule from an object, select the rule on the Associations tab and select **Actions: Remove** from the dialog menu. This does not delete the rule, only the association to the object.

---

See also “Associating Parent-and-child Object Rules” on page 52.

## Associating Parent-and-child Object Rules

Parent-and-child object rules are associated with service hierarchies. If the parent filter matches a mapped object in the hierarchy, child objects that match the child filter are added below the parent.

There are several ways of adding parent-and-child object rules to Service Configuration. The quickest way is to use the shortcut menu to add a rule to a hierarchy. You can also manually associate a rule with a service hierarchy using the hierarchy's properties dialog box.

### To associate parent-and-child object rules with a service hierarchy:

1. Load the service hierarchy into the Service Configuration console:
  - a. Select **File: Load** from the menu bar. The Load Service Hierarchy dialog box opens.
  - b. Select the service hierarchy you want to load.  
Sort the service hierarchies by clicking the appropriate column heading to help you quickly find the one you want to load.
  - c. Click **Choose**. The selected service hierarchy is displayed in the Service Configuration main console.
2. Double-click the Services object in the map view or select **Service Hierarchy Properties** from the shortcut menu. Alternatively, select **File: Service Hierarchy Properties** from the menu bar. A property summary dialog box for the hierarchy opens, displaying the Attributes tab by default.
3. On the Attributes tab, click the **Parent-and-child object rules** link. The Parent-and-Child Object Rule Editor opens
4. In the Parent-and-Child Object Rule Editor, choose a rule from the list by selecting the appropriate check box. You can select more than one rule if desired.

---

**TIP**

---

Sort the rules by name to help you locate the rule you are looking for.

5. Double-click the selected rule or click **Choose** to choose the rule(s) and close the Parent-and-Child Object Rule Editor.

The associated rule is now listed on the Attributes tab.

---

**NOTE**

To remove a parent-and-child object rule from a hierarchy, clear the check box in the Parent-and-Child Object Rule Editor. This does not delete the rule, only the association to the service hierarchy.

---

See also “Associating Child Object Rules” on page 50.

## Applying Service Hierarchy Rules

When applying service hierarchy rules, Service Configuration first evaluates all child object rules, then it evaluates all parent-and-child object rules.

If a rule or a Service Desk object has been changed or removed, the rule filter may no longer select the objects it previously selected. These objects will then be removed from the service hierarchy. Manually added or discovered objects will also be removed from the hierarchy if the parent objects are no longer found by the rule.

Manual changes to generated objects are preserved. The service hierarchy rules do, however, add, remove, or change child objects below generated parent objects.

---

**TIP**

The tool `cadm_ApplyHierarchyRules (1m)` applies service hierarchy rules to a given service hierarchy by using the command line. To apply rules on a regular basis, schedule a task on the computer where the Service Configuration client is installed. This helps you keep your service hierarchies up to date in a changing Service Desk environment.

---

### To apply service hierarchy rules:

1. Select **File: Apply Service Hierarchy Rules** or click the **Apply Service Hierarchy Rules** toolbar button.

If there are unsaved changes in the service hierarchy, a dialog box opens where you can choose to save the hierarchy.

2. Click **Continue** to save the hierarchy or **Cancel** to apply the service hierarchy rules without first saving the hierarchy.

---

**TIP**

It is recommended to save the hierarchy so that you can reload it if the application of rules does not produce the expected result.

---

3. A message informs you about the progress of the application of rules. Click **Close** when the process has finished.

See also “Monitoring the Application of Service Hierarchy Rules” on page 55.

## Monitoring the Application of Service Hierarchy Rules

Before service hierarchy rules are applied, you are asked to save the service hierarchy. It is generally recommended to save your changes so that you can reload the hierarchy if the applied rules do not produce the expected result.

The application of service hierarchy rules may take some time, depending on the number of objects in the Service Desk database.

Once all rules have been applied, a message confirms that the application of rules finished successfully. Click **Close** to close the dialog box.

See also “Stopping the Application of Service Hierarchy Rules” on page 55.

## Stopping the Application of Service Hierarchy Rules

If the application of rules hangs, you can stop it by clicking the **Stop** button. Be aware though that stopping the application of rules may result in inconsistent data because not all applicable objects may have been found yet.

If you do have to stop the application of rules, for example because the process hangs, you should reload the service hierarchy and re-apply the rules.

See also “Monitoring the Application of Service Hierarchy Rules” on page 55.

## Adding Rules Using the Service Hierarchy Wizard

When using the Service Desk integration, the Service Configuration Service Hierarchy Wizard is modified to let you set up service hierarchy rules:

### ❑ **Child object rules**

On the Root Objects page of the Service Hierarchy Wizard, click **New Rule** to open the Child Object Rule Editor. In the Child Object Rule Editor, select an existing rule or click **New** to create a new rule. See “Managing Child Object Rules” on page 30 for more information about child object rules.

### ❑ **Parent-and-child object rules**

On the Root Objects page of the Service Hierarchy Wizard, click **Next Page >** to open the Parent-and-child object rules page. Select an existing rule or click **New** to create a new rule. See “Adding Parent-and-child Object Rules” on page 57 for more information about adding parent-and-child object rules using the Service Hierarchy Wizard.



## Adding Parent-and-child Object Rules

Use the Service Hierarchy Wizard to create new service hierarchies. To add parent-and-child object rules to an existing service hierarchy, use the Parent-and-Child Object Rule Editor.

### To add parent-and-child object rules:

1. Start the Service Hierarchy Wizard.

Select **File: New -> Service Hierarchy** from the menu bar, or click the **Service Hierarchy** toolbar button. The Service Hierarchy Wizard opens and displays the General page.

2. Configure the General page and the Root Objects page.

---

#### NOTE

---

You should add at least one child object rule on the Root Objects page before you can add a parent-and-child object rule.

When you have done this, click **Next Page >** to open the Parent-and-child object rules page. Alternatively, click the **Parent-and-child object rules** link in the summary section of the wizard page.

3. On the **Parent-and-child object rules** page, add a rule:

- If rules already exist, choose a rule from the list by selecting the appropriate check box. You can select more than one rule, if you want.
- If no rules are configured yet, click **New** to configure a new parent-and-child object rule. See also “Creating a New Parent-and-child Object Rule” on page 37.

To edit a parent-and-child object rule, double-click it or select it and click **Edit**. Use the **Delete** button to delete a rule.

4. When you are satisfied with the list of parent-and-child object rules, click **Finish** to complete the configuration of the service hierarchy and to close the wizard. The new service hierarchy is displayed in the Service Configuration console and you can now start further refining it.

---

**NOTE**

Check the summary section at the bottom of the wizard page to get a quick overview of the currently added parent-and-child object rules.

---

## Viewing Service Hierarchy Properties

To view service hierarchy properties:

1. Load the service hierarchy into the Service Configuration console:
  - a. Select **File: Load** from the menu bar. The Load Service Hierarchy dialog box opens.
  - b. Select the service hierarchy you want to load.  
  
Sort the service hierarchies by clicking the appropriate column heading to help you quickly find the one you want to load.
  - c. Click **Choose**. The selected service hierarchy is displayed in the Service Configuration main console.
2. Double-click the Services object in the map view or select **Service Hierarchy Properties** from the shortcut menu. Alternatively, select **File: Service Hierarchy Properties** from the menu bar. A property summary dialog box for the service hierarchy opens, displaying the **Attributes** tab.
3. You may need to scroll through the summary section of the **Attributes** tab to view all attributes.
4. Click the **General** link to view the general properties of the service hierarchy.
5. Click the **Parent-and-child object rules** link to open the Parent-and-Child Object Rule Editor. See “Managing Parent-and-child Object Rules” on page 35 for details. You can also click the link to an associated rule to view the rule’s properties.
6. Click the **Close** toolbar button or select **File: Close** from the menu bar of the object properties dialog box to close the dialog box.

## Manually Adding Service Desk Objects as Child Objects

When you add a Service Desk object to a service hierarchy, Service Configuration does not add the Service Desk object itself but creates an association between a Service Configuration object and the original Service Desk object. The Service Configuration object acts as a proxy object between Service Configuration and Service Desk and has the same name, label, and icon as the Service Desk object.

---

### NOTE

When a Service Configuration object is mapped to a Service Desk object of type Service or Configuration Item, and has a non-empty SN Name attribute, the value of this attribute will be used as name of the new Service Configuration object.

---

Use the following procedure to manually add Service Desk objects to a service hierarchy.

#### To manually add Service Desk objects as child objects:

1. Load the service hierarchy into the Service Configuration console.
2. Expand the service hierarchy and select the parent object under which you want to add the child object.
3. Select **Add Child Object** from the shortcut menu, or **Edit: Add Child Object** from the menu bar. Alternatively, click the **Add Child Object** toolbar button.

The Add Child Object dialog box opens and displays all Service Configuration objects that are configured for the current service hierarchy.

---

### NOTE

Service Configuration sometimes displays Advanced Find instead of the Add Child Object dialog box. This is the case when more objects are configured than the Add Child Object dialog box can display. Use Advanced Find to search for objects you want to add as child objects.

---

4. Choose the kind of object you want to add:

- *Already configured objects*

If you want to add an already configured Service Configuration object to the service hierarchy, select **Show objects already defined in a service hierarchy**.

- *Service Desk objects*

If you want to add a Service Desk object to the service hierarchy, select **Show all Service Desk objects** and select the Service Desk object type.

5. Double-click the desired object in the list or select it and click **Choose**.

If you want to inspect the properties of the selected object before choosing it, click **Properties**. Properties can only be displayed for Service Configuration objects, not for Service Desk objects.

If you cannot find an appropriate object, click **New** to create a new Service Configuration object. You cannot create new Service Desk objects from within Service Configuration.

a. To find an object, type a few characters of the object you are looking for in the **Look for** field.

b. Click **Find Next**.

Service Configuration selects the first object that matches the typed characters. If the object found is not the one you are looking for, click **Find Next** again or change the characters you typed into the Look for field. Service Configuration always searches all columns.

c. If there are too many objects in the list, click the **Advanced Find** button to define a more sophisticated search pattern.

6. When the child object has been added, select **File: Save** from the menu bar to save your changes.

See also “Adding the Same Service Desk Object More Than Once to the Same Service Hierarchy” on page 62.

## Adding the Same Service Desk Object More Than Once to the Same Service Hierarchy

You can add the same Service Desk object to the same service hierarchy but under different parent objects. Service Configuration simply creates an association between the first occurrence of the object and the new addition. This has the following effects:

### ❑ Identical objects in Service Configuration

All occurrences of the object in the same service hierarchy are identical. They all share the same attributes such as name and label, as well as the same associations to actions.

All occurrences of the object use the same status rules and have the same child objects which is why all occurrences of the object in the same service hierarchy have the *same status* in Service Navigator.

### ❑ Modifying the object in Service Desk

When the object is modified in Service Desk, the occurrences of the object in Service Configuration are *not* updated. Only the information in the Mapping to Service Desk objects dialog box is refreshed. See “Mapping Objects to Service Desk Objects” on page 64.

### ❑ Deleting the object in Service Desk

When you delete the Service Desk object, the Service Configuration object that was mapped to this Service Desk object no longer refers to it. The Service configuration object’s properties contain no information about the corresponding Service Desk object (Object: Unknown), and when you click Mapping to Service Desk objects, a warning message is displayed.

See also “Adding the Same Service Desk Object to Different Service Hierarchies” on page 63.

## Adding the Same Service Desk Object to Different Service Hierarchies

You can add the same Service Desk object to different service hierarchies. Service Configuration simply creates a new association to the Service Desk object. This has the following effects:

### ❑ Individual objects

Each occurrence of the object in each service hierarchy has its own, individual attributes such as name and label and individual associations to actions.

Each occurrence of the object in each service hierarchy can use different status rules and have different child objects in Service Configuration. Hence each occurrence of the object can have a *different status* in Service Navigator.

---

### NOTE

In the Service Navigator operator console, the operator can use the “Find in Service Views” tool to locate all occurrences of an object in different service hierarchies.

---

### ❑ Modifying the object in Service Desk

When the object is modified in Service Desk, the occurrences of the object in Service Configuration are *not* updated. Only the information in the Mapping to Service Desk objects dialog box is refreshed. See “Mapping Objects to Service Desk Objects” on page 64.

### ❑ Deleting the object in Service Desk

When you delete the Service Desk object, the Service Configuration object can no longer be mapped to it and a warning message is displayed.

See also “Adding the Same Service Desk Object More Than Once to the Same Service Hierarchy” on page 62.

## Mapping Objects to Service Desk Objects

Mapping Service Configuration objects to Service Desk objects creates an association between the objects in the shared repository.

You can map existing objects to the corresponding Service Desk objects if you have first manually added objects to Service Configuration and have only later started working with Service Desk.

---

### NOTE

If, for some reason, the mapped Service Desk object can no longer be accessed, the name of the mapped Service Configuration object changes to “unknown” and an error message is displayed. In this case you should verify whether the object still exists in Service Desk and, if necessary, re-map the Service Configuration object.

---

### NOTE

When a Service Configuration object is mapped to a Service Desk object of type Service or Configuration Item, and has a non-empty SN Name attribute, the value of this attribute will be used as name of the new Service Configuration object.

---

### To map objects to Service Desk objects:

1. Select the object in the Service Configuration console.
2. Double-click the object to display its properties. You can also select **Object Properties** from the context menu or select **Edit: Object Properties** from the menu bar. Alternatively, click the **Object Properties** toolbar button. A property summary dialog box for the selected object opens and displays the Attributes tab.
3. Click the Mapping to Service Desk Objects link. The Mapping to Service Desk Objects dialog box opens.
4. Click **Mapped to a Service Desk object**. Quick Find opens where you can look for the desired Service Desk object.
5. Click **OK** to save the mapping and close the dialog box.



6. A message asks you whether you want to change the name, label, and icon of the Service Configuration object to those of the mapped Service Desk object. Click **Yes** to change these attributes.

---

**NOTE**

If you change the name of an object in this way while in simulation mode, any simulated messages that matched the object no longer match. Therefore these messages do not contribute to the object's status anymore and as a result the object's status may change.

For more information on status simulation, refer to the *Service Configuration for Service Navigator Concepts Guide*.

---

If you click **No** in the confirmation message, the attributes will not be changed which means that the operator will not be able to use the "Find in Service" Views tool in the Service Navigator operator console.

7. Select **File: Save** from the menu bar to save your service hierarchy. See also "Manually Adding Service Desk Objects as Child Objects" on page 60.

## Before You Map to a Service Desk Object

When you map an existing Service Configuration object to a Service Desk object, you can choose to let Service Configuration change the name, label, and icon of the object to those of the mapped Service Desk object. Click **Yes** to change these attributes.

If you click **No**, the attributes of the existing Service Configuration object will not be changed which means that the “Find in Service” Views tool of the Service Navigator operator console will not be able to locate this object.

If child object rules are associated with the Service Configuration object, they will be removed, regardless of whether you agree to change the object attributes or not. The next time service hierarchy rules are applied, the parent filter of a parent-and-child object rule may match the mapped Service Desk object. If so, the child filter of the rule will add child objects below the mapped Service Desk object.

See also “Before You Change a Service Desk Mapping” on page 66.

## Before You Change a Service Desk Mapping

When you change the mapping of an object to another Service Desk object, Service Configuration asks you whether you also want to change the name, label, and icon of the object to the name, label, and icon of the other Service Desk object. Click **Yes** to update these attributes.

If you click **No**, the attributes of the existing Service Configuration object are not updated which means that the “Find in Service” Views tool of the Service Navigator operator console will not be able to locate the correct object.

Regardless of whether you choose to update the attributes or not, the parent filter of parent-and-child object rules may no longer match, which could cause existing child objects to be removed the next time service hierarchy rules are applied. New child objects may be added if other rules apply.

---

**TIP**

When you change the mapping of an object generated by service hierarchy rules, the origin property of the object changes from generated to manually added.

---

See also “Before You Remove a Service Desk Mapping” on page 67.

### **Before You Remove a Service Desk Mapping**

When you remove a mapping, only the association between the Service Configuration object and the Service Desk object is removed. The name, label, and icon of the Service Configuration object continue to be those of the formerly mapped Service Desk object. Since they are no longer valid for the unmapped Service Configuration object, you should manually change them.

If you do not at least change the name, the object retains the identifier of the formerly mapped Service Desk object. This means that it may continue to be located by any “Find in Service Views” searches conducted by the operator in the Service Navigator operator console, and the mapping may be added again the next time service hierarchy rules are applied.

Click **OK** to close the warning dialog box and use the General and Presentation property dialog boxes to update the name, label, and icon of the Service Configuration object.

If parent-and-child object rules added child objects below the currently mapped object, the child objects will be removed the next time service hierarchy rules are applied. This is because parent-and-child object rules only apply to mapped Service Desk objects.

---

**TIP**

When you remove the mapping of an object generated by service hierarchy rules, the origin property of the object changes from generated to manually added.

---

See also “Before You Map to a Service Desk Object” on page 66.



---

---

**3****Service Desk Integration for the  
OVO Operator**

## About the Service Desk Integration for the OVO Operator

The Service Desk integration for the OVO operator combines three products in one: the operator can access Service Desk Service Pages and Service Desk data forms and Web Console forms for a selected object from within the Service Navigator console, instead of having to manually search for related data in the Service Desk application.

The integration covers the following aspects:

### ❑ **Service Pages**

The Service Pages application is integrated into the workspace of the Service Navigator operator console so that the operator can launch Service Pages by starting an action from the shortcut menu of the selected service or configuration item. Depending on the selected action, related service calls, incidents, problems, changes, or workorders are displayed.

This integration is ideal for you if you already use Service Desk with Service Pages and OVO.

See “Service Pages Integration for the OVO Operator” on page 72 for details.

### ❑ **Service Desk Data Forms**

With the Service Desk data form integration configured, operators can launch the Service Desk client console in the context of a particular Service Desk object from within the Service Navigator console. The Service Desk client console displays detailed information about the selected object in a Service Desk data form.

This integration requires an installation of the Service Desk client on the same system as the Service Navigator console.

This integration is supported only on Windows platform.

See “Service Desk Data Form Integration for the OVO Operator” on page 81 for details.

❑ **Web Console integration**

With the Web Console integration enabled, operators using the Service Navigator operator console can open a Web Console form of a selected Service Desk object by accessing the shortcut menu for that object. The Web Console Form displays the properties of a selected Service Desk object and offers a possibility to start a number of actions previously defined in Service Desk.

## Service Pages Integration for the OVO Operator

This section describes the following aspects of the Service Pages integration:

- ❑ **About the Service Pages Integration**  
See “About the Service Pages Integration” on page 73.
- ❑ **How the Service Pages Integration Works**  
See “How the Service Pages Integration Works” on page 75.
- ❑ **Configuring the Service Pages Integration**  
See “Configuring the Service Pages Integration” on page 76.
- ❑ **Automating the Service Pages Authentication**  
See “Automating the Service Pages Authentication” on page 78.
- ❑ **Troubleshooting the Service Pages Integration**  
See “Troubleshooting the Service Pages Integration” on page 79.



## About the Service Pages Integration

With the Service Pages integration configured, operators using the Service Navigator operator console can browse Service Desk data related to a particular service or configuration item. Service Pages are easy to launch from the Service Navigator console: operators only need to access the shortcut menu of the Service Desk service or configuration item, and select the desired action from the menu. The following actions can be enabled:

- ❑ **Service calls**  
Related service calls for services and configuration items.
- ❑ **Incidents**  
Related incidents for configuration items.
- ❑ **Problems**  
Related problems for configuration items.
- ❑ **Changes**  
Related changes for configuration items.
- ❑ **Workorders**  
Related workorders for configuration items.

Figure 3-1 shows the shortcut menu for a Service Desk service in Service Navigator. The Show related service calls menu item launches Service Pages and displays all service calls related to the selected service.

**Figure 3-1** Launching Service Pages from the Service Navigator Console

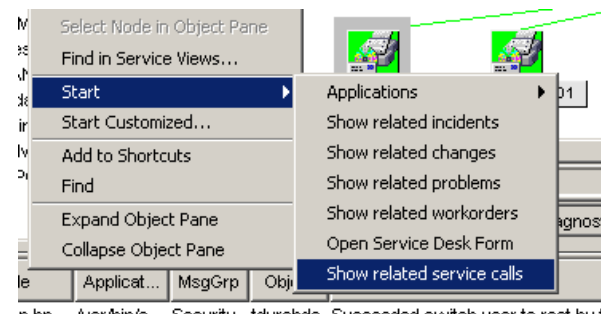
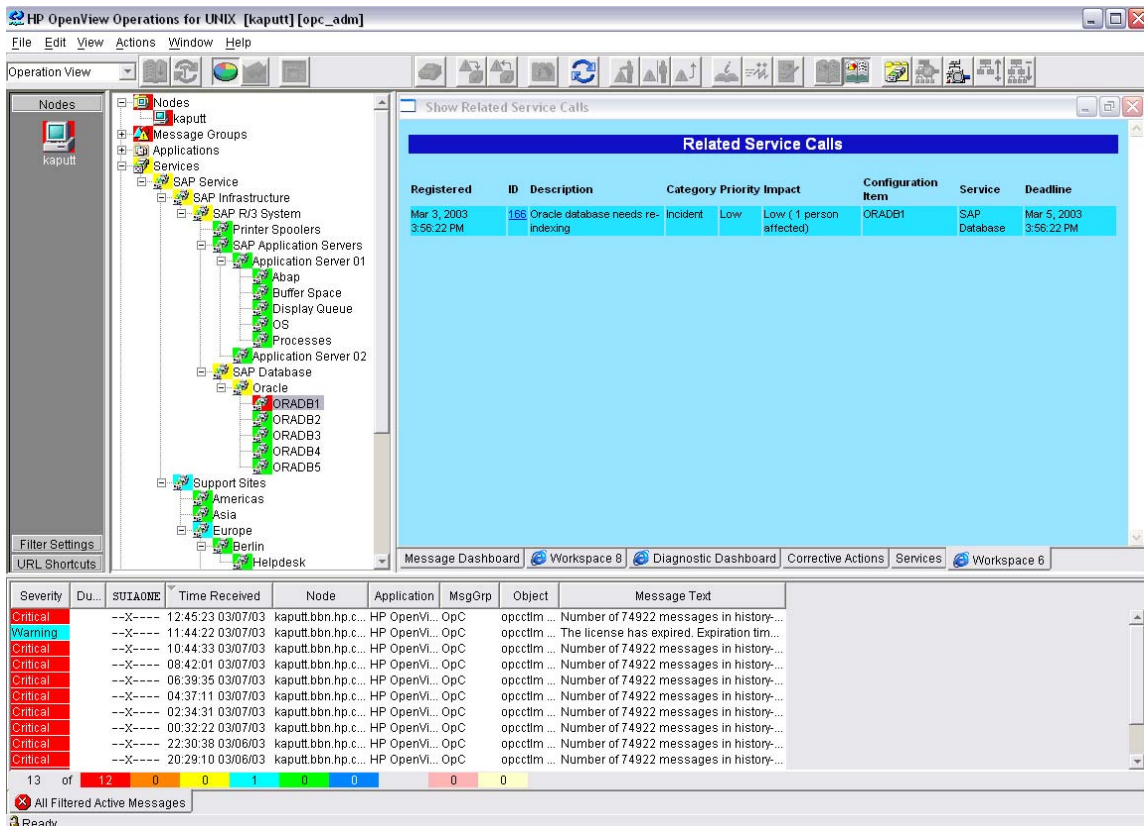


Figure 3-2 shows the Service Navigator operator console with Service Pages showing all service calls related to the service ORADB1. This is the result of the action shown in Figure 3-1 on page 73.

**Figure 3-2** Accessing Service Pages from the Service Navigator Operator Console



See also “How the Service Pages Integration Works” on page 75.

## How the Service Pages Integration Works

The Service Pages integration works on the basis of URL actions that call related service calls, incidents, problems, changes, or workorders for a selected Service Desk service or configuration item. These actions come with the Service Configuration application. All you have to do is install the Service Configuration module for the Service Pages integration on the Service Pages web server, and then enable the integration using the Service Configuration console. See “Configuring the Service Pages Integration” on page 76 for details.

When deploying a service hierarchy, the location of the Tomcat server is entered into the URL of these actions and the actions you have chosen are associated automatically with all Service Desk objects of the type service or configuration item contained in the deployed service hierarchy. Since the actions need to know the Service Desk object ID and object type, this information is added to each Service Desk service and configuration item in the form of user-defined attributes. The attribute `sdtargettype` contains the Service Desk object type, for example `service` or `ci`. The attribute `sdtargetoid` contains the Service Desk OID (object identifier), for example `00000000-0000-0000-0000-00002f720004`.

---

### NOTE

The operator has to supply the user name and password for a Service Desk account to be able to launch Service Pages. This information can be supplied each time Service Pages are launched, or it can be stored in the web server application file. See “Automating the Service Pages Authentication” on page 78 for more information.

---

See also “Configuring the Service Pages Integration” on page 76.

## Configuring the Service Pages Integration

The Service Pages integration is supported on all platforms and in all web browsers supported by the Service Navigator operator console. This includes web browsers that are external to the Java GUI such as Internet Explorer on Windows and Netscape Navigator on UNIX, as well as web browsers that are integrated into the Java GUI such as the embedded browser on both Windows and UNIX.

---

### NOTE

The Service Configuration server (the Service Desk application server) and the Service Pages web server (the Tomcat server) must be running when launching Service Pages from the Service Navigator operator console.

---

### To configure the Service Pages integration:

1. Install the Service Configuration integration module for Service Pages on the Service Pages web server. See the *Service Configuration for Service Navigator Installation Guide* for details about the installation.
2. Configure options for the Service Pages integration:
  - a. Select **Tools: Options** from the menu bar. The Options dialog box opens and displays the General tab.
  - b. Click the Integrations tab.
    - a. Specify the name and port of the Tomcat server for your Service Pages application. Enter the fully qualified hostname of the server and append the port number separated by a colon (:), as shown in the following example:  
  
Example: `myserver.bbn.hp.com`  
  
If you do not specify a name, the following defaults will be used:  
  
`localhost`
  - b. Choose the actions you want to enable for the Service Pages integration.
  - c. Click **OK** to save your changes and close the dialog box.

When deploying a service hierarchy, the location of the Tomcat server is entered into the URL of the actions and the actions you have chosen are automatically associated with all Service Desk objects of the type service or configuration item contained in the deployed service hierarchy.

---

**NOTE**

Only deploy configurations with the same Tomcat server to an OVO management server. If you deploy a configuration from a Service Configuration installation with a different Tomcat server, you will overwrite the actions that already exist on the OVO management server.

---

**TIP**

To use a different application server for Service Pages, edit the web server application file `web.xml` on the Service Pages web server. Change the name of the application server in the `sd_application_server` context. This should take the form of `Named|Machine Name.Domain.ITP number` (the name is optional). If the application server is on the same system as Service Pages, then `LOCALHOST` can be used for the application server hostname. See the *Service Desk Installation Guide* for more details.

---

See also “Automating the Service Pages Authentication” on page 78.

## Automating the Service Pages Authentication

The operator has to supply the user name and password for a Service Desk account to be able to launch Service Pages. This information can be supplied each time Service Pages are launched, or it can be stored in the web server application file.

### To automate the Service Pages authentication:

1. Locate the web server application file on your Service Pages web server:

```
<Install Dir>\webapps\ServicePages\WEB-INF\web.xml  
<Install Dir> is usually on Windows c:\Program Files\  
HP OpenView\www\
```

2. Add the following lines to the web.xml file and replace the values for the user name and password with your own values:

```
<context-param>  
  <param-name>login_username</param-name>  
  <param-value>YourUserName</param-value>  
  <description>Change YourUserName as required</description>  
</context-param>  
  
<context-param>  
  <param-name>login_password</param-name>  
  <param-value>YourPassword</param-value>  
  <description>Change YourPassword as required</description>  
</context-param>
```

3. Save the web.xml file and restart the tomcat server.

See also “Troubleshooting the Service Pages Integration” on page 79.

## Troubleshooting the Service Pages Integration

Sometimes Service Pages cannot be launched because of one of the following reasons:

❑ **No related data found.**

If no Service Desk data is related to the selected service or configuration item, you receive a message similar to the following:

```
No Service Calls related to Service Abap found.
```

You can safely ignore this message. There are simply no service calls related to the currently selected service.

❑ **Session timeout.**

An error message similar to the following appears:

```
Your session on the server has timed out. Click OK to log on, then try again.
```

This message appears when you are working with multiple Service Pages windows for different services or configuration items. If you change and submit data in one window, the session also closes for all other open windows. You can still view the data in the other windows, but it is no longer possible to submit changes in them.

❑ **Service Pages login.**

You are asked to log in to Service Pages each time you request data from Service Desk unless you have automated the authentication process as described in “Automating the Service Pages Authentication” on page 78. You also have to enter your user name and password after a Service Pages session has timed out as described above.

❑ **Access denied.**

Access to Service Pages is denied if you entered an incorrect user name or the password. Enter the correct user name and password.

❑ **Error when Tomcat server is not running.**

An error message similar to the following appears:

Action canceled. Internet Explorer was unable to link to the Web page you requested. The page may be temporarily unavailable.

This indicates that the Service Pages web server (the Tomcat server) is not running. Start the web server and try again.

❑ **Embedded web browser cannot display a page.**

An error message similar to the following appears:

The page cannot be displayed. The page you are looking for is currently unavailable. The Web site may be experiencing technical difficulties, or you may need to adjust your browser settings.

If the embedded web browser displays the error message described above, it probably cannot connect to the Internet. Check the proxy settings of the embedded web browser in the Preferences dialog box of the OVO Java GUI.

See also “About the Service Pages Integration” on page 73.



## **Service Desk Data Form Integration for the OVO Operator**

This section describes the following aspects of the data form integration:

- ❑ **About the Service Desk Data Form Integration**  
See “About the Service Desk Data Form Integration” on page 82.
- ❑ **How the Data Form Integration Works**  
See “How the Data Form Integration Works” on page 84.
- ❑ **Configuring the Data Form Integration**  
See “Configuring the Data Form Integration” on page 85.
- ❑ **Troubleshooting the Data Form Integration**  
See “Troubleshooting the Data Form Integration” on page 86.

## About the Service Desk Data Form Integration

With the Service Desk data form integration configured, operators using the Service Navigator operator console can start the Service Desk client console in the context of a particular Service Desk object. The Service Desk client console displays detailed information about the selected object in a Service Desk data form.

Figure 3-3 shows how access to Service Desk data forms is integrated into the shortcut menu of objects in Service Navigator.

**Figure 3-3** Launching a Data Form from the Service Navigator Console

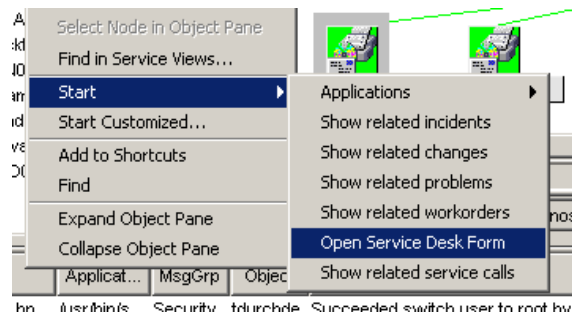


Figure 3-4 shows a Service Desk data form for the service Desktop.

**Figure 3-4** Opening a Service Desk Data Form from the Service Navigator Operator Console

The screenshot shows a window titled "10 - Service" with a menu bar (File, Edit, View, Tools, Actions, Help) and a toolbar. The main area is divided into several sections:

- Form Fields:**
  - ID: 10
  - Name: Desktop
  - Description: Desktop Applications
  - Service type: Business (selected)
  - Status: Supported (selected)
  - Folder: [empty]
  - Parent service: [empty]
  - Service Mana...: Rousel, Alan
  - Default Work...: Desktop Specialists
  - Checklist: New Employee
- Configuration Items:** A table with columns: Search code, Name 1, Location, Category, and Un... (checkbox).

| Search code   | Name 1                    | Location | Category       | Un...                    |
|---------------|---------------------------|----------|----------------|--------------------------|
| MSOFFICE2000  | Microsoft Office 2000     |          | Application    | <input type="checkbox"/> |
| MSOUTLOOK2000 | Microsoft Outlook 2000    |          | Communicati... | <input type="checkbox"/> |
| PCKAY007      | PCKAYAK P3 900 Mhz        | USA      | Business PC    | <input type="checkbox"/> |
| PCKAY008      | PCKAYAK P3 900 Mhz        | USA01    | Business PC    | <input type="checkbox"/> |
| PCKAY006      | PCKAYAK P3 900 Mhz        | USA03    | Business PC    | <input type="checkbox"/> |
| PCKAY005      | PCKAYAK P3 900 Mhz        | USA01    | Business PC    | <input type="checkbox"/> |
| PCKAY004      | PCKAYAK P3 900 Mhz        | USA01    | Business PC    | <input type="checkbox"/> |
| PCKAY003      | PCKAYAK P3 900 Mhz        | EUR002   | Business PC    | <input type="checkbox"/> |
| PCKAY002      | PCKAYAK P3 900 Mhz        | USA03    | Business PC    | <input type="checkbox"/> |
| PCKAY001      | PCKAYAK P3 900Mhz         | USA02    | Business PC    | <input type="checkbox"/> |
| OSWIN95       | Microsoft Windows 2000    |          | OS (Operati... | <input type="checkbox"/> |
| OSNT4.0       | Microsoft Windows NT...   |          | OS (Operati... | <input type="checkbox"/> |
| MSXCEL2000    | Microsoft Excel 2000      |          | Spreadsheet    | <input type="checkbox"/> |
| MSWORD2000    | Microsoft Word 2000       |          | Word proce...  | <input type="checkbox"/> |
| MSPDWERP2000  | Microsoft Powerpoint 2... |          | Presentation   | <input type="checkbox"/> |
| PCKAY010      | PCKAYAK P3 900 Mhz        | USA03    | Business PC    | <input type="checkbox"/> |
| PCKAY009      | PCKAYAK P3 900 Mhz        | USA03    | Business PC    | <input type="checkbox"/> |
- Buttons:** Add..., Modify..., Relate..., Unrelate
- Service Navigator:** SN Status: Undefined, SN Name: [empty], SN Socket Address: [empty]

See also “How the Data Form Integration Works” on page 84.

## How the Data Form Integration Works

The data form integration works on the basis of a program action that launches the Service Desk client console for a selected Service Desk object and opens a data form with detailed information about the object. The action is part of the Service Configuration application. All you have to do is to enable the integration using the Service Configuration console. See “Configuring the Data Form Integration” on page 85 for details.

When deploying a service hierarchy, the data form action is associated automatically with all Service Desk objects contained in the deployed service hierarchy. Since the action needs to know the Service Desk object ID and object type, this information is added to each Service Desk object in the form of user-defined attributes. The attribute `sdtargettype` contains the Service Desk object type, for example `service` or `ci`. The attribute `sdtargetoid` contains the Service Desk object identifier (OID), for example `3098279942`.

---

### NOTE

The operator does not have to supply a user name and password if the Service Desk client is connected correctly with the server. The Service Desk client is launched using a batch file that launches the client console with a command line tool.

---

See also “Configuring the Data Form Integration” on page 85.

## Configuring the Data Form Integration

This integration requires an installation of the Service Desk client on the same system as the Service Navigator console. Since the Service Desk client console is only available on Windows, the integration can only run on this platform.

---

### NOTE

The Service Configuration server (the Service Desk application server) must be running when launching a Service Desk data form from the Service Navigator operator console.

---

### To configure the data form integration:

1. Ensure that a Service Desk client is installed on the system where the Service Navigator console is installed. You may need to update the Service Desk client with a Service Desk service pack so that it has the same version as the application server.
2. Configure options for the Service Pages integration:
  - a. Select **Tools: Options** from the menu bar. The Options dialog box opens and displays the General tab.
  - b. Click the Integrations tab.
  - c. Select the Service Desk data form integration check box to enable the data form integration.
  - d. Click **OK** to save your changes and close the dialog box.

When deploying a service hierarchy, the data form action is automatically associated with all Service Desk objects contained in the deployed service hierarchy. Data forms can be launched for all Service Desk object types.

See also “Troubleshooting the Data Form Integration” on page 86.

## Troubleshooting the Data Form Integration

Sometimes data forms cannot be launched because of the error when application server is not running.

An error message similar to the following appears:

```
Failed to connect with server. Check that the application
server is running.
```

If you receive this error message, start the application server.

---

**TIP**

On standard Windows installations it is not required that the Service Desk client is running when attempting to open data forms. However, on some Windows installations this may be necessary to successfully launch data forms from with the Service Navigator operator console.

---

See also “About the Service Desk Data Form Integration” on page 82.

## Web Console Integration for the OVO Operator

This section describes the following aspects of the Web Console integration:

- ❑ **About the Web Console Integration**  
See “About the Web Console Integration” on page 88.
- ❑ **How the Web Console Integration Works**  
See “How the Web Console Integration Works” on page 91.
- ❑ **Configuring the Web Console Integration**  
See “How the Web Console Integration Works” on page 91.

### Related Topics

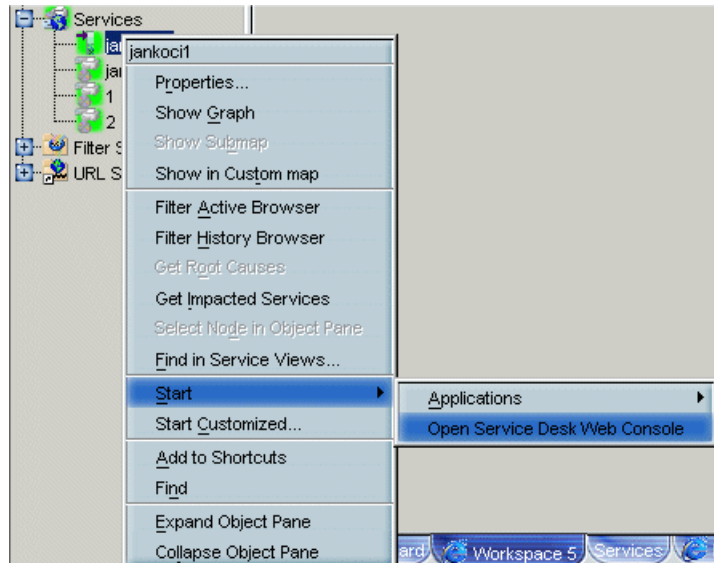
Service Desk Data Form Integration for the OVO Operator  
About the Service Desk Integration for the OVO Operator  
Service Pages Integration for the OVO Operator

## About the Web Console Integration

With the Web Console integration configured, operators can launch Service Desk Web Console in the context of particular Service Desk object. Service Desk Web Console is easy to launch from the Service Navigator console: operators only need to access the shortcut menu of the Service Desk service or configuration item, and select the Open Service Desk Web Console action from the menu. This action opens a Web Console form which displays the properties of the selected Service Desk object in the Service Navigator.

Figure 3-3 shows how access to Web Console forms is integrated into the shortcut menu of objects in Service Navigator.

**Figure 3-5** Launching a Web Console Form in the Service Navigator



In the Web Console Form, operators can run a number of actions defined in Service Desk, or configure them by themselves. These actions, as distinguished from actions used in Service Pages integration, does not come with the Service Configuration application. They are defined in the Service Desk, and can also be changed or deleted, therefore it is not possible to manage them from Service Configuration console. See Service Desk documentation for more information about defining actions in Service Desk.



Figure 3-6 shows a Web Console form for the service APIT04.0 displayed in the Service Navigator workspace. The All calls for this CI action is chosen.

**Figure 3-6** Choosing an Action from the Web Console Form in the Service Navigator

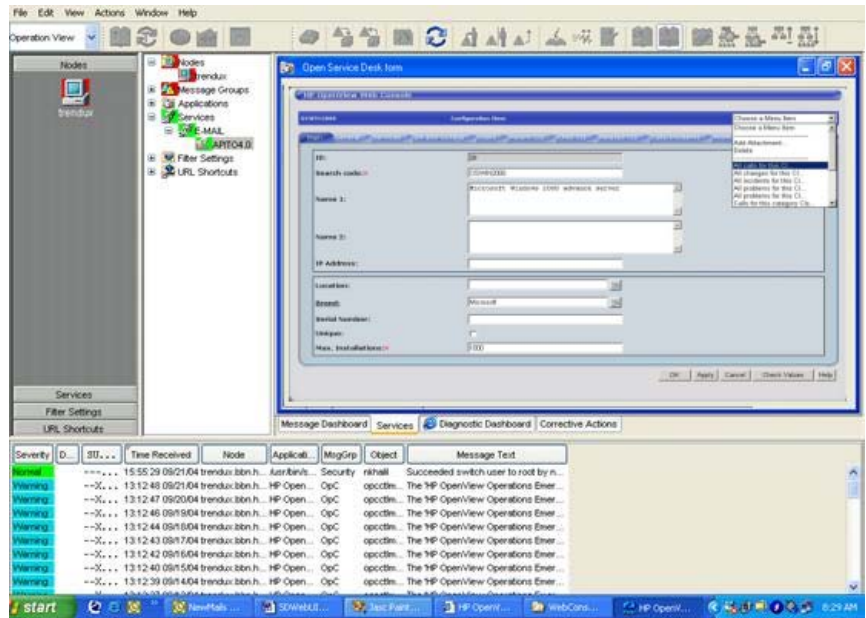
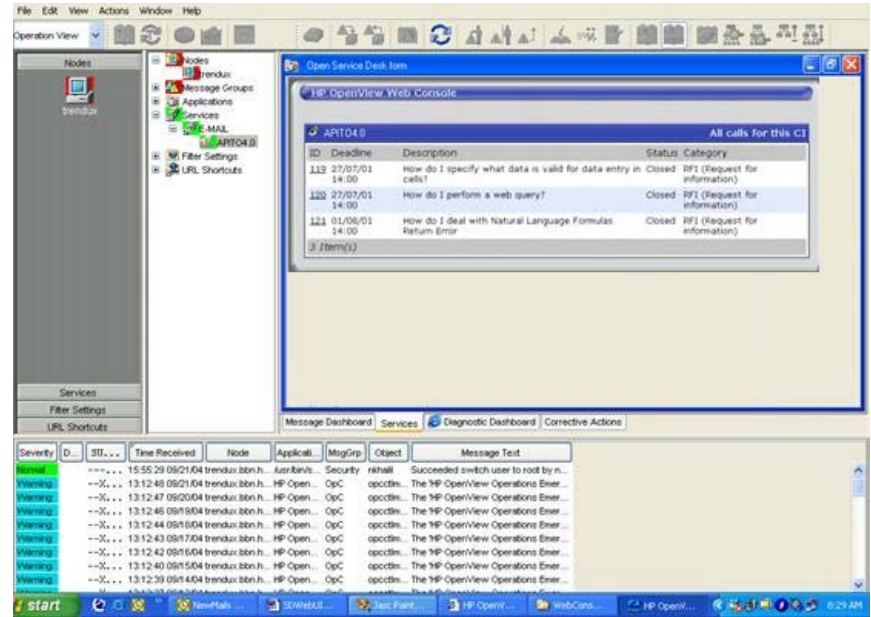


Figure 3-7 shows the result of the All calls for this CI action performed in the Service Desk Web Console.

**Figure 3-7** Result of All calls for this CI Action in the Web Console



**Related Topics**

- How the Web Console Integration Works
- Configuring the Web Console Integration
- Service Pages Integration for the OVO Operator
- About the Service Desk Integration for the OVO Operator
- Service Desk Data Form Integration for the OVO Operator
- CadmSDIntegDataFormTrouble

## How the Web Console Integration Works

The Web Console integration works on the basis of URL action that launches the Service Desk Web Console and opens a Web Console form which displays the properties of a selected Service Desk object, and also offers a possibility to start a number of actions previously defined in Service Desk. All overview and smart actions defined in Service Desk can be started from the Web Console in the context of the selected Service Desk object in the Service Navigator. See the Service Desk documentation for more information about defining actions in Service Desk.

To use Service Desk Web Console integration, you have to select this integration type and specify the Service Desk web server name. Also supply the username and password for a Service Desk account to be able to launch Service Desk Web Console. This information is specified in the Service Configuration console. See “Configuring the Web Console Integration” on page 92 for more information.

If you fail to specify the username and password for a Service Desk account in the Service Configuration console, you can do that from the Service Navigator console. Namely, when you start the Open Service Desk Web Console action from the shortcut menu of the selected Service Desk object, a web page containing fields where you can specify username and password is displayed in the workspace of the Service Navigator. After successful authorization, the Web Console form for the selected object opens.

See also “Configuring the Web Console Integration” on page 92.

### Related Topics

- About the Web Console Integration
- Configuring the Web Console Integration
- Service Pages Integration for the OVO Operator
- About the Service Desk Integration for the OVO Operator
- Service Desk Data Form Integration for the OVO Operator

CadmSDIntegDataFormTrouble

## Configuring the Web Console Integration

The Web Console integration is supported on all platforms and in all web browsers supported by the Service Navigator operator console. This includes web browsers that are external to the Java GUI such as Internet Explorer on Windows and Mozilla on UNIX, as well as web browsers that are integrated into the Java GUI such as the embedded browser on both Windows and UNIX.

---

**NOTE**

The Service Configuration server (the Service Desk application server) and the Service Desk web server must be running when launching Service Desk Web Console from the Service Navigator.

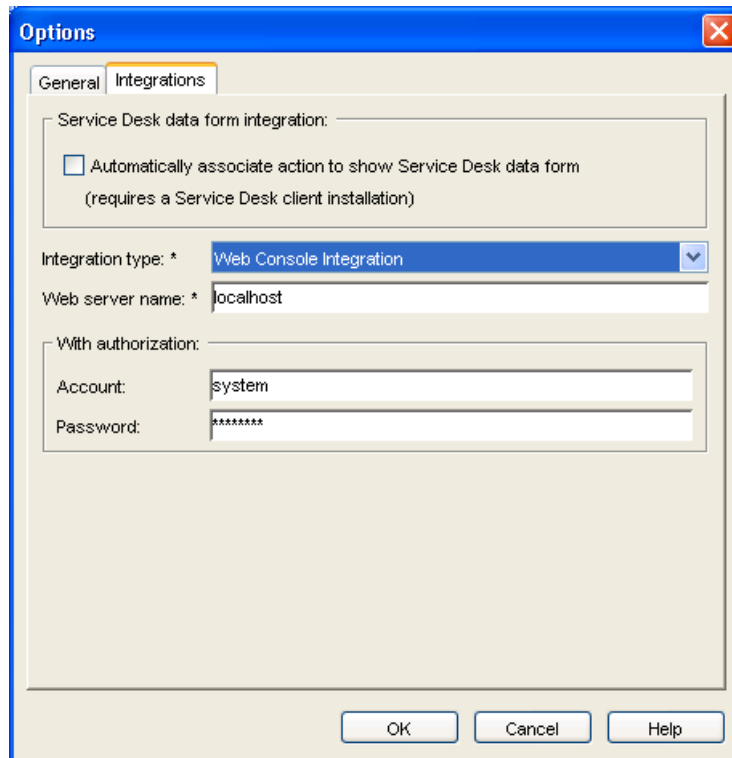
---

### To configure the Web Console integration:

1. Select **Tools: Options** from the menu bar. The Options dialog box opens and displays the General tab.
2. Click the Integrations tab.
3. Select Web Console Integration from Integration type drop-down list.
4. Specify the Web server name.
5. Specify Service Desk account and password for the web server.
6. Click **OK** to save your changes and close the dialog box.

Figure 3-8 shows an Integrations tab with Web Console Integration options configured.

**Figure 3-8** Web Console Integration Options



See also “About the Web Console Integration” on page 88.



---

# Glossary

## A

**action** An operation that can be carried out by an OpenView application. Actions are typically performed on managed objects. Actions can be manually executed by users through a menu item or toolbar buttons. Actions can also be configured to automatically occur in response to an event, message, or a change in information in the management database.

**activate** To make active or functional.

**adapter** Software that allows interoperability between two or more software products or components.

**advanced customization** Less common types of customization which are more flexible in their capabilities and complex in their implementation than typical customizations. As with other customizations, advanced customizations are done to meet the needs and preferences of a particular customer or user.

**agent** A program or process running on a remote device or computer system that responds to management requests, performs management operations, and/or sends performance and event notification. An agent can provide access to managed objects and MIB variables, interpret policy for resources and do configuration of resources.

**API** See *application program interface (API)*.

**application** Packaged software that provides functionality that is designed to accomplish a set of related tasks. An application is generally more complex than a tool.

**application program interface (API)** An interface that enables programmatic access to an application.

**ASCII** American Standard Code for Information Interchange.

**attribute** An attribute is a characteristic or property of an object that can be described through a name value pair. An example of an attribute for a person would be Name: John Smith.

**attribute name-value pair** An attribute name-value pair is a combination of an attribute identifier and the value of that attribute for a specific object. An example of an attribute name-value pair for a person would be Name: John Smith.

## B

**badge** A miniature icon, placed on or near an object's main icon, that provides information about the object's characteristics.

**Boolean operator** A logical operator that defines the context in which attribute values will be compared to satisfy a query or policy. The operators tell the system that:

**AND** Both conditions have to be satisfied.

**OR** At least one condition has to be satisfied.

**NOT** No instance of this condition is allowed.

**browser** A module within a workspace that presents one or more views of objects and provides functionality for interacting with the objects and the views.

## C

**child object** An entry in an object hierarchy that has a relationship to one or more parent objects within the hierarchy. A child object can at the same time be the parent of one or more child objects.

**client** A program or executable process that requests a service from a server

**client** A computer system on a network that accesses a service from another computer (server).

**client console** An instance of the user interface that appears on the client system while the application runs on a server.

**command** An instruction to a computer program that causes a specified operation to be carried out. Commands are typically typed by users on a command line.

**configuration** The combination of settings of software parameters and attributes that determine the way the software works, the way it is used, and/or how it appears. The combination of settings of software parameters and attributes that determine the way the software works, the way it is used, and/or how it appears.

**configuration file** A file that contains specifications or information that can be used for determining how a software program should look and operate.



**configure** To define and/or modify specified software settings to fulfill the requirements of a specified environment, application and/or usage.

**console** An instance of the user interface from which the user can control an application or set of applications.

**customer** An entity (person or organization) that receives services offered by a service provider. Typically services are provided to a customer based on a contractual relationship.

**customization** The process of designing, constructing and/or modifying software to meet the needs and preferences of a particular customer or user.

**customize** To design, construct and/or modify software to meet the needs and preferences of a particular customer or user.

## D

**database** A repository of data that is electronically stored. Typically databases are organized so that data can be retrieved and updated.

**deactivate** To deliberately stop a component or object from working.

**deploy** To install and activate software, hardware, capabilities, and/or services so that they work in the business environment.

**deployment** The process of installing and activating software, hardware, capabilities and/or services so that they work in the business environment.

**double-click** To press and release a pointing device's button twice in rapid succession. Double-clicking is a time-dependent action. Clicking twice in the same location at slow speed (click-delay-click) is not a double-click.

**dynamic parameters** Parameters whose values are determined during program execution.

## E

**error log** An output file containing error messages.

**export** To format and move information from the current application to a location outside the current application.

## **F**

**filter** A software feature or program that functions to screen data so that only a subset of the data is presented or passed. Filters allow matching-relevant information to be extracted and acted on while non-matching-irrelevant information is held back.

**find** The act of seeking of specific data or objects within the management application or set management applications based on specified criteria.

## **G**

**group** A collection of elements that can be treated as a unit. The set of elements in a group may be based on a rule specifying characteristics of group members or it may be arbitrary based on user selection. Assignments of associations and parameters can be made at the group level and inherited by the members of the group.

## **H**

**hierarchy** Elements organized in successive levels with each lower level being subordinate to the one above.

**HP OpenView** A family of network and system management products, and an architecture for those products. HP OpenView includes development environments and a wide variety of management applications.

## **I, J, K**

**icon** An on-screen image that represents objects that can be monitored or manipulated by the user or actions that can be executed by the user.

**identifier** A name that within a given scope that uniquely identifies the object with which it is associated.

**import** To format and move information from a location outside the current application into the current application.

**install** To load a product or component of a product onto a computer system or other network or system device. Installation typically involves running initial configuration scripts that are part of the installation process.

## **L**

**list** A set of selectable items.

## **M**

**managed node** A computer system or device in a network that is both monitored for performance, status and messages and is manipulated by means of actions in the management software.

**managed object** A network, system, software or service object that is both monitored for performance, status and messages and is manipulated by means of actions in the management software.

**management server** A server that provides management services, processes, and/or a management user interface to clients.

**map** A view of objects in the management environment (for example, systems, services, devices) in a format that shows one or more topological relationships between the objects.

**message** A structured, readable notification that is generated as a result of an event, the evaluation of one or more events relative to specified conditions, or a change in application, system, network, or service status.

**message browser** A graphical user interface that presents notifications that are generated as a result of an event, the evaluation of one or more events relative to specified conditions or a change in application, system, network, or service status.

**message description** Detailed information about an event or message.

**message severity level** A property of a message indicating the level of impact of the event or notification that initiated the message.

**metadata** Data that defines data.

**monitored object** A network, system, software or service object that is monitored for performance, status and messages by the management software.

Note that monitored objects cannot be manipulated by the management software.

## N

**node** A computer system or device (for example, printer, router, bridge) in a network.

## O

**object** A logical or physical entity that can be described by a distinct named set of attributes. Examples of objects are a customer, a service, a network, a computer, an interface, or a process.

**object hierarchy** Specification of parent-child relationships (in terms of grouping, not inheritance) to create a tree-like structure and the specification of views for each object in the structure. Each parent in the object hierarchy can have more than one child, and each child can have one or more parents.

**object identifier (OID)** A unique sequence of numbers or string of characters used for specifying the identity of an object. This identifier is obtained from either an authorized registration authority or an algorithm designed to generate universally unique values.

**object rule** A rule that determines what conditions must be met before an object will be classified as a child object or a parent in an object hierarchy.

**object type** An abstraction or categorization of objects based on the attributes of the object. Examples of object types are services, customers, hardware elements.

**operator** The role of a person who uses a management application to perform a specified subset of management tasks.

**operator console** A management console that is constrained to provide functionality that is consistent with operator level tasks.

## P, Q

**parameter** A variable or attribute that may be given an arbitrary value for use during an execution of either a computer program or a procedure within a program.

**parameter group** A set of parameter values that have been categorized together on some logical basis. A parameter group can be applied to a variable as a set allowing multiple values to be assigned with one operation.

**parameter type** An abstraction or categorization of a parameter that determines the particular kind of data that is valid for the parameter. For example a parameter type could be IP Address which indicates that parameter values must have 4 numbers separated by decimals with the value for each number being in the range of 0 to 255.

**parameter value** A value that is given to a variable.

**parent object** An entry in an object hierarchy that has one or more child objects in the hierarchy. A parent object can at the same time be the child of another parent object.

**path** A sequence of directory names separated by slashes that specifies the location of any file or directory.

**port** A location for passing information into and out of a network device.

**process** A coordinated set of activities directed to achieve a business goal.

## R

**refresh** To bring up to date. The process of refreshing synchronizes the data that is loaded in a client console with the data available on the server. Refreshing includes bringing in new data as well as removing obsolete data.

**reserved word** Words that have special meaning in Service Configuration and cannot be used for any other kind of identifier.

**root object** Topmost object or objects in an object hierarchy. A root object does not have a parent object.

## S

**server** A program or executable process that responds to and services requests issued by clients.

**server** A computer system that provides a service (for example, management capabilities, file storage capabilities) to other computer systems (clients) on the network.

**service** A customer-based or user-oriented functionality (for example, E-mail, network bandwidth, application access) provided by a set of service resources.

**service contract** A contract between a service provider and a customer that is designed to create a common understanding about services, priorities, responsibilities, and price. Service contracts may or may not include Service Level Agreements (SLAs).

**service hierarchy** A type of object hierarchy with a service being the root object.

**service level agreement (SLA)** A negotiated agreement between the service provider and the customer that contains agreed upon Service Level Objectives ( performance, availability, and so on) that specify the Quality of Service (QoS) to be maintained. It also contains processes and procedures to be followed when the objective is not met.

**service level management** The set of management functions that enables the process of measuring, reporting, and improving the quality of service being provided.

**service level objective (SLO)** A component within an SLA that specifies a Quality of Service metric and value that is to be met subject to specified constraints (for example, in a specified time frame).

**service model** A representation of the relationship between components of a service.

**service offering** A collection of different services that make up the products offered by a service provider. For example, a service offering can be a single service such as email or a bundled service like “web-hosting” that consists of the services: “web page hosting”, “e-mail”, and “customer help desk”.

**service provider** A company or organization that provides services to customers.

**service report** A report that provides information about the functioning and status of one or more services.

**service resource** A hardware or software component or an underlying communication medium that is used to provide a service.

**severity level** A property of an object indicating the status of the object. Severity level is based on the impact of events or messages associated with the object.

**Smart Plug-in (SPI)** Prepackaged software that installs into a management console and provides management capabilities specific to a given type of business application, database, operating system, or service.

**status calculation** Process used to determine the status of a service or object based on the status of the source objects that contribute to the status of the service or object.

**status propagation** The way that the status of a status source is defined for status calculation of an object.

**status propagation rule** An algorithm or specification that determines how a status source will impact the overall status of another object (for example, service, group).

**status source** An object or objects (for example, message, service, node, event) whose severity level or status contributes to the status of another object.

**syntax** The rules governing the structure and content of a language or the description of an object.

**system administrator** The role of a person who does configuration and maintenance on a computer system and/or the software on the system.

## T

**tab** A page in the user interface that has a small index-card like projection. The projection typically presents the name for the page and allows navigation to the page by clicking.

**target** A network element, system, device, interface or software component to which management software or policies will be deployed, or on which actions will be executed.

**terminate** To permanently stop the execution of a computer operation.

## U

**UI server** A server that can provide user interface components and data to multiple client consoles.

**uninstall** To remove previously installed software.

**user** A person who uses a software application, a computer or service. A user can have one or more roles. For example, a network administrator or operator role.

**user configuration file** A configuration file that contains specifications or information that can be used for determining how a software program should look and operate for a given user.

**user ID** An identifier that uniquely identifies a principal that is a person.

## V

**view** A data presentation usually in graphical or tabular format. Views allow users to obtain information on the objects (for example, status) and may allow interaction with the objects.

**view type** A categorization of the presentation format for a view. View types include table, chart, graphical, tree, and explorer.



## **W, Z, Y, Z**

**workspace** A working area within the OpenView Console that provides views of objects and access to functionality from the perspective of a given task domain. The objects presented and the actions that are most readily available are scoped by the context of the task domain.



**A**

- about
  - associating service hierarchy rules, 49
  - child object rules, 17
  - parent-and-child object rules, 19
  - Service Desk data form integration, 82
  - Service Desk objects, 14
  - service hierarchy rules, 15
  - Service Pages integration, 73
  - Web Console integration, 88
- adding
  - child objects
    - from Service Desk, 14
- applying
  - service hierarchy rules, 54
- associating
  - child object rules, 50
  - parent-and-child object rules, 52
  - service hierarchy rules, 49
- associations tab
  - filter for service hierarchy rules, 44
- attributes tab
  - filter for service hierarchy rules, 42
- authentication, Service Pages, 78

**C**

- child object rules
  - about, 17
  - associating, 50
  - creating, 32
  - deleting, 34
  - editing, 33
  - managing, 30
  - viewing, 31
- child objects
  - adding
    - from Service Desk, 14
- configuring
  - Service Desk data form integration, 85
  - Service Pages integration, 76
  - Web Console integration, 92
- creating
  - child object rules, 32
  - parent-and-child object rules, 37
- creating service hierarchies, 27

**D**

- default properties
  - for service hierarchy rules, 46

- deleting
  - child object rules, 34
  - parent-and-child object rules, 40

**E**

- editing
  - child object rules, 33
  - parent-and-child object rules, 39
- examples
  - service hierarchy rules, 21

**F**

- filters
  - service hierarchy rules
    - by associations, 44
    - by attributes, 42
    - specifying, 41

**I**

- Index, 107
- integration
  - Service Desk, 11
    - for OVO operator, 70
  - Service Desk data form, 82
    - configuring, 85
    - described, 84
    - for OVO operator, 81
    - troubleshooting, 86
  - Service Pages
    - about, 73
    - configuring, 76
    - described, 75
    - for OVO operator, 72
    - troubleshooting, 79
  - Web Console
    - about, 88
    - configuring, 92
    - described, 91
    - for OVO operator, 87

**M**

- managing
  - child object rules, 30
  - parent-and-child object rules, 35
- mapping objects to Service Desk objects
  - change, 66
  - create, 66
  - overview, 64

---

# Index

remove, 67  
monitoring  
  application of service hierarchy rules, 55

## O

objects  
  child, adding Service desk objects as, 60  
  Service Desk  
    about, 14  
    adding multiple to different service hierarchies, 63  
    adding multiple to same service hierarchy, 62  
    mapping, 64  
    types, 14  
OVO  
  operator  
    Service Desk integration for, 70  
    Service Pages integration for, 72  
    Web Console integration for, 87  
  operator, Service Desk data form integration for, 81

## P

parent-and-child object rules  
  about, 19  
  adding with Service Hierarchy Wizard, 57  
  adding with wizard, 57  
  associating, 52  
  creating, 37  
  deleting, 40  
  editing, 39  
  managing, 35  
  viewing, 36  
properties  
  default for service hierarchy rules, 46  
  service hierarchies, 59

## S

Service Desk, 27  
  child object rules  
    about, 17  
    associating, 50  
    creating, 32  
    deleting, 34  
    editing, 33  
    managing, 30  
    viewing, 31

data form integration, 82  
  configuring, 85  
  described, 84  
  for OVO operator, 81  
  troubleshooting, 86  
integration, 11  
  for OVO operator, 70  
objects  
  about, 14  
  adding manually as child objects, 60  
  adding multiple to different service hierarchies, 63  
  adding multiple to same service hierarchy, 62  
  adding rule-based, 14  
  types, 14  
objects, mapping, 64  
parent-and-child object rules  
  about, 19  
  adding with wizard, 57  
  associating, 52  
  creating, 37  
  deleting, 40  
  editing, 39  
  managing, 35  
  viewing, 36  
service hierarchy rules  
  about, 15  
  adding with wizard, 56  
  applying, 54  
  associating, 49  
  association filter, 44  
  attribute filter, 42  
  examples, 21  
  setting default properties, 46  
  specifying filters, 41  
  working with, 28  
service hierarchies  
  about Service Desk objects, 14  
  creating, 27  
  objects  
    adding multiple to different hierarchies, 63  
    adding multiple to same hierarchy, 62  
  properties, 59  
  Service Desk object types, 14  
service hierarchy rules  
  about, 15  
  adding with wizard, 56

- applying
  - about, 54
  - monitoring progress, 55
  - stopping, 55
- associating
  - about, 49
  - child object rules, 50
  - parent-and-child object rules, 52
- child object rules, 17
- examples, 21
- filters
  - by associations, 44
  - by attributes, 42
  - specifying, 41
- parent-and-child object rules, 19
- setting default properties, 46
- working with, 28
- Service Hierarchy Wizard
  - adding service hierarchy rules, 56
- Service Pages
  - authentication, automating, 78
  - integration
    - configuring, 76
    - described, 75
    - for OVO operator, 72
- Service Pages integration
  - troubleshooting, 79
- specifying filters
  - for service hierarchy rules, 41
- stopping
  - application of service hierarchy rules, 55
  
- T**
- troubleshooting
  - Service Desk data form integration, 86
  - Service Pages integration, 79
- types of Service Desk object, 14
  
- V**
- view builder. *See* service hierarchy rules
- viewing
  - child object rules, 31
  - parent-and-child object rules, 36
  
- W**
- Web Console
  - integration
    - for OVO operator, 87
- Web Console integration
  - configuring, 92
  - described, 91
  - working with
    - service hierarchy rules, 28

