

Internal Service Requests



HELP.COOMOPA

Release 4.6C



Copyright

© Copyright 2001 SAP AG. All rights reserved.

No part of this publication may be reproduced or transmitted in any form or for any purpose without the express permission of SAP AG. The information contained herein may be changed without prior notice.

Some software products marketed by SAP AG and its distributors contain proprietary software components of other software vendors.

Microsoft[®], WINDOWS[®], NT[®], EXCEL[®], Word[®], PowerPoint[®] and SQL Server[®] are registered trademarks of Microsoft Corporation.

IBM[®], DB2[®], OS/2[®], DB2/6000[®], Parallel Sysplex[®], MVS/ESA[®], RS/6000[®], AIX[®], S/390[®], AS/400[®], OS/390[®], and OS/400[®] are registered trademarks of IBM Corporation.

ORACLE[®] is a registered trademark of ORACLE Corporation.

INFORMIX[®]-OnLine for SAP and Informix[®] Dynamic Server[™] are registered trademarks of Informix Software Incorporated.

UNIX[®], X/Open[®], OSF/1[®], and Motif[®] are registered trademarks of the Open Group.

HTML, DHTML, XML, XHTML are trademarks or registered trademarks of W3C[®], World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA[®] is a registered trademark of Sun Microsystems, Inc.

JAVASCRIPT[®] is a registered trademark of Sun Microsystems, Inc., used under license for technology invented and implemented by Netscape.

SAP, SAP Logo, R/2, RIVA, R/3, ABAP, SAP ArchiveLink, SAP Business Workflow, WebFlow, SAP EarlyWatch, BAPI, SAPPHIRE, Management Cockpit, mySAP.com Logo and mySAP.com are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other products mentioned are trademarks or registered trademarks of their respective companies.

Icons

Icon	Meaning
	Caution
	Example
	Note
	Recommendation
	Syntax

Inhalt

Internal Service Requests	6
Preparing The Internal Service Request	8
Definition Of A Scenario	9
Scenario	10
Cost-Incurring Scenarios	11
Easy Cost Planning and Execution Services.....	12
Using Easy Cost Planning	15
Executing Ad Hoc Cost Estimates.....	20
Triggering the Execution Services.....	22
Template for Easy Cost Planning.....	25
Creating Templates	27
Notification.....	28
Notification Type	29
Task.....	30
Entering Tasks	31
Follow-Up Actions for Tasks	32
Action Box	33
Standard Follow-Up Functions.....	36
Activity.....	38
Entering Activities	39
Defining Scenarios	40
Business Add-Ins	44
Definition Of A Form	46
Preparing The Internet Service	47
Generating Services	48
Publishing a Service	50
Executing a Service	51
Form Definition.....	53
Define Form	54
Example: HTML Framework for HTML Templates.....	60
Example: Specific Form Elements in HTML Templates (SAP Design)	61
Example: Specific Form Elements in HTML Templates (SAP Design)	66
Web Application Builder.....	70
Creating HTML Templates.....	71
Extending HTML Templates	74
Adding MIME Objects	75
Entering An Internal Service Request.....	77
Searching For Solutions: Flow	78
Request Without Approval Requirement: Flow.....	81
Request With Approval Requirement: Flow	84
Approval Requirement for Notifications	87
Scenarios Provided	88
General Request.....	89

Request For Office Removal.....	90
Request For Budget Increase	91
Request For Master Data Change	92
Request For An Adjustment Posting.....	93
Change Notification	94
Processing The Internal Service Request	95
Processing Notifications	96
Process Notification	97
Processing Notifications Using Worklist.....	99
Select and Process Notifications Using Worklist	100
Processing Notifications Using Workflow.....	104
Partners in Notification Processing	105
Processing the Partner Information	107
Task Processing	108
Process Specific Task in a Notification	109
Processing Tasks Using Worklist.....	111
Select and Process Tasks Using Worklist	112
Processing Tasks Using Workflow.....	116
Status Query	117
Status Management for Notifications	118
Displaying Status Information	119
Assignment of System Statuses	121
Assigning and Changing User Statuses	122
Status Management for Tasks	123
Document Flow.....	125
Displaying the Document Flow for a Notification	127
Example: Entering And Processing An Internal Service Request	128

Internal Service Requests

Use

Internal service requests enable you to request and process any service using a simple form, or a freely formulated text. You do not require SAP R/3 experience. This kind of request action could be a removal for example. The request is forwarded to the approving manager or the processor. If the service request incurs costs, you can send the request to a cost center manager or budget manager (who can execute an automatic cost calculation) for approval.

This function enables you to process numerous requests in the intranet that you previously processed by mail or telephone. You can enter the requests at any time of day or night, every day of the week. All those involved can also call up the status at any time. The integrated cost posting ensures that the data is transferred without errors to Controlling, and the costs of the line item reports are monitored.

The internal service request covers three business processes:

- [Finding solutions to a particular problem \[Page 78\]](#)
- [Having the non-cost incurring request \(such as: removal request\) entered and processed \[Page 81\]](#)
- [Having the cost incurring request \(such as: budget change\) entered and processed \[Page 84\]](#)

Advantages

You can create your own request forms without ABAP skills

You have **one** initial screen for entry and status query of the request.

The system finds the processors for the request using the details in the request form.

The SAP workflow is used to inform the request processors

For more information on the *SAP Workflow*, see *Basis* → *Business Management* → [SAP Business Workflow \[Ext.\]](#).

You can use Easy Cost Planning to calculate incurred costs for the request

For more information, see the SAP Library under *Financials* → *Controlling* → *Product Cost Controlling* → *Product Cost Planning* → [Easy Cost Planning and Execution Services \[Page 12\]](#).

Costs are automatically posted to the corresponding cost center or another *fixed account assignment object*.

Process Flow

The Environment

Intranet	SAP System
Entered by	Person who processes/approves the request

Entry

You can select predefined standard services, or create your own service request scenarios in Customizing.

You can add notes to the service request at any time during the processing time.

You enter an internal service request via the LaunchPad in your my SAP.com Workplace.

Approval

In Customizing, you can specify under which conditions an approval for a service request is to be made.

The cost center manager or budget manager automatically receives an overview of all the service requests relevant for them. This overview is provided via a worklist or a workflow.

The manager can:

- Change the person responsible, and thus forward the service request The request can be forwarded automatically, for example, to a representative if someone is on vacation.
- Before approval, send a note with a query on the service request to the person who entered the request.
- Forward the service request for a "cost estimate" to the processor.

The cost center manager or budget manager can use a cost collector (such as an internal order) that is linked to the internal service request to monitor the request costs.

Carrying Out The Request

The service request processor is determined in the same way as the cost center manager or budget manager.

This person is informed about the service request, using the workflow.

If the processor indicates the service request as completed, then the incurred costs are posted via integrated cost posting.

You call up processing

- For the notification via:
 - *Office* → *Notification* → *Change* or
 - *Office* → *Notification* → *Worklist* → *Notifications*.
- For the notification task via:
 - *Office* → *Task* → *Process* or
 - *Office* → *Notification* → *Worklist* → *Tasks*.

Result

You sent a request that was processed or executed after approval (if required).

For more information on customizing internal service requests, see the implementation guide (IMG) under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Scenario Definition* → *Define Scenarios*.

Preparing The Internal Service Request

Preparing The Internal Service Request

Use

To be able to use the [internal service request \[Ext.\]](#), you need to make some default settings in Customizing under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request Scenario Definition* → [Define Scenarios \[Ext.\]](#).
For more information, see also: [Definition Of A Scenario \[Page 9\]](#).

You can also create the form layout for each request yourself.

Prerequisites

Customizing for internal service requests includes the previous customizing settings.

For more information, see the implementation guide (IMG) under *Cross-Application Components* → *Internet / Intranet Services* → *Scenario Definition* → [Prepare General Notification. \[Ext.\]](#)

Features

Preparing internal service requests includes defining the characteristics for the general notification, and for the scenario.

Activities

You prepare the general notification in Customizing. As a minimum, you need to define notification types before you can set up scenarios. Use the notification types defined by SAP also.

Define a scenario.

Depending on the type of scenario, develop templates for calculation of the costs incurred by the execution of the request, or use Easy Cost Planning and Execution Services.

For more information, see the SAP Library under *Financials* → *Controlling* → *Product Cost Controlling* → [Easy Cost Planning and Execution Services \[Page 12\]](#).

Transport the template settings.

Definition Of A Scenario

Use

You need to define a scenario for each specific [internal service request \[Ext.\]](#) for which you want to later provide a form.

Prerequisites

As the internal service request is based on the general notification, you need to make the settings in Customizing for notifications. For more information, see the implementation guide (IMG) under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Scenario Definition* → [Prepare General Notification \[Ext.\]](#).

Features

In a scenario you specify the fields that are ready for input, which are visible in the request form, and you also specify their function.

Activities

To define scenarios, choose the following in Customizing: *Cross-Application Components* → *Internet- / Intranet Services* → *Internal Service Request* → *Definition of Scenarios* → [Define Scenarios \[Ext.\]](#).



Note that there are already predefined scenarios that you can copy and adapt to your requirements.

In the basic data, you need to make the following minimum entries: a name, an explanatory short and long text, and a name for the ITS service.

Create characteristics and name them. Categorize the characteristic if required.

Create tasks and name them. Enter the name of a person responsible for each task (directly or using a standard role).

Scenario

Scenario

Definition

Technical description of a particular [internal service request \[Ext.\]](#) in Customizing.

Use

A scenario specifies the appearance and the flow of an internal service request.

Structure

The following are some of the items stored in a scenario:

- A descriptive short and long text.
- For a cost-incurring scenario: Rules for calculating the incurred costs.
- The entry type for the request and the ITS service.
- The fields or characteristics that are to later appear on the request form.
- The tasks that are triggered by this request.

To define scenarios, choose the required default settings in Customizing under: *Cross-Application Components* → *Internet- / Intranet Services* → *Internal Service Request* → *Definition of Scenarios* → [Define Scenarios. \[Ext.\]](#)

Cost-Incurring Scenarios

If costs are incurred by the processing or execution of the [internal service request \[Ext.\]](#) that you are planning in the scenario, then you can mark the scenario as being cost-incurring. You can use the following to calculate the incurred costs:

- Enter a rough estimate
- Define a template, a costing variant and strategy
For more information, see the SAP Library under *Financials* → *Controlling* → *Product Cost Controlling* → *Product Cost Planning* → [Easy Cost Planning and Execution Services \[Page 12\]](#).

Activities

To define a cost-incurring scenario, choose the following in Customizing: *Cross-Application Components* → *Internet- / Intranet Services* → *Internal Service Request* → *Scenario Definition* → [Define Scenarios \[Ext.\]](#).

In the basic data of the scenario activate the *Cost incurring* indicator.

Easy Cost Planning and Execution Services

Easy Cost Planning and Execution Services

Purpose

Easy Cost Planning and Execution Services is a simplified form of cost planning that is integrated into the SAP System. It is of particular use when several planners are involved or when your planning is carried out using similar methods.

[Example of the Easy Cost Planning of a Project \[Ext.\]](#)

You can use this method for the following:

- [Internal orders \[Ext.\]](#)
- [WBS elements \[Ext.\]](#)
- [Internal Service Requests \[Ext.\]](#)
- [Appropriation requests \[Ext.\]](#)
- [Ad hoc cost estimates \[Ext.\]](#)

The Execution Services enable you to trigger the following processes based on the costing results:

- [Purchase requisitions \[Ext.\]](#)
- [Purchase orders \[Ext.\]](#)
- [Reservation \[Ext.\]](#)
- [Goods issue \[Ext.\]](#)
- [Internal activity allocation \[Ext.\]](#) either with or without [Workflow \[Ext.\]](#)

For more information about how Easy Cost Planning can be used, see the following:

[Easy Cost Planning for Internal Orders \[Ext.\]](#)

[Easy Cost Planning in the Project System \[Ext.\]](#)

[Cost-Incurring Scenarios \[Page 11\]](#)

[Easy Cost Planning for Appropriation Requests \[Ext.\]](#)

Integration

Cost planning for the object concerned is carried out in the corresponding application component, namely [Internal Orders \(CO-OM-OPA\) \[Ext.\]](#), [Project System \(PS\) \[Ext.\]](#), [Investment Management \(IM\) \[Ext.\]](#), *Product Cost Planning* (CO-PC-PCP) or using the mySAP Workplace.

The following components are used (called up) in the background. User knowledge in these areas is not required.

In order to complete the respective Execution Service, the application components *Purchasing* (MM-PUR), *Inventory Management* (MM-IM), *Cost Center Accounting* (CO-OM-CCA), and/or *Activity-Based Costing* (CO-OM-ABC) are used, either with or without the *SAP Business Workflow*.

Easy Cost Planning and Execution Services

Easy Cost Planning uses unit costing from *Product Cost Planning* (CO-PC-PCP), the characteristics function of the *Classification system* (CA-CL) and the [template \[Ext.\]](#) function from *Activity-Based Costing* (CO-OM-ABC) to calculate the costs.

Features

Easy Cost Planning enables you to represent the [structure of the planning processes in the system \[Ext.\]](#). These can then be used as [planning forms by all cost planners \[Page 15\]](#). The use of this planning form ensures that all of the relevant cost incurring functions are included. This means you can avoid errors.

The intentions of the planner are [translated into the terms and data of the unit costing \[Ext.\]](#). When planning, the planner enters the cost incurring functions in the form of characteristics, from which the system generates and then costs costing items for the unit costing.

The planning object can be structured whichever way the planner chooses, and the substructures that the user defines can be costed separately. Existing structures, such as those in the *Project System* for example, are included in the costing.

You can define the entry screen in accordance with your [individual \[Ext.\]](#) requirements for each planning form. It is automatically generated in HTML format and you can enhance it, for example with hyperlinks for the planner to determine additional data, and to include your company logo. You can also insert information in the form of explanatory text. A notes field has been created in the entry screen so that planners can enter notes either for themselves or other users who need to analyze the costing, which are saved together with the entries for the costing.

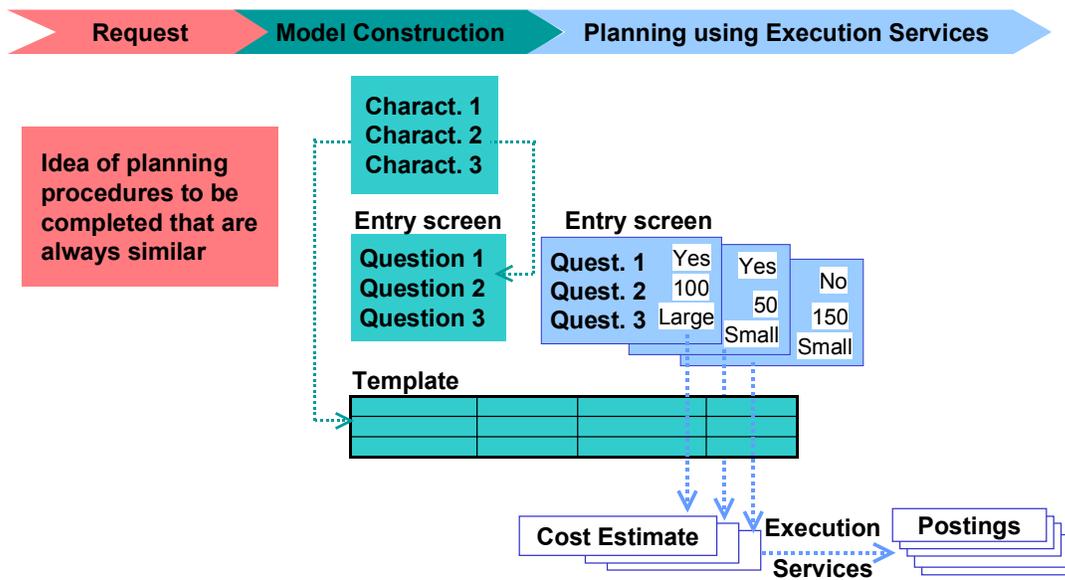
You can adapt the display of individual screens [for individual roles \[Ext.\]](#), for example to simplify the display for planners who only use this function occasionally and do not need to know about every single function.

The Execution Services function selects the costing items for the process to be triggered. You can change, delete and add the costing items selected. From this display it is possible to [trigger the process for the selected items \[Page 22\]](#).

It is possible for you to display existing postings, that are already in the system for the respective object, for the services selected. For example, if you want to trigger a purchase order for an internal order, you can display the existing postings for this order, in order to avoid a duplication of the purchase order.

Easy Cost Planning and Execution Services

Standardized Planning Method using Easy Cost Planning & Execution Services



Constraints

You cannot use this costing method for the following unit costing reference objects:

- Materials
- Sales documents
- CO production orders
- General cost objects
- Network activities

You cannot use the Execution Services for appropriation requests. You can only use them for ad hoc cost estimates if you have assigned an [account assignment object \[Ext.\]](#) to the ad hoc cost estimate.

Using Easy Cost Planning

Use

Easy Cost Planning is used to determine the costs for a particular object (for example, internal order, WBS element, internal service requests, appropriation request) or generally to determine costs for a specific purpose using an ad hoc cost estimate.

Integration

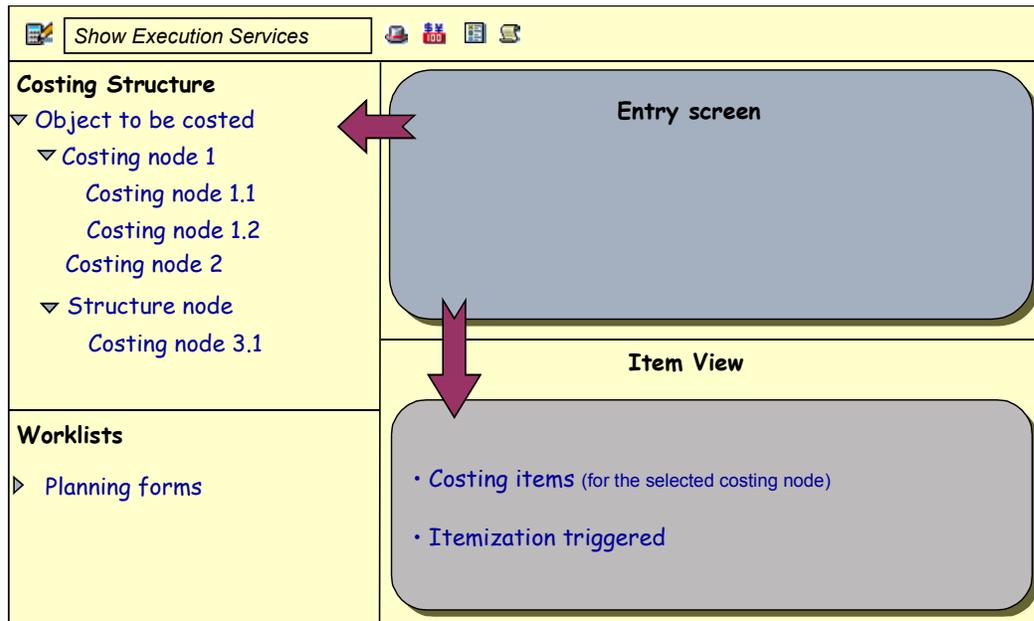
To activate the cost planning, the corresponding application component is used, such as *Internal Orders* (CO-OM-OPA), *Project System* (PS), *SAP Internet Applications*, *Investment Management* (IM) or *Product Cost Planning* (CO-PC-PCP).

Prerequisites

If you want to use a planning from for the costing, you must have already defined this in the system. For more information, see [Processing Costing Models and Assignment of Attributes \[Ext.\]](#).

Features

The costing screen is constructed as follows:



Depending on the settings for the [Role \[Ext.\]](#) to which you have been assigned, you may find that you are not able to use all of the functions set out below.

You are able to do the following:

- Change the size of the screen areas

Using Easy Cost Planning

- Display or hide the *costing structure*, *worklists* and *item view* as appropriate.

Screen area *Costing structure*

The top node is the object to be planned or the ad hoc cost estimate. You cannot cost structure nodes, meaning you cannot assign them a planning form. For project structures these are WBS elements which are not planning elements.

You can structure the object you wish to plan or your ad hoc cost estimate by adding new nodes (costing nodes). Select the node that you want to substructure and choose . Enter a description and choose . The node is added. You calculate the costs for each costing node separately.

To delete a costing node, select it and choose . Technically a costing is deleted this way.

You can use  to close the cost estimate, to prevent further changes being made to it. The close is recorded in the history. You can check this in the header data. There you can undo or set the closing of a cost estimate. If you want to undo it, choose  again.

The [document flow \[Ext.\]](#)  displays for each costing item whether [documents have been posted through an Execution Service \[Page 22\]](#) already for this costing item and object to be planned.

Screen area *Worklist*

If you regularly create cost estimates using Easy Cost Planning, you can load the planning forms that you use frequently into your worklist. To do this either choose  immediately next to the node or choose .

You can create, insert , rename worklists and extend the display of worklists (from user-specific to role-specific or globally to all users), or restrict  them. Using  you can call up any worklists that are in the system but have not yet been displayed. To delete worklists, choose . To remove subdirectories or planning forms from a worklist choose . To save your worklists, choose  in the screen area *Worklists*.

Screen area *Item view*

This area is used to enter, change and display costing items. For more information, see [Creating Costing Items \[Ext.\]](#) under the heading *Features*. It is not possible to make entries in the detail screen.

The item category for planning forms is *J*. For more information about item categories, see [Master Data for Unit Costing \[Ext.\]](#). In Easy Cost Planning you cannot use item categories *O* and *S*.



You should note that the *item number* is not necessarily the same as the item of the line in the item view. If you need this, define a layout that sorts using the item number.

For WBS elements and internal orders you can define a period for a costing item by entering a *latest end date*. The costs are distributed evenly across the periods for the period you defined.

Further Item View Functions

Using Easy Cost Planning

	The log of the costing node currently selected is displayed.
	Reevaluate [Ext.] the costing items with the current prices.
	Refresh the totals and subtotals
	You can overwrite the price, cost element or the description for an item with the current plan and master data. To do this select the line.
<i>Goto → Triggered itemization on/off</i>	<p>If you have costed several costing nodes in one structure, all of the costing items for all costing nodes in this structure are displayed in the itemization triggered. The top node in the costing structure must be selected for this. The display can be seen in the item view.</p> <p>By choosing  you can multiply the quantities of individual costing items by a factor. The change is accepted immediately and the costing repeated.</p>

For more information on working with this kind of list display see [SAP List Viewer \[Ext.\]](#).

To cost the entries changes, choose *Confirm*.

Functions in the Upper Application Toolbar

<i>Show Execution Services</i>	After cost planning you can trigger Execution Services [Page 22] for some objects.
	Display header data
	You can switch between various currencies.
	Only displayed for ad hoc cost estimates [Page 20]
	Explains all of the symbol legends in the <i>costing structure</i>
	The error messages are all displayed for every costing node.

Activities

- Create either the object you want to plan or an ad hoc cost estimate, or else call up an existing one in change mode. Choose the cost planning. For more information, see the documentation for the object to be planned:
 - For internal orders: [Easy Cost Planning for Internal Orders \[Ext.\]](#)
 - For WBS elements: [Using Easy Cost Planning \[Ext.\]](#)

Using Easy Cost Planning

- For appropriation requests: [editing cost estimates for appropriation requests \[Ext.\]](#)
- For ad hoc cost estimates: [executing ad hoc cost estimates \[Page 20\]](#)
- You have the following options for calculating the planned costs:
 - With Planning Forms**
 - Click on the node to be costed in the *costing structure*. Choose *Choose planning form* in the right hand screen area and select a planning form (*costing model*) or click on the corresponding node for the planning form in the worklist if you have already loaded the required planning form. The system displays the entry screen.
 - If you have selected the wrong planning form you can assign a new one. To do so you have to select the costing node. You now have the following options:
 - From the worklist choose a new planning form by clicking on the corresponding node, and confirm the prompt with *Yes*.
 - Choose  and confirm the prompt with *Yes*, choose *Choose planning form* in the right hand screen area and select a new planning form.
 - Enter the required planning data and choose *Confirm*.
 - You can change the costing items in the *item view*.
 - Without Planning Forms**
 - For this showing the *Item view* must not be excluded by your [Role \[Ext.\]](#) .
 - Click on the node to be costed in the *costing structure*. Enter the costing items directly in the *item view*.

Result

The results are displayed directly in the *costing structure* for every costing node. The costs are displayed subdivided according to costing items. In essence this display corresponds both in use and structure to the [valuated BOMs \[Ext.\]](#) of the unit costing. The symbols next to the costing items specify the item category. By choosing  for the legends you can ascertain which item category you are dealing with.

The costing items are displayed in a non-hierarchical list in the *item view*. This essentially corresponds to the display of the itemization. This means the respective costing items for the currently selected costing node are displayed. The SAP standard delivery makes a variety of itemization layouts available to you. You can adapt these to meet your individual requirements by [creating your own layouts \[Ext.\]](#). Using various factors you can [calculate and delete subtotals \[Ext.\]](#).



If currencies are translated, the system uses the exchange rate applicable on the valuation date of the cost estimate (not the costing item, if this is different).



If you leave Easy Cost Planning via , your cost estimate is temporarily saved. The system saves to the database only when you save the object to be planned.

Using Easy Cost Planning

Exception: Ad hoc cost estimates In this case you are asked whether the data is to be saved directly upon leaving Easy Cost Planning. If you choose *No*, the data is not saved and is lost.

You can only save or leave the cost estimate if it does not contain any errors.

Therefore, you can only leave a cost estimate which does contain errors by choosing



For internal orders, WBS elements and ad hoc cost estimates you can now [trigger particular Execution Services \[Page 22\]](#).

See also:

[Easy Cost Planning and Execution Services \[Page 12\]](#)

Executing Ad Hoc Cost Estimates

Executing Ad Hoc Cost Estimates

Use

If you want to plan costs without creating an object in the system, you can use ad hoc costing to obtain a speedy result.

Prerequisites

If you want to use a planning form for the costing, you must have already defined this in the system. For more information, see [Processing Costing Models and Assignment of Attributes \[Ext.\]](#).

Procedure

Choose *Accounting* → *Controlling* → *Product Cost Controlling* → *Product Cost Planning* → *Easy Cost Planning & Execution Services* → *Edit Ad Hoc Cost Estimate*.

Creating Ad Hoc Cost Estimates

1. Enter a description for your planning projects and choose *Create*.
2. On the tab page *Additional data* enter a *company code* or a *plant*.
3. To start an ad hoc cost estimate with a planning form directly, you have the following options on the tab page *Create with planning form*:
 - Click on any planning form displayed. The cost estimate is started immediately.



If no planning form is displayed for instance because you have not yet created an ad hoc cost estimate, then choose either *All planning forms* under *Subject area* or some other role-specific workload.

- Under *With planning form*, enter the (technical) name of a planning form direct, and choose *Create*.
 - To find a planning form enter a search term and choose *Find*. The search is made in the names and in the descriptions available in your logon language for the planning forms. You can also display the descriptions via  next to the planning forms.
4. For more information, see [Using Easy Cost Planning \[Page 15\]](#).

To forward a cost estimate to another user for further processing or analysis, choose the service *Send* → *Send object with note* from the object services in the title bar of the symbol. . You can find more information on object services under *Services for object*  → *Help for the object services*. The user to whom the cost estimate has been sent accesses the display mode of the cost estimate on opening the attachment to the note.

Changing Ad Hoc Cost Estimates

You have the following options:

- Under *Description* enter the name of the ad hoc cost estimate which you want to change and choose *Change*.

Executing Ad Hoc Cost Estimates

- Choose the tab page *Planning forms last used* and click on the ad hoc cost estimate that you want to change. You go directly to change mode.

For more information, see [Using Easy Cost Planning \[Page 15\]](#).

Triggering the Execution Service

To be able to trigger an Execution Service you have to assign an [account assignment object \[Ext.\]](#). This is used to post all of the costs incurred through triggering the Execution Service (for instance through a purchase order).

1. You are in the create or change mode of the cost estimate. Choose .
2. Choose *Object type*.
3. Enter the key of the account assignment object that is to include the costs.
4. For more information, see [Triggering the Execution Services \[Page 22\]](#).

Triggering the Execution Services

Triggering the Execution Services

Use

Having planned costs [using Easy Cost Planning \[Page 15\]](#), you want to trigger one of the following processes: [Purchase requestion \[Ext.\]](#), [purchase order \[Ext.\]](#), [reservation \[Ext.\]](#), [goods issue \[Ext.\]](#), or [internal activity allocation \[Ext.\]](#) with and without [Workflow \[Ext.\]](#). To do this, the system will utilize the data that you entered for planning the costs.

This is possible for:

- Internal orders
- WBS elements
- Ad hoc cost estimates to which you have assigned an account assignment object.

Integration

The application components triggered for the Execution Service are the application components *Purchasing* (MM-PUR), *Inventory Management* (MM-IM), *Cost Center Accounting* (CO-OM-CCA), *Activity-Based Costing* (CO-OM-ABC) and [SAP Business Workflow \[Ext.\]](#).

Prerequisites

The required settings have been made in Customizing under *Execution Services*.

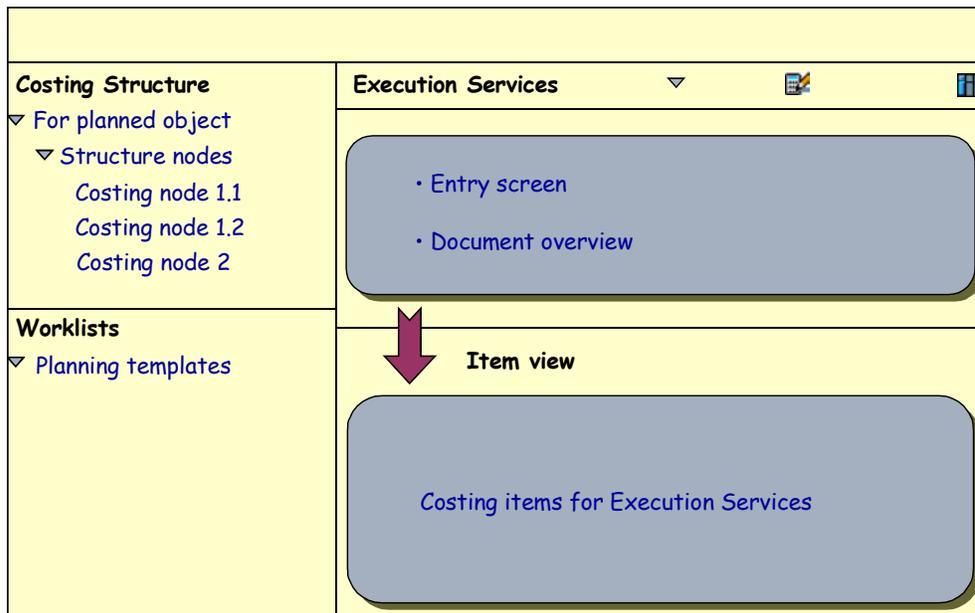
You have executed the cost planning, and the object (internal order or WBS element) exists in the system.

Showing the *Item view* must not be excluded by your [Role \[Ext.\]](#).

Features

The costing screen is constructed as follows:

Triggering the Execution Services



Under *Execution Services* you are offered all of the Execution Services that are possible for the object to be planned. Before you select an Execution Service, it is possible to obtain an overview of all the existing purchase requests, purchase orders and so on for the costing items that have been created through the Execution Service. Click on the costing item in the screen area *Costing structure*, and choose .

The items relevant to the Execution Service are selected and, where applicable, expanded to include additional data. By using  you can also display the *Document overview* for the Execution Service and the object in question, enabling you to avoid a double posting.

The Execution Service is posted in the system. Choose  in the *document overview* to display the document posted.

Activities

1. Choose *Show Execution Services* and select an Execution Service. The system automatically selects those items relevant to the Service you selected, for instance for internal activity allocations E (*internal activity*), P (Process - manual), X (Process costs ABC) and V (*variable*) items are selected.
2. Select the costing items for which the Execution Service is to be triggered. You can also change the items, for example by entering a different quantity.
3. Post the items.

If an employee of the cost center that supplied the internal activity is also to post the internal activity allocation, you can start a **Workflow**. Select *Internal activity allocation* in order to select the relevant costing items. Select the item(s) and choose .

The screen *Change Document: Cost Center XYZ: Create Note* appears. In this screen, you can create messages for the workflow processor. Save (even if you have not created a message) and go back to the initial screen.

Result:

Triggering the Execution Services

The items are in the processor's [workflow inbox \[Ext.\]](#). When the processor calls the task, the screen *Easy Cost Planning: Execution Services* appears. The processor can then post the internal activity allocation in this screen. By confirming the processing when leaving the screen, the task is deleted from the processor's inbox.



You cannot start a workflow for business process items.

See also:

[SAP List Viewer \[Ext.\]](#)

Template for Easy Cost Planning

Definition

The template is a dynamic calculation tool that uses [functions \[Ext.\]](#) and [formulas \[Ext.\]](#) to calculate numerical values and determine the results of Boolean expressions (true or false). Templates for Easy Cost Planning are created in environments 200, 205-208, and 214-215. The environment is defined automatically by the system depending on the object to be planned (internal order, WBS element, and so on). With the exception of environment 214, you cannot create environments through Customizing. Instead, you can only do so by defining a [costing model \[Ext.\]](#).

Structure

The template contains a table in which you can make the following line entries:

Column	Possible Entry/Process
<p><i>Item category in template</i></p>  <p>The key of this item category is not always the same as that of the costing item.</p>	<p>Comment line, process, calculation row (process), cost center/activity type, calculation row (cost center/activity type), costing model, external activities, subcontracting, material, service, base planning object, text item, variable item</p>  <p>For the item category <i>calculation row</i>, you cannot call the editor or define methods in the object column.</p> <p>You can only select costing models that are valid either for all planning objects or for the same planning object as the costing model in which you want to insert this model as a submodel.</p>
<p><i>Description</i></p>	<p>The description is displayed in the cost estimate as the description for the costing items. The system determines the description for most of the item categories automatically once you have confirmed your entries. (The <i>description</i> is used in the case of the base planning object.) Consequently you only need to make an entry here if the description cannot be determined by the system or if you need to overwrite it.</p>

Template for Easy Cost Planning

<i>Object</i>	<p>Enter the object (such as a material), depending on the item category specified. You can either enter the object directly, or define methods [Ext.].</p> <p>For <i>item categories of the template J to R</i>, you can use the input help to make the required entries. Always enter a cost element if the system is unable to determine one. This applies to those item categories for which the cost element was requested in the possible entries help; with base planning objects the cost element can be determined via the master data providing you have entered one there. If the item does not have a cost element, it is not possible to assign the costs to a cost component. This means that the costs cannot be rolled up if they are incurred in a cost estimate which is part of a costing structure with superior cost estimates.</p> <p>For the categories <i>Process</i> and <i>Cost center/Activity type</i>, you can either predefine an object or determine dynamically one or more processes or cost centers/activity types. For more information, see Object Determination [Ext.].</p>
<i>Quantity</i>	<p>Enter a quantity or characteristic directly. You can enter the appropriate characteristic by defining a formula [Ext.].</p> <p>For more information, see Activity Quantity Determination [Ext.].</p>
<i>Activation</i>	<p>Specify the condition under which an item is active. For the <i>activation</i> of an item, you can predefine values as <i>active</i> or <i>inactive</i> or define a method that returns <i>active</i> or <i>inactive</i> at the point of evaluation. If you do not enter anything in the column, the item is active.</p> <p></p> <p>The item (MATERIAL; Gravel; 0001 / GRAVEL01; M3; QUANTITY_FILLER;  METHOD) with method: FILLER = "Gravel" is only active if gravel was selected as the FILLER. For all other characteristic values, the item is not included in the costing.</p> <p>For more information, see Activation [Ext.].</p>
<i>Price</i>	<p>Price for a variable item This column is not utilized for any other item category.</p>

Depending on the item category in the template, there are various functions available to define methods and formulas.

For more information on the costing item categories, refer to [Master Data for Unit Costing \[Ext.\]](#).

Creating Templates

Use

Using the template to create an [internal service request \[Ext.\]](#) is only useful if it is for a cost-incurring request, such as a removal. In a template, you store rules for calculation of the costs that are incurred during request processing.

Prerequisites

You have indicated the scenario as *cost-incurring*.

Tasks

You store the use of a template, and its definition in Customizing under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Definition of Scenarios* → [Define Scenarios \[Ext.\]](#).

After you have defined a template, it is not automatically transported with the other details that you stored in the scenario. To transport the template settings, go to Customizing, and choose *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Scenario Definition* → *Settings for Cost Incurring Scenarios* → [Transport Template Settings \[Ext.\]](#).



Note that you can use **only** the kind of templates for costing scenarios that you defined in the scenario Customizing (see above).

You have provided your cost-incurring scenario with a template. This template calculates the incurred costs using the different fields filled in the request.

For more information on templates, see the SAP *Library* under *Financials* → *Controlling* → *Overhead Cost Controlling* → *Activity-Based Costing* → [Template \[Ext.\]](#).

Notification

Notification

Definition

Data record that a user can use to inform an application area about a particular subject matter.

Use

You can use the notifications in the SAP System and on the Intranet to record and process various business-related information. The notification supports you with a number of functions that you can use to process this information. The most important of these functions allow you to:

- Record problems or information
- Define and execute [tasks \[Page 30\]](#)
- Log [activities \[Page 38\]](#)
- Record costs
- Manage the status of notifications and tasks
- Communicate with persons or departments affected by the subject matter
- Evaluate notification data

Structure

In the SAP System, you can process notifications in a simplified or extended view. The structure of a notification differs on the basis of this processing view:

- [Notification structure \[Ext.\]](#) for simplified notification processing
- [Notification structure \[Ext.\]](#) for extended notification processing

If you have the appropriate authorization, you can switch between the simplified and extended processing view.

Integration

The SAP System currently supports the following notification categories:

- Quality notification
- Maintenance notification
- Service notification
- Claim
- Feedback notification (R/3 Support Line Feedback)
- Internal service request
- Change notification
- General notification

Notification Type

Definition

Key that specifies the origin, informational content, and other attributes of a notification. In the standard system, a notification type is assigned various notification origins, which in turn are grouped together in a notification category.

The standard system supports the following notifications (examples):

Notification Type	Notification Origin	Notification Category
M1 - Maintenance request	01 - Maintenance request	Maintenance notification
M2 - Malfunction report	02 - Malfunction report	
M3 - Activity report	03 - Activity report	
S1 - Problem notification	04 - Problem notification	Service notification
S2 - Service request	05 - Activity report (Service)	
S3 - Activity report	06 - General notification (Service)	
Q1 - Customer complaint	Q1 - Customer complaint	Quality notification
Q2 - Complaint against vendor	Q2 - Complaint against vendor	
Q3 - Internal problem notification	Q3 - Internal problem notification	
\$\$ - Feedback notification	R3 - Feedback notification	
C1 - Claim (internal)	C1 - Internal claim	Claim
C2 - Claim (external)	C2 - External claim	
G0 - General notification	G0 - General notification	General notification

In addition to the notification types listed above, the standard system may also contain other notification types.

Use

When you create a notification in the system, you choose a notification type that suits the subject matter of the notification from a business standpoint. You can either use the notification type that was predefined in Customizing or in the user default values (default notification type), or you can select another notification type.

You can also define your own notification types in Customizing. When you create a new notification type, you must assign a notification origin predefined in Customizing to this notification type. On the basis of this assignment, the system automatically determines the notification category.

You cannot define notification origins or notification categories in Customizing.

Task

Task

Definition

Planned or completed activity that describes a planning and organizational measure in a notification. A task can:

- Apply to the notification as a whole or to individual items in a notification
- Have different statuses

Use

Using a task, you can plan how different people can interact to process a notification and track how various activities are completed within a specified period of time. The task data in a notification is displayed in varying levels of detail in the:

- Task overview
- Task detail screen

You can record task information for the notification header and for individual notification items:

- Tasks that apply to the notification as a whole (or notification header) are designated as "immediate tasks." These tasks usually provide a "quick fix" for a problem that requires immediate attention.
- Tasks for notification items are referred to as "corrective tasks." These tasks usually provide permanent solutions that will prevent the problem from reoccurring in the future.

For each task, you can enter the:

- Key for the task to be performed and a brief instruction describing what is to be done
- Planned start and end dates for the task
- Task status

Integration

If the appropriate data is maintained in Customizing, the system can trigger certain follow-up actions on the basis of the task code (for example, printing a paper or calling up an SAP transaction). You can define follow-up actions to meet your individual requirements. (For more information, see [Follow-up Actions for Tasks \[Ext.\]](#).)

Entering Tasks

Procedure

To use the functions in the following table, call up the notification in the create or change mode.

Function	Tab page / Pushbutton / Menu path	What you should know
Record immediate tasks for the notification header	Choose the <i>Tasks</i> tab page.	Enter the data on the task overview.
Record immediate tasks on task detail screen	Select a task on the task overview and choose <i>Detail view</i> .	Enter the data on the task detail screen.
Record corrective tasks for a notification item	Choose the <i>Items</i> tab, select an item, and then choose the <i>Item tasks</i> tab page.	Enter the task data for the item.
Record corrective tasks on the task detail screen	Choose the <i>Items</i> tab page and then select an item. Choose the <i>Item tasks</i> tab page, select a task and then choose <i>Detail view</i> .	Enter the planned start and end dates for the task on this screen. The dates you enter may be changed based on the status you assign to the task (see Status Management for Tasks [Page 123]).

See also:

[Automatic Determination of Tasks \[Ext.\]](#)

Follow-Up Actions for Tasks

Follow-Up Actions for Tasks

Use

You can define a follow-up action for a task that will automatically be triggered and executed by the system. A follow-up action can call one or more function modules that execute specific functions (for example, printing a shop paper or calling up a transaction).

Prerequisites

In Customizing for *Cross-Application Components* under:

- *Notification* → *Notification Creation* → *Notification Content* → *Maintain Catalogs*, the indicator for follow-up actions is set in the task catalog (catalog type 2)
- *Notification* → *Notification Processing* → *Additional Notification Functions* → *Define Follow-Up Actions for Tasks*, you have defined a follow-up action key and assigned one or more function modules to be called to this key

A follow-up function is only executed if the business transaction that is assigned to the follow-up action allows it.

Features

The system automatically executes a follow-up action after you have:

1. Entered a task code (for the notification header or for a notification item) that has a follow-up action assigned to it.
2. Saved the notification

You can display a log on the task detail screen that lists all follow-up actions that were executed.

Action Box

Use

When you process notifications, you can use the **action box** to execute **follow-up functions**. A follow-up function is a function that:

- Can either be executed as part of an overall business process or independently of such a process
- You can select and execute in the action box by means of a mouse click while you are processing a notification or task

Prerequisites

The follow-up functions in the action box are defined in Customizing for *Cross-Application Components* under *Notification* → *Notification processing* → *Additional Notification Functions*. The follow-up functions that:

- Have been predefined in the standard system can be used without any further preparations
- Are displayed in the action box, but which you do not need, can be deleted in Customizing
- You have programmed yourself and are to be included in the action box must also be defined in Customizing

Features

The following table contains the most important information you need to know about setting up and using the action box.

Function	What you should know
Appearance of the action box	<p>In the standard system, the action box is displayed as an overview tree. In this display mode, the follow-up functions are displayed in a list in different colors. The colors have the following meaning:</p> <ul style="list-style-type: none"> • Blue: The follow-up function is active and can be executed. • Black: The follow-up function is inactive and cannot be executed (yet). • Gray: The follow-up function has been executed and cannot be executed again. <p>If you set the <i>Action box: Table</i> indicator in the user default values, the system displays the action box as a table. In this mode, the system only displays the follow-up functions that can be executed (without the color coding).</p>

Action Box

Follow-up functions and dependent follow-up functions	<p>Depending on your settings in Customizing, the action box can include follow-up functions that you:</p> <ul style="list-style-type: none"> • Can execute more than once and independently of other follow-up functions in the action box • Can execute only once and/or after certain other follow-up functions have been executed (for example, follow-up functions within a process)
Control parameters for a follow-up function	<p>In Customizing, you can specify:</p> <ul style="list-style-type: none"> • In which processing view a follow-up function will be displayed for use (for example, in the simplified or extended notification processing modes) • In which transactions a follow-up function will be displayed for use (for example, in the create or change mode) • Whether a follow-up function will be displayed for use in the transactions for notification processing and/or task processing • Whether a follow-up function will be documented as a task or activity in the notification, or whether it will not be documented at all That all follow-up functions documented as tasks or activities will be available for use in the action box when you create or change a notification The follow-up functions that are not documented will also be available for use in the action box when you display a notification
Assigning follow-up functions to scenarios	<p>In addition to assigning follow-up functions to a notification type, you can also assign them to a scenario. For example, if you are processing a notification that has a scenario assigned to it, the action box will contain follow-up functions that are assigned to the notification type, as well as those that are assigned to the scenario.</p>
Standard follow-up functions	<p>The action box contains several follow-up functions that have been predefined in Customizing. For more information about these functions, see Standard Follow-Up Functions [Page 36].</p>
Business transactions	<p>In Customizing, you can assign a business transaction to a follow-up function. A business transaction determines whether a follow-up function can be executed, based on the current status of a notification.</p>

Action Box

<p>Authorization check</p>	<p>Each follow-up function that is documented as a task or activity is assigned a code group in Customizing. When you process a notification, the system automatically checks whether you are authorized to access the code groups assigned to the various follow-up functions. If you do not have authorization for a particular code group, the corresponding follow-up function will not appear in the action box. This means that you can only use the follow-up functions for which you have authorization.</p>
<p>Workflow link</p>	<p>You can link follow-up functions that are documented as tasks to the functions of the <i>SAP Business Workflow</i> component. The system can then trigger workflow tasks once a follow-up function (task) has been executed and the notification has been saved.</p>
<p>Difference between follow-up functions and follow-up actions</p>	<p>Follow-up functions in the action box are different from task-related follow-up actions [Page 32]. Both follow-up functions and follow-up actions call function modules. With a follow-up action, however, the system executes the corresponding function module when you save a notification. With a follow-up function, the system executes the function module as soon as you execute the follow-up function in the action box. Also, when you process follow-up functions, you enter information interactively in a dialog box. This is not possible with follow-up actions.</p>

Activities

To execute a follow-up function in the action box, you click the text or double-click the symbol for a follow-up function (displayed in blue). A dialog box appears, in which you can enter the data that is required to execute the function.



You can only execute a follow-up function if you select it from the action box. If you create an activity or task in the notification using the input help for a catalog, the system does not execute a follow-up function.

The system then executes the selected function. Depending on the settings in Customizing, the follow-up function is:

- Documented as an activity in the notification
- Documented as a task for the notification header
- Not documented

Standard Follow-Up Functions

Standard Follow-Up Functions

Use

The action box in the standard system contains several follow-up functions that can be used by all applications to make it easier to process notifications. When you execute one of these follow-up functions, the system documents the execution of this function as an activity or task for the notification header.

Features

Follow-up function	What you should know
Telephone contact using <i>SAPphone</i>	<p>You can use this function to make a telephone call using <i>SAPphone</i>. The system records the details of your phone call as a notification activity.</p> <p>To use this function, the telephone and computer in your office must be interconnected and the <i>SAPphone</i> component must be active.</p>
Telephone contact without <i>SAPphone</i>	<p>You can use this function to record a telephone call that you initiated while processing a notification. The system records the details of your phone call as a notification activity.</p>
Internal memo	<p>You can use this function to record an internal message or remark that relates to the subject matter of the notification. The system records the memo as a long text for an activity.</p>
Solution database [Ext.]	<p>You can use this function to search the solution database for symptoms and solutions that are similar to the problem you are currently processing.</p> <p>If you find a symptom or solution in the database that applies to the problem in your current notification, you can:</p> <ul style="list-style-type: none"> • Document your database search as an activity for the notification • Copy the tasks associated with the solution into your notification

Standard Follow-Up Functions

<p>Send notices</p>	<p>Using this function, you can postpone the processing of a notification. The system can then print this data or send it as a fax or e-mail to an internal or external address (Internet).</p> <p>The system copies the long text associated with the corresponding code in the task catalog as a text proposal. You can change this text as necessary without changing the original long text. If the notification contains several partners, you can select a partner before you process the text.</p> <p>You can view the document in a print preview before you save the notification. When you save the notification, the system sends the notice by executing a follow-up action.</p>
<p>Send e-mail</p>	<p>You can use this follow-up function to send an e-mail (with or without an attachment) to a partner in the notification.</p> <p>The system copies the long text associated with the corresponding code in the task catalog as a text proposal.</p>

Activity

Activity

Definition

Completed action or information recorded during the processing of a notification.

Use

You can enter the following information for each activity:

- A key for the action performed and a short text that can be changed
- Start and end of the activity
- Quantity factor (for example, if an activity consisted of sending three units of a material to a laboratory for testing, you can enter the value "3" for the quantity factor)

An activity can apply to the notification header as well as to individual notification items. The activity data is displayed in various degrees of detail on two screens in the notification:

- On the activity overview
- On the activity detail screen

The activity overview and detail screens are the same for all notification types.



The main difference between an activity and a task is that an activity does not have a status and cannot be assigned a partner. The activity is used strictly for documentation purposes.

Entering Activities

To use the functions in the following table, call up the notification in the create or change mode.

Function	Tab page / Pushbutton / Menu path	What you should know
Record activities for the notification header	Choose the <i>Activities</i> tab page.	Enter the data in the activity overview.
Record activities for the notification header in the activities detail screen	Select an activity in the activity overview and choose <i>Detail view</i> .	Enter the data in the activity detail screen. Enter the date and time when the activity was performed. To create a new activity from the detail screen, choose <i>New entry</i> .
Record activities for a single item	Choose the <i>Items</i> tab page, select an item, and then choose the <i>Item activities</i> tab page.	Enter the activity data for the item.

Defining Scenarios

Defining Scenarios

Use

You standardize the application and execution of a service request when you define its scenario.

Prerequisites

You need to create notification types first. In the notification type, you have specified whether the service request requires approval, and if so, who is to approve it. Note that you can use a notification type for more than one scenario.

Check whether all notification types from client 000 are in client 003. If you need notification types that are missing, import them using the **QISR_SM29** transaction from client 000. If these are not sufficient, then you can create your own notification types. The same procedure applies to catalogs.

For more information, see Customizing under *Cross-Application Components* → *Internet / Intranet Services* → *Scenario Definition* → *Prepare General Notification* → *Notification Creation* → [Notification Type \[Ext.\]](#).

Procedure



All of the scenarios provided by SAP are in the **S**.namespace.

1. In Customizing, choose *Cross-Application Components* → *Internet-/Intranet Services* → *Internal Service request* → *Scenario Definition* → [Define Scenarios \[Ext.\]](#).
2. Choose *New entries*.
3. In the *general data*, enter the data required.
4. Use this opportunity to store a long text for the scenario. The system displays this description as an explanation when you call up the request form. You can also use this description to find a suitable request form in the intranet. You do not need to restrict your text to key words when you enter it. The search engine does not just filter the text using the words, but also includes their grammar, and uses word families for the search.
For more information on the search machine, see the SAP Library under *Financials* → *Controlling* → *Cost Center Accounting* → *Information System* → *Interactive Information System*, then in *Report Documentation*, search under the *Document Search Using The Retrieval System*.
5. In the details for the form, decide how the internal request can be entered. You can choose between:
 - Text entry alone
 - Entry with form
 - Entry with notification transaction.

If you chose *Entry with form*, then the following procedures apply.

You only need to specify an ITS service if you selected the *Entry with form* entry type. You use an HTML form especially designed for the request scenario to enter the request.

Defining Scenarios

To do this, you need to assign an internet service to the scenario. This internet service must contain the HTML templates and language-dependent resources that are required for the request. There is also a choice of business add-ins (BADIs) available, in which you can define specific requirements.

You can assign an Internet service to maximum one scenario.

6. To create a new Internet service, choose *Generate*.

You have two options:

- **Generate Internet Service With Reference**
If you specify a scenario with an existing Internet service as a reference when you generate the Internet service, then this is copied to the new Internet service name.
- **Generate New Internet Service (SAPDesign)**
If you do **not** specify a scenario with an existing Internet service as a reference when you generate the Internet service, then the system copies the standard Internet service *SR00*. An Internet service generated in this way contains all the settings required for immediate testing of the scenario definition. The system automatically inserts a *Request data* area (with the corresponding entry fields) for the request-specific characteristics defined in the scenario.
- **Generate New Internet Service (Plain HTML)**
The same conditions apply as for generation using SAPDesign.



This type of Internet service creation is only possible using Customizing for scenarios for internal service requests.

7. You need to manually publish the Internet service before you can test it. After generation, the processing screen for the Internet service in the Development Workbench automatically appears, where you can publish the entire Internet service. Next you can test the ITS service when in Customizing, by choosing *Test*.
8. To add processing logic to forms, choose *Business Add-In*. If you need special initializations or checks for the request, you can use a business add-in (BADI *QISR1*). For more information on BADIs, see the *SAP Library*, under *Basis* → *Change and Transport System - Overview* → *Transactions and Tools* → *BC Changes to SAP Standards* → [Business-Add-In \[Page 44\]](#). As a standard measure, each scenario has its own BADI implementation (from the *QISR1* BADI definition) assigned to it. However, you can use a BADI implementation for more than one scenario if required.
- Assigning a BADI implementation to more than one scenario
 - Referencing a scenario to the BADI of another scenario
This is particularly recommended, as you do not need to modify any existing BADI implementation.
9. To change the action box that appears in request processing, choose *Activity box*. All follow-up activities that were defined for the notification type appear. You can now add follow-up activities that are scenario-specific, or delete ones already defaulted for this scenario. For more information on the action box, see [Action Box \[Page 33\]](#).
10. Decide whether executing this scenario incurs costs. If you activate the *Cost incurring* indicator, you can either specify *estimated costs*, or store a

Defining Scenarios

template with *costing variant* and *strategy*, which determines values for the cost allocation using the details in the form.

For more information on the template, see [Create Template \[Page 27\]](#).

11. If you have not yet prepared a template, choose *Create template*.
If you decide to use the costing with template, you can use the Easy Cost Planning functions.
For more information, see the SAP Library under *Financials* → *Controlling* → *Product Cost Controlling* → [Processing Costing Variants And Assigning Attributes \[Ext.\]](#).

12. Choose *Characteristics* from the selection area.

13. Now enter the fields that you want to have in the form for the scenario.
You can choose between:

- Uncategorized characteristics
Characteristic with up to 250 characters
- Dictionary Type Characteristics
Characteristic from the dictionary

For data elements, the system transfers the category and description of the characteristic from the dictionary. For structures, the system takes all the data elements from the structure.

The characteristic category is automatically included in the form.



The following restrictions apply to the definition of scenario characteristics:

- Characteristic names or structure names must be unique within a scenario.
- Characteristic names or structure names may not begin with ISR_
- Characteristic names or structure names should not be named as HTML business indicators, for example LABEL, NAME, VALUE, DIM, MAXSIZE, TYPE, EXISTS, ENABLED or VISSIZE.
- The length of the characteristic names or structure names should have a maximum of 28 characters.
- The data length of the characteristic or structure fields is restricted to 250 characters.

14. Choose *Tasks* from the selection area.
Use the workflow or worklists in the SAP System to specify the processing of the internal service request.
For more information on tasks, see: [Tasks \[Page 30\]](#).
15. Give each task in the processing a description.
16. If you want to evaluate tasks, store a key.
Follow-up activities are automatically filled if you choose a key. A follow-up activity is an action that is automatically executed when the task is processed.
17. You can store a subtemplate for a cost-incurring scenario
For several tasks, a subtemplate enables you to group and assign the rules for the template to the task. You can assign items correctly using the subtemplate during Easy Execution.
18. To find the correct processor for the task, you can store a standard role. You can define some standard roles in the transaction **PFAC**.

In the container, you store characteristic names or field names from one of the following tables:

- TQ80
- VIQMEL
- VIQMSM

The stored fields enable the correct processor to be found, using the standard role. If one of the fields stored in the container is filled in the form, then the correct processor is found during runtime.

If you want your user-defined fields (not in the tables named above) to be used for standard role determination, use the business add-in **QISR3**.

For more information on roles, see the SAP Library under *Basis Components* → *Business Management* → *Organizational Management* → *Integration With SAP Business Workflow* → *Role Resolution* → *Role Definition* → *Define Roles Using Responsibilities*.

See also:

[Definition Of The Form](#)

Business Add-Ins

Business Add-Ins

Business Add-Ins are a new SAP enhancement technique based on ABAP Objects. They can be inserted into the SAP System to accommodate user requirements too specific to be included in the standard delivery. Since specific industries often require special functions, SAP allows you to predefine these points in your software.

As with customer exits ([SMOD/CMOD \[Ext.\]](#)), two different views are available:

- In the definition view, an application programmer predefines exit points in a source that allow specific industry sectors, partners, and customers to attach additional software to standard SAP source code without having to modify the original object.
- In the implementation view, the users of Business Add-Ins can customize the logic they need or use a standard logic if one is available.

In contrast to customer exits, Business Add-Ins no longer assume a two-system infrastructure (SAP and customers), but instead allow for multiple levels of software development (by SAP, partners, and customers, and as country versions, industry solutions, and the like). Definitions and implementations of Business Add-Ins can be created at each level within such a system infrastructure.

SAP guarantees the upward compatibility of all Business Add-In interfaces. Release upgrades do not affect enhancement calls from within the standard software nor do they affect the validity of call interfaces. You do not have to register Business Add-Ins in SSCR.

The Business Add-In enhancement technique differentiates between enhancements that can only be implemented once and enhancements that can be used actively by any number of customers at the same time.

In addition, Business Add-Ins can be defined according to filter values. This allows you to control add-in implementation and make it dependent on specific criteria (on a specific *Country* value, for example).

All ABAP sources, screens, GUIs, and table interfaces created using this enhancement technique are defined in a manner that allows customers to include their own enhancements in the standard.

A single Business Add-In contains all of the interfaces necessary to implement a specific task. In Release 4.6A, program and menu enhancements can be made with Business Add-Ins.

The actual program code is enhanced using ABAP Objects. In order to better understand the programming techniques behind the Business Add-In enhancement concept, SAP recommends reading the section on [ABAP Objects \[Ext.\]](#).

More information about Business Add-Ins is contained in the following sections:

[Business Add-Ins: Architecture \[Ext.\]](#)

[A Comparison of Different Enhancement Techniques \[Ext.\]](#)

[Defining Business Add-Ins \[Ext.\]](#)

[Calling Add-Ins from Application Programs \[Ext.\]](#)

[Implementing Business Add-Ins \[Ext.\]](#)

[Filter-Dependent Business Add-Ins \[Ext.\]](#)

[Multiple Use Business Add-Ins \[Ext.\]](#)

[Menu Enhancements \[Ext.\]](#)

[Business Add-Ins: Import Procedure \[Ext.\]](#)

Definition Of A Form

Definition Of A Form

Use

You store the form layout for an [internal service request \[Ext.\]](#) in the HTML template definition.

Prerequisites

You have made all Customizing settings under *Cross-Application Components* → *Internet/Intranet Services* → *Internal Service Request* → *Definition of Scenarios* → [Define Scenarios \[Ext.\]](#)

Flow

You can choose to work in either the [Web Application Builder \[Page 70\]](#), or the [ITS-SAP@Web Studio \(BC-FES-ITS\) \[Ext.\]](#).

For more information on the adaptation tools, see the SAP Library under

- *Basis* → *BC-ABAP Workbench* → *Integration of Internet Services* → Web Application Builder
- *Basis* → *Frontend Services (BC-FES)* → ITS-SAP@Web Studio (BC-FES-ITS)

See also:

[Define Form \[Page 54\]](#)

Preparing The Internet Service

You are in Customizing for the [internal service request \[Ext.\]](#) under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Definition Of A Scenario* → [Define Scenarios \[Ext.\]](#).

Before you can use the scenario in the intranet as an internal service request, you need to generate and publish an internet service.

After you have done this, you can process the request form.

See also:

[Generate Service \[Page 48\]](#)

[Publishing The Service \[Page 50\]](#)

[Definition Of The Form \[Page 53\]](#)

Generating Services

Generating Services

Use

You require an internet service as the form for the internal service request.

Prerequisites

In Customizing under *Cross-Application Components* → *Internet-/Intranet Services* → *Internal Service Request* → *Scenario Definition* → [Define Scenario \[Ext.\]](#), you selected *Entry with form* as the *entry type* for the *web entry*.

You now need to generate an *internet service* automatically.

Procedure



You can assign **one** ITS service to maximum **one** scenario.

1. For the internet service, enter a name that has not yet been used.

2. Choose *Generate*.

You can generate a new internet service in three ways:

a. **Generate Internet Service With Reference**

If you specify a scenario with an existing Internet service as a reference when you generate the Internet service, then this is copied to the new Internet service name. You can copy an existing Internet service in the SAP@Web-Studio or Web Application Builder. However, you then need to manually adapt the internet service parameter *SCENARIO* to the new scenario.

For more information on the SAP@Web-Studio, see the *SAP Library* under *Basis Components* → *Frontend Services (BC-FES)* → [ITS-SAP@Web Studio \(BC-FES-ITS\) \[Ext.\]](#) or *Basis Components* → *BC-ABAP Workbench* → *Integration of Internet Services* → [Web Application Builder \[Page 70\]](#).

b. **Generate New Internet Service (SAPDesign)**

If you do **not** specify a scenario with an existing Internet service as a reference when you generate the Internet service, then the system copies the standard Internet service **SR00**.

An Internet service generated in this way contains all the settings required for immediate testing of the scenario definition. The system automatically inserts a *Request data* area (with the corresponding entry fields) for the request-specific characteristics defined in the scenario.

c. *Generate New Internet Service (Plain HTML)*

The same conditions apply as for generation using SAPDesign.



The last two methods of internet service creation are only possible using Customizing for internal service requests.

d. You need to manually publish the internet service before you can test it.

You go automatically to processing for the internet service in the web application builder.

3. Select the service and choose *Publish* → *Entire service*.
Next you can test the internet service in Customizing, by choosing *Test*.

Result

You have created the required HTML templates for the form for your scenario. Next you can adapt the request form according to your requirements.

See also:

[Define Scenario \[Page 40\]](#)

[Publish Services \[Page 50\]](#)

[Define Form \[Page 54\]](#)

[Web Application Builder \[Page 70\]](#)

Publishing a Service

Publishing a Service

Use

In order for an Internet service to be executed by the ITS, it must be stored in the ITS file system. This is known as publishing the service. You can choose to publish the entire service or just parts of it. When you publish the whole service, the corresponding Internet service and its HTML templates are placed in the file system of the AGate server, and the MIME objects are placed in the file system of the WGate server.



Note that by default, the Internet service is published on all Internet Transaction Servers. However, you can choose to restrict the publication to an ITS assigned to your particular R/3 System. To do this, choose *Utilities* → *Settings* and then, under *ITS*, enter the required server. For further information, refer to [User Settings \[Ext.\]](#).

Prerequisites

At least one ITS must have been assigned to the R/3 System, and it must be active.

Procedure

To publish an entire Internet service from the object list:

1. Right-click the relevant service.
2. Choose *Publish* → *Entire service* from the context menu.

If an error occurs while the system is publishing the service, a log is generated containing the relevant message texts.

If no errors occur, the system displays the message *The object has been published successfully*.

Result

Once you have published the entire service, you can start your Web application.

See also:

[User Settings for Internet Services \[Ext.\]](#)

[Executing a Service \[Page 51\]](#)

Executing a Service

Use

Use this function to test a Web transaction or MiniApp from the ABAP Workbench.

Prerequisites

You must already have published the entire service on the ITS. The ITS must be active.

Procedures

To start the Web application from the Object Navigator:

1. Select the relevant service.
2. Choose *Execute*.

The system starts the Web browser and displays a logon window.

3. Check the logon language, and log onto the ITS by choosing *Logon*.
4. Run the Web application.

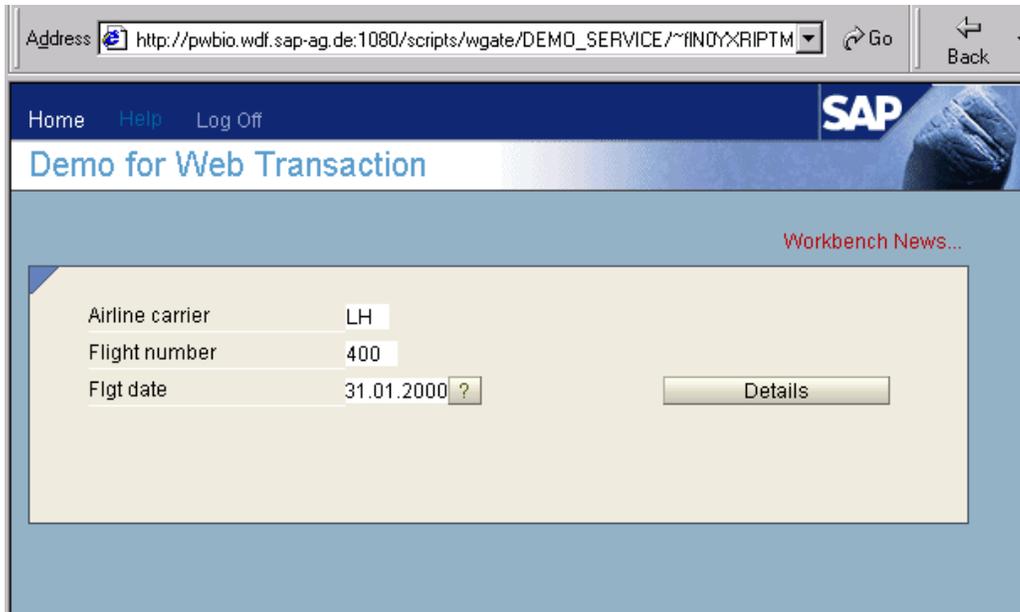


You can suppress the logon window by specifying a logon language in the ~LANGUAGE parameter.

Result

The service is started using the HTTP address **http://<ITS>:<Port>/scripts/wgate/<service>!**

Executing a Service



The screenshot shows a web browser window with the address bar containing the URL: `http://pwbio.wdf.sap-ag.de:1080/scripts/wgate/DEMO_SERVICE/~INDYXRIPTM`. The browser's navigation buttons (Go, Back) are visible. The page header includes links for Home, Help, and Log Off, along with the SAP logo. The main content area is titled "Demo for Web Transaction" and features a "Workbench News..." link. Below this is a form with three input fields: "Airline carrier" with the value "LH", "Flight number" with the value "400", and "Flgt date" with the value "31.01.2000" and a help icon. A "Details" button is positioned to the right of the date field.

Airline carrier	LH
Flight number	400
Flgt date	31.01.2000 ?

Details



The way in which a Web transaction is displayed depends on the classification of the underlying R/3 transaction. *Professional User Transactions* are displayed with the normal R/3 menus, command field, standard toolbar, and application toolbar in the Web browser. *Easy Web Transactions* (EWTs), on the other hand, are displayed with a special EWT header.

Form Definition

Use

In the HTML template definition, you store the form layout for an [internal service request \[Ext.\]](#).

Prerequisites

In Customizing, under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Definition of Scenarios* → [Define Scenarios \[Ext.\]](#) you have made all Customizing settings.

Flow

To define the HTML template, you can choose between working in the web application builder, and the SAP@WEB-Studio.

For more information on adaptation tools, see the SAP Library under:

- *Basis* → *BC-ABAP Workbench* → *Integration of Internet Services* → [Web Application Builder \[Page 70\]](#)
- *Basis* → *Frontend Services (BC-FES)* → [ITS-SAP@Web Studio \(BC-FES-ITS\) \[Ext.\]](#)

See also:

[Defining Forms \[Page 54\]](#)

Define Form

Define Form

Use

You define a form for a specific [internal service request scenario \[Ext.\]](#). This form is later called up and filled out by the person entering the request.

Prerequisites

1. In Customizing under *Cross Application Components* → *Internet-/Intranet Services* → *Internal Service Request* → *Scenario Definition* → [Define Scenario \[Ext.\]](#), you defined a scenario.

For more information on the scenario, see [Definition of A Scenario \[Page 9\]](#).



You define special fields for the request in Customizing for scenarios. You can use the following characteristic categories for this:

- Characteristics with a reference to a category=data element
- Characteristics with a reference to a category=structure
- Uncategorized characteristics

Define Form

You can evaluate characteristics on the HTML template under the name specified there.

The only exceptions are *characteristics with reference to a structure*. In this case, the structure fields can be used with their names (without the prefix of <structure name> or similar names) in the HTML template. This means that structures are continuously broken down for the request and can, therefore, be seen as a more compact definition instead of a selection of single characteristics that refer to data elements.

In all three cases, the characteristic description is available as a field label in the HTML template as well as the name and the characteristic content.

For the categorized characteristics, the field length is also available. In addition, the input/output format is executed for characteristics that refer to data elements or structure fields, in the same way as for screen fields.

Uncategorized characteristics have the implicit **CHAR250** category without input/output conversion.

The following restrictions apply to the definition of scenario characteristics:

- Characteristic names or structure field names must be unique within a scenario.
- Characteristic names or structure field names may not begin with `ISR_`
- Characteristic names or structure field names should not be named as HTML business indicators (for example, LABEL, NAME, VALUE, DIM, MAXSIZE, TYPE, EXISTS, ENABLED or VISSIZE).
- The length of the characteristic names or structure field names should have a **maximum** of 28 characters.
- The data length of the characteristic or structure fields is restricted to 250 characters.

2. You have generated and published an ITS service.

For more information on generating an ITS service, see: [Generating Services \[Page 48\]](#).

Process Flow

An *Internet service* for an internal service request has the following components:

- Service file with ITS parameters
- HTML templates for the different processing modes in the request
- File with the language-dependent resources from the HTML template.

You can process Internet services in the SAP@Web-Studio, or in the Web Application Builder (access from Customizing for scenarios). However, compared with the Web Application Builder, the SAP@Web-Studio has several advantages (syntax check, an editor that is easy to use, clearer management for each of the components and several Internet services at the same time, and improved help functions). You can also switch between tools during processing. You can use the source control in the SAP@Web-Studio to check Internet services in and out of the SAP@Web-Studio.

For more information on adaptation tools, see the SAP Library under:

- *Basis* → *BC-ABAP Workbench* → *Integration of Internet Services* → [Web Application Builder \[Page 70\]](#)
- *Basis* → *SAP Internet Applications* → [SAP@Web Studio \[Ext.\]](#)

Define Form**Service Parameters**

In an *Internet service*, you need the following parameters for an internal service request:

Parameter	Value
SCENARIO	Scenario name
~TRANSACTION	QISR1
~WEBTRANSACTIONTYPE	EWT
~GENERATEDYNPRO	1
~STYLE	DHTML
~AUTOSCROLL	0
~LISTSCROLLING	0
~POPUPS	1

HTML Templates

When you start the internal service request, you call up the **QISR1** transaction in the SAP System (**SAPLQISR3** program). This goes to one of the following screens, regardless of processing mode: For each of these subscreen containers, you can define your own HTML template to generate the corresponding HTML form:

Mode	Screen	HTML Templates
CREATE	110	sapqisr3_110.html
CHANGE	120	sapqisr3_120.html
DISPLAY	130	sapqisr3_130.html

All HTML templates should be assigned to *Default-Theme 99*. You can leave out the HTML template for processing modes that you do not require.

If you do not require any variants for the request for the different processing modes, define the HTML template for the *display* mode. You can enter the following HTML business statement for all other HTML templates:

```
`include(~theme="99", ~language="", ~name="sapqisr3_110.html")`
```

This statement ensures that for each HTML template, the *sapqisr3_110.html* HTML template for the *create* mode is referenced.

In the HTML template for *creating*, if you use the functions from the ISR-HTML library (for SAP Design) or from the ISR-Plain-HTML-Library (for Plain HTML), then the entry fields are automatically switched to *Not ready for input* depending on the processing mode.

Definition Of An HTML Template For The ISR

The simplest HTML template contains the HTML code for displaying the request in the web browser. You can embed the HTML business statement into the HTML template for a dynamic HTML page layout.

For more information on HTMLBusiness, see the *SAP@Web-Studio* or the *Web Application Builder* under *Application Help*. You can use HTML business to create HTML statements under certain circumstances for example (*if (...) ... else ... end;*) or in loops (*for ... ;*). In the HTML template, the following generally applies:

- Statements included in ` (backtick) contain HTML business code
- Statements not included in ` (backtick) contain HTML-Code, which is generated directly into the HTML page.

You can also refer to data from the SAP R/3 System in the HTML template, using HTML business. For example, the

``BUILDING_TO``

statement means that the *BUILDING_TO* characteristic is output in the HTML page that was created using the template.



As well as the special characteristics defined in Customizing for scenarios, you can use the following general data from the internal service request in the HTML template.

Name	Description	Name in BADIs
ISR_HEADER-NOTIF_NO	Notification number	GENERAL_DATA -NOTIF_NO
ISR_HEADER-NOTIF_TYPE	Notification type	GENERAL_DATA-NOTIF_TYPE
ISR_HEADER-SCENARIO	Scenario	GENERAL_DATA-SCENARIO
ISR_HEADER-CREATION_DATE	Creation date	GENERAL_DATA-CREATION_DATE
ISR_HEADER-CREATION_TIME	Creation time	GENERAL_DATA-CREATION_TIME
ISR_HEADER-CALC_COSTS	Calculated costs	GENERAL_DATA-CALC_COSTS
ISR_HEADER-CALC_COSTS_CURR	Currency key for CALC_COSTS	GENERAL_DATA-CALC_COSTS_CURR
ISR_HEADER-LANGUAGE	Language	GENERAL_DATA-LANGUAGE
ISR_HEADER-START_DATE	Start date	GENERAL_DATA-START_DATE
ISR_HEADER-START_TIME	Start time	GENERAL_DATA-START_TIME
ISR_HEADER-REQ_DATE	Required date	GENERAL_DATA-REQ_DATE
ISR_HEADER-REQ_TIME	Required time	GENERAL_DATA-REQ_TIME
ISR_HEADER-PRIORITY	Priority	GENERAL_DATA-PRIORITY
ISR_PAGE	Current page number	PAGE
ISR_HEADER-EXT_REF_NUMBER	External reference number	GENERAL_DATA-EXT_REF_NUMBER
ISR_HEADER-SHORT_TEXT	Short text	GENERAL_DATA-SHORT_TEXT

Define Form

ISR_HEADER-ESTIM_COSTS	Estimated costs	GENERAL_DATA-ESTIM_COSTS
ISR_HEADER-ESTIM_COSTS_CURR	Currency key for ESTIM_COSTS	GENERAL_DATA-ESTIM_COSTS_CURR
ISR_MODE	Current mode (CREATE, DISPLAY, and so on)	MODE
ISR_FORM_VIEW	View	<i>Default:</i> ISR_REQUEST <i>If processor calls the form for the message:</i> ISR_PROCESS
ISR_CREATED_BY-USER_ID	Created by: User name	ISR_GENERAL_DATA-CREATED_BY-USER_ID
ISR_CREATED_BY-FIRSTNAME	Created by: First name	ISR_GENERAL_DATA-CREATED_BY-FIRSTNAME
ISR_CREATED_BY-LASTNAME	Created by: Last name	ISR_GENERAL_DATA-CREATED_BY-LASTNAME
ISR_CREATED_BY-FULLNAME	Created by: Full name	ISR_GENERAL_DATA-CREATED_BY-FULLNAME
ISR_CREATED_BY-TITLE	Created by: Title	ISR_GENERAL_DATA-CREATED_BY-TITLE
ISR_CREATED_BY-DEPARTMENT	Created by: Department	ISR_GENERAL_DATA-CREATED_BY-DEPARTMENT
ISR_CREATED_BY-BUILDING	Created by: Building	ISR_GENERAL_DATA-CREATED_BY-BUILDING
ISR_CREATED_BY-FLOOR	Created by: Floor	ISR_GENERAL_DATA-CREATED_BY-FLOOR
ISR_CREATED_BY-ROOM_NO	Created by: Room number	ISR_GENERAL_DATA-CREATED_BY-ROOM_NO
ISR_CREATED_BY-E_MAIL	Created by: eMail address	ISR_GENERAL_DATA-CREATED_BY-E_MAIL
ISR_CREATED_BY-TEL_NUMBER	Created by: Telephone	ISR_GENERAL_DATA-CREATED_BY-TEL_NUMBER
ISR_CREATED_BY-TEL_EXTENSION	Created by: Telephone extension	ISR_GENERAL_DATA-CREATED_BY-TEL_EXTENSION
ISR_CREATED_BY-FAX_NUMBER	Created by: Fax	ISR_GENERAL_DATA-CREATED_BY-FAX_NUMBER
ISR_CREATED_BY-FAX_EXTENSION	Created by: Fax extension	ISR_GENERAL_DATA-CREATED_BY-FAX_EXTENSION
ISR_CREATED_BY-COMPANY	Created by: Company	ISR_GENERAL_DATA-CREATED_BY-COMPANY
ISR_CREATED_BY-MASTER_CCTR	Created by: Master cost center	ISR_GENERAL_DATA-CREATED_BY-MASTER_CCTR
ISR_CREATED_BY-MASTER_COAREA	Created by: Master controlling area	ISR_GENERAL_DATA-CREATED_BY-MASTER_COAREA
ISR_INITIATED_BY- ...	Person placing request	ISR_GENERAL_DATA-INITIATED_BY- ...
ISR_APPROVED_BY- ...	Person approving request	ISR_GENERAL_DATA-APPROVED_BY- ...
ISR_PROCESSOR- ...	Person processing request	ISR_GENERAL_DATA-PROCESSOR- ...
ISR_CONTACT_PERSON- ...	Contact person for scenario	ISR_GENERAL_DATA-CONTACT_PERSON- ...

The following functions are available in the SAP General HTML-Business-Functions Library to enable you to create the most important form elements (buttons, fields, labels etc.) quickly and easily:

- **SAPDesign**

To make the creation of HTML templates for internal service requests even easier, there is an extension of the *SAP General HTML-Business-Functions Library* for internal service requests. You include this library in HTML templates using the following statement:

```
`include(~service="sr_library", ~theme="99", ~language="",  
        ~name="ISR_Template_Library.html")`
```

If you use this library, you can automatically use the functions of the *SAP General HTML-Business-Functions Library*.

See also:

[Example: Framework of HTML Templates for Internal Service Requests \[Page 60\]](#)

[Example of Specific Form Elements in HTML Templates \(SAPDesign\) \[Page 61\]](#)

- **Plain HTML**

To make it easier to create Plain-HTML-Templates, you can use the following statement to include a library in your template:

```
`include(~service="sr_library", ~theme="99", ~language="",  
        ~name="ISR_Template_Library_Plain_HTML.html")`
```

See also:

[Example: Specific Form Elements in HTML Templates \(Plain HTML\) \[Ext.\]](#)

Example: HTML Framework for HTML Templates

Example: HTML Framework for HTML Templates

The framework of every HTML template for internal service requests should appear as follows:

```
`include(~service="sr_library", ~theme="99", ~language="",
~name="ISR_Template_Library.html)`

`SAP_TopInclude()`

<html>

<head>
  `SAP_PageTitle()`
  `SAP_Stylesheet()`
  `SAP_JavaScript()`
</head>

<body `SAP_TemplateBodyAttributes()` onload=""`SAP_OnloadJavaScript()``">

  `SAP_LoadInfoTable()`
  `SAP_TemplateHeader()`

  `SAP_BodyContentBegin()`
  `SAP_ISRFormBegin()`
  ...`SAP_TemplateFormEnd()`
  ...`SAP_BodyContentEnd()`
</body>
</html>
```

The part of the HTML template that is specific to the scenario begins after the statement ``SAP_ISRFormBegin()``. In this specific part of the HTML template, you can define the form elements for the request.

See also:

[Example of Specific Form Elements in HTML Templates \[Page 61\]](#)

Example: Specific Form Elements in HTML Templates (SAP Design)

The part of the HTML template that is specific to the scenario begins after the statement ``SAP_ISRFormBegin()``.

You can use the following functions to define the form elements for the request. The functions are defined in the *sr_library* Internet Service, in the *isr_template_library* template.



To study the functions, load the form elements in the SAP@Web-Studio, or display the **SR_LIBRARY** internet service in the Web Application Builder. You find the functions using the syntax check.

For more information, see the SAP Library under *Basis* → *SAP Internet Applications* → [SAP@Web Studio \[Ext.\]](#) or *Basis* → *BC-ABAP Workbench* → *Integration of Internet Services* → [Web Application Builder \[Page 70\]](#).

Form Elements



The examples come from the standard scenario *Room change request (SR01)*. To study the examples described, call this scenario.

SAP_ISRGetLabel(name)

This returns the *name* of the description of a special or general characteristic.

Example:

```
`myLabel = SAP_ISRGetLabel(name="BUILDING_TO")`
```

Result:

The *myLabel* variant defined by assignment contains the text "By building".

SAP_ISRGetMaxSize(name)

Returns the maximum length for the *name* characteristic.

Example:

```
`myLabel = SAP_ISRGetLabel(name="BUILDING_TO")`
```

Result:

The *mySize* variant defined by assignment contains the value 2.

SAP_ISRAdjustMode(mode)

Converts the mode for an entry field, dependent on the current processing mode (CREATE or DISPLAY).

Example:

```
`myMode = SAP_ISRAdjustMode(mode)`
```

Result:

If you call the request in *Display* mode, *myMode* contains the value *disabled*. In all other cases, it

Example: Specific Form Elements in HTML Templates (SAP Design)

contains the value transferred using the *mode* parameter.

SAP_ISRFieldListBegin(left, top, labelWidth=SAP_ISRdefaultLabelWidth)

and

SAP_ISRFieldListEnd()

Defines a list of form elements that are arranged under each other in rows. The starting point of the list is defined by *left* (columns from the right edge of the form) and *top* (rows from the top of the form). *labelWidth* (optional parameter with the default value SAP_ISRdefaultLabelWidth = 30) determines the width of the labels for the input fields in the list.

Example:

```
`x = 1; y = 1`
`SAP_ISRFieldListBegin( x , y , 018) `
  `SAP_ISRLabel( labelText="from office", left = x+13); `
  `SAP_ISRField( name="ISR_INITIATED_BY-BUILDING" ) `
  `SAP_ISRField( name="ISR_INITIATED_BY-FLOOR" ) `
  `SAP_ISRField( name="ISR_INITIATED_BY-ROOM_NO" ) `
`SAP_ISRFieldListEnd( ) ``
```

SAP_ISRFieldListRowBegin()

and

SAP_ISRFieldListRowEnd()

Defines a series of form elements in a list of form elements that was created using SAP_ISRFieldListBegin().

Example:

```
`x = 1; y = 1`
`SAP_ISRFieldListBegin( x , y ,018) `
  `SAP_ISRField( name="ISR_INITIATED_BY-FIRSTNAME" ) `
  `SAP_ISRFieldListRowBegin( ) `
    `SAP_ISRField( name="ISR_INITIATED_BY-TEL_NUMBER" ) `
    `SAP_ISRLabel( labelText="-", left = x+23); `
    `SAP_ISRField( name="ISR_INITIATED_BY-TEL_EXTENSION",
fieldLabel="noLabel", left=x+22 ) `
  `SAP_ISRFieldListRowEnd( ) `
    `SAP_ISRField( name="ISR_INITIATED_BY-E_MAIL" ) `
    `SAP_ISRField( name="ISR_INITIATED_BY-DEPARTMENT" ) `
`SAP_ISRFieldListEnd( ) `
```

SAP_ISRFieldListRowEnd()

Defines an empty row a list of form elements that was created using SAP_ISRFieldListBegin().

Example: Specific Form Elements in HTML Templates (SAP Design)

SAP_ISRField(id="", fieldLabel="noLabelSpec", fieldLabelWidth=0, name, mode="enabled", size=0, maxLength=0, inspectionText="", left=0, top=0)

Defines an input field with a label and optional text after the input field.

The individual parameters mean the following:

id

optional

The ID of the form element, if you want to work in JavaScript.

fieldLabel

optional

Label before the input field. *Label="noLabel"* results in no label being output.

fieldLabelWidth

optional

Label width.

It is not necessary to specify a width in a field list (see above).

name

Name of the characteristic for which you create the input field.

mode

Optional

enabled (ready for input) or *disabled* (not ready for input)

If you define an input field as *ready for input* but call up the request in the *display* mode, then the system automatically switches to *not ready for input*.

size

optional

Output length of the input field.

If you do not specify an output length, the system automatically determines the output length of the field.

maxLength

Optional

Maximum length of the entry in the input field.

If you do not specify an output length, the system automatically determines it.

inspectionText

optional

Text after the input field.

left

optional

Position of the input field, calculated in columns from the left edge of the form.

If you use a field list, the coordinate is calculated automatically.

top

optional

Position of the input field, calculated in rows from the top edge of the form.

If you use a field list, the coordinate is calculated automatically.

Example:

```
`SAP_ISRField( name="ISR_INITIATED_BY-FIRSTNAME" )`
```

Example: Specific Form Elements in HTML Templates (SAP Design)

SAP_ISRPullDownList(id="", fieldLabel="noLabelSpec", fieldLabelWidth=0, name, key, content="", mode="enabled", size=0, maxLength=0, inspectionText="", onchange="", left=0, top=0)

Defines a combo box with a label and optional text after the combo box. In the *mode="disabled"* mode, the system displays a field not ready for entry (rather than a combo box) that contains the value currently selected.

The meaning of a parameter corresponds to its function in **SAP_ISRField()**.

If you use a field list, the coordinate is calculated automatically.

Additional parameters:

key

Name of the table containing the key values for the combo box (for example, you can create the table in *BADI_SCENARIO_SET_ADDITIONAL_VALUES*). Values from the *key* table are not issued. If you select a row in the combo box, the corresponding value from the *key* table is set in the characteristic as *name*.

content

Name of the table containing the labels for the combo box (for example, you can create the table in *BADI_SCENARIO_SET_ADDITIONAL_VALUES*). These are issued in the combo box list.

Example:

```
`SAP_ISRPullDownList(name="BUILDING_TO",
key="BUILDING_OPTION", content="BUILDING_OPTION")`
```

SAP_ISRUserName(id="", fieldLabel="noLabelSpec", fieldLabelWidth=0, name, user, mode="enabled", size=0, maxLength=0, inspectionText="", left=0, top=0)

Defines input fields for the parts of a name:

...-FIRSTNAME

...-LASTNAME

...-FULLNAME

A special feature of this function is that if you enter an ambiguous name, or a part of a name and a wild card (*), in the name field, the system generates all users that match this entry. Instead of an input field for the parts of the name, the system presents these users in a combo box, from which you make your selection. The name of the user chosen is copied to the input fields for the parts of the name.

The meaning of a parameter corresponds to its function in **SAP_ISRField()**.

Additional parameters:

user

Name of the characteristic for the *UserID* that belongs to the *name* entered in the input fields.

Example:

```
`SAP_ISRUserName( name="ISR_INITIATED_BY-LASTNAME",
user="ISR_INITIATED_BY-USER_ID")`
```

SAP_ISRGroupBox(left, top, width, height, groupBoxLabel)

Defines a group box by the coordinates *left*, *top*, *width* and *height*.

GroupBoxLabel is the label for the box.

Example:

Example: Specific Form Elements in HTML Templates (SAP Design)

``SAP_ISRGroupBox(001, 001, 034, 008, FRAME_INITIATED_BY.label)``

or

``SAP_ISRGroupBox(001, 001, 034, 008, „Rahmentext“)``

Result:

Draws box for the person making request, or box with the label *Rahmentext*.

SAP_ISRLongTextInput(cols=80, rows=4, left=0, top=0)

Defines the input field for the request long text by its position from the *left* and *top*, and with *cols* (columns) and *rows*.

Example:

``SAP_ISRLongTextInput(cols=110, rows=5, left = 1, top = 1)``

SAP_ISRCheckBox(id="", name, mode="enabled", text="noTextSpec", left=0, top=0, labelWidth=0)

Defines a check box for the *name* field by its position from the *left* and *top*, with the label *text* and *labelWidth*. If you use a field list, the coordinate is calculated automatically.

Example:

``SAP_ISRCheckBox(name="CHANGE_PHONE")``

SAP_ISRLabel(labelText="", labelIconName="", id="", left=0, top=0)

Defines a simple text on the form.

If you use a field list, the coordinate is calculated automatically.

Example:

``SAP_ISRLabel(labelText="einText")``

SAP_ISRStandardButtons(left , top, buttonWidth=018)

Outputs the necessary standard pushbuttons for each session in the *left*, *top* position. *buttonWidth* is the width of every button.

Example:

``SAP_ISRStandardButtons(1 , 20)``

Example: Specific Form Elements in HTML Templates (SAP Design)

Example: Specific Form Elements in HTML Templates (SAP Design)

To execute the output of error messages, you call the JavaScript function `SAPBodyOnload()` in the body tag:

```
<body onload="SAPBodyOnload()">
```

After the two statements necessary for the definition of the form:

```
<form id="myForm" name="myForm" action="\wgateURL()" method="post">
`include(~service="sr_library", ~theme="99", ~language="",
~name="ISR_Template_Library_Plain_Html.html")`
```

there is the part of the HTML template that is specific to the scenario.

You can use the following functions to define the form elements for the request.

The functions are defined in the `sr_library` Internet Service.



To study the functions, load the form elements in the SAP@Web-Studio, or display the **SR_LIBRARY** internet service in the Web Application Builder. You find the functions using the syntax check.

For more information, see the SAP Library under *Basis* → *SAP Internet Applications* → [SAP@Web Studio \[Ext.\]](#) or *Basis* → *BC-ABAP Workbench* → *Integration of Internet Services* → [Web Application Builder \[Page 70\]](#).

Form Elements

getLabel(name)

This returns the *name* of the description of a special or general characteristic.

Example:

```
`myLabel = getLabel(name="BUILDING_TO")`
```

Result:

The `myLabel` variant defined by assignment contains the text "To building".

getSize(name, maxwidth)

This returns the *name* length of a characteristic. To enter a maximum length, you use the optional parameter *maxwidth*.

Example:

```
`mySize = getSize(name="ISR_HEADER-NOTIF_TYPE")`
```

Result:

The `mySize` variant defined by assignment contains the value 2.

getName(name, index)

This returns the *name* of the characteristic. To specify a row index for access to tables, you can use the optional parameter *index*.

Example: Specific Form Elements in HTML Templates (SAP Design)

Example:

```
`myName = GetName(name="BUILDING_TO")`
```

Result:

The *myName* variant defined by assignment contains the text "To building".

Disabled(name)

This sets the display mode for the *name* to correspond to the request mode (create, display, change). You use this function in the HTML tag `<input type="text" ...>` `<input type="radio" ...>` `<input type="checkbox" ...>` and `<select ...>`.

Example:

```
<input type="text" name="BUILDING_TO" value="BUILDING_TO" `
  Disabled("BUILDING_TO")`>
```

Result:

The input field is ready for input in the *Create* and *Change* request modes only, not in *Display* mode.

Readonly(name)

This sets the display mode for the *name* to correspond to the request mode (create, display, change). You use this function in the HTML tag `<textarea ...>`.

Example:

```
<textarea wrap="physical" rows="12" cols="65"
  name="`GetTextAreaName("MY_TEXT",65)`" `readonly("MY_TEXT")`>
```

Result:

The text field is ready for input in the *Create* and *Change* modes only, not in *Display* mode.

Selected(name, value)

This sets the correct entry in a dropdown list box dependent on the *name* value. You use this function in the HTML tag `<option...>`.

Example: Dropdown list box (variant 1)

```
<select id="COLOR" name="COLOR" size=1 `disabled("COLOR")`>
  <option value="red" `Selected("COLOR","red")` > you like the red color?
  <option value="green" `Selected("COLOR","green")`> or the green color?
  <option value="blue" `Selected("COLOR","blue")`> or the blue color?
</select>
```

Result:

If you chose the "or the green color?" entry, the value "green" is assigned to the characteristic "COLOR". If you call up the sent form in display mode, the "or the green color?" entry is displayed automatically.

SetOptions(name, options, labels)

This enables you to set up a dropdown list box for the *name* characteristic using the corresponding *options* and *labels*. You use this function after the HTML tag `<select...>`.

Example: Dropdown list box (variant 2)

Example: Specific Form Elements in HTML Templates (SAP Design)

```

`COLOR_Option[1] = "red";
COLOR_Option[2] = "green";
COLOR_Option[3] = "blue";
COLOR_Label[1] = "you like the red color?";
COLOR_Label[2] = "or the green color?";
COLOR_Label[3] = "or the blue color?";

<select id="COLOR" name="COLOR" size="1" `disabled("COLOR")>
  `SetOptions("COLOR","COLOR_OPTION","COLOR_LABEL")`
</select>

```

Result:

If you chose the "or the green color?" entry, the value "green" is assigned to the characteristic "COLOR". If you call up the sent form in display mode, the "or the green color?" entry is displayed automatically.

Checked(name, index, value)

This sets the correct selection for radio buttons and checkboxes dependent on the *name* value. You use this function in the HTML tags `<input type="radio" ...>` and `<input type="checkbox" ...>`.

Example: Radio buttons

```

`disabled("COLOR")`>you like the red color?<br>
  <input name="COLOR" value="red" type="radio" `checked("COLOR",1,"red")`
  <input name="COLOR" value="green" type="radio" `checked("COLOR",1,"green")`
  <input name="COLOR" value="blue" type="radio" `checked("COLOR",1,"blue")`
`disabled("COLOR")`>or the green color?<br>
`disabled("COLOR")`>or the blue color?<br>

```

Result:

If you chose the "or the green color?" entry, the value "green" is assigned to the characteristic "COLOR". If you call up the sent form in display mode, the "or the green color?" entry is selected automatically.

CheckboxProlog(name, index)

Necessary prolog for the implementation of checkboxes. This sets the correct entry in checkboxes dependent on the *name* value. You use this function directly before the HTML tag `<input type="checkbox" ...>`.

Example: Checkbox

```

`CheckboxProlog("COLOR[1]")`
  <input name="COLOR[1]" value="red" type="checkbox" `Checked("COLOR",1,"red")`
  <input name="COLOR[2]" value="green" type="checkbox"
`Checked("COLOR",2,"green")` `disabled("COLOR")`>and the green color? <br>

```

Example: Specific Form Elements in HTML Templates (SAP Design)

```
`CheckboxProlog("COLOR[3]")`
```

```
<input name="COLOR[3]" value="blue" type="checkbox"
`Checked("COLOR",3,"blue")` `disabled("COLOR")`>and the blue color? <br>
```

Result:

If you chose both the "and the green color?" and "and the blue color?" entries, the values "green" and "blue" are assigned to the characteristic "COLOR" in the second and third rows respectively. If you call up the sent form in display mode, the "and the green color?" and "and the blue color?" entries are selected automatically.

getTextAreaName(name, width)

Returns the *name* and the *width*.

Example:

```
<textarea wrap="physical" rows="5" cols="75"
name="`GetTextAreaName("ISR_TEXT",75)`" `readonly("ISR_TEXT")`>
```

getTextAreaContent(name)

Returns the *name*.



Normally you use both *ISR_NEW_TEXT*, for the entry of new comments, and *ISR_TEXT*, for the display of existing comments (see examples 1 + 2 below). The comments are logged chronologically in the long text for the message.

Example 1: Output field for ISR long text

```
`if ( ISR_MODE != "CREATE" );`
  <textarea wrap="physical" rows="5" cols="75"
name="`GetTextAreaName("ISR_TEXT",75)`" `readonly("ISR_TEXT")`>
  `GetTextAreaContent("ISR_TEXT")`
</textarea>
`end`
```

Example 2: Input field for ISR long text

```
`if ( ISR_MODE != "DISPLAY" );`
  <textarea wrap="physical" rows="5" cols="75"
name="`GetTextAreaName("ISR_NEW_TEXT",75)`" `readonly("ISR_NEW_TEXT")`>
  `GetTextAreaContent("ISR_NEW_TEXT")`
</textarea>
`end`
```

Web Application Builder

Purpose

The Web Application Builder allows you to create Web development objects within the ABAP Workbench. Existing R/3 transactions require these objects to allow them to run as Web transactions in a Web Browser. You can also use the Web Application Builder as an integrated environment for creating MiniApps.

Integration

The Web Application Builder is a fully integrated tool within the ABAP Workbench. Objects that you create with it, such as service files, HTML templates, and MIME objects, are stored in the R/3 Repository and are connected to the R/3 Change and Transport System.

Features

- Creating Internet services for existing R/3 transactions or MiniApps.
- Implementing the dialog logic.
- Generating the HTML templates for the screens of a transaction. These contain standard HTML and HTML^{Business} statements that map the screen layout.
- Editing the generated HTML templates using HTML and HTML^{Business} to develop them further.
- Including MIME objects (icons, graphics, Java applets, animation...) to improve the layout further.
- Creating language-specific texts (language resources).
- Publishing the services or individual service components on the Internet Transaction Server (ITS)
- Executing the complete Web transaction from the ABAP Workbench.
- Connection to the Change and Transport System (CTS).
- Connection to Version Management.

Constraints

Certain functions are not yet available:

- There is no syntax check
- HTML^{Business} and the flow logic are not yet integrated with the Debugger.

Creating HTML Templates

Use

When you implement a MiniApp, you must create HTML templates. The dialog logic of a MiniApp runs on the ITS, not in R/3.

For each transaction, you can choose whether you want to generate HTML templates for all screens, for some screens ([mixed mode \[Ext.\]](#)), or at all. Templates that you create explicitly are identical to the HTML documents that are generated automatically by the WebGUI.

Generating templates explicitly is useful if the WebGUI features are insufficient for your needs and you would need to adapt the standard generated template anyway. This will particularly be the case if you are trying to improve the layout of a screen or if you want to include hyperlinks.



Standard template generation from the WebGUI should be sufficient for most transactions. The WebGUI can display the screen elements of a simple transaction (text fields, input/output fields, checkboxes, radio buttons, tabstrip controls, table controls, subscreens...) without you having to go to the effort of creating a template.

Prerequisites

- You must already have created the service.
- You have sufficient knowledge of HTML and HTML^{Business} to take advantage of the template-based approach.

Procedure

To create an HTML template from the tree display in the object list:

3. Right-click the name of the service.
4. From the context menu, choose *Create* → *Template*.

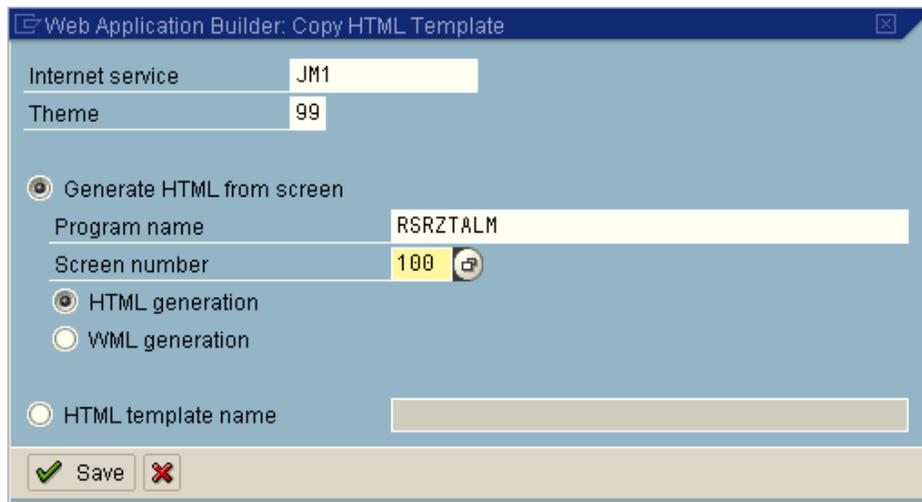
The *Create Template* dialog box appears.

5. Enter the theme for the service and fill out the remaining fields.

If the Web application is a Web transaction and you want to generate a template for a particular screen, select *Generate HTML from screen* and enter the program name and screen number.

If the application has no corresponding R/3 screen (MiniApps), select *Name of template* and enter the name.

Creating HTML Templates



6. Confirm by choosing  Save.

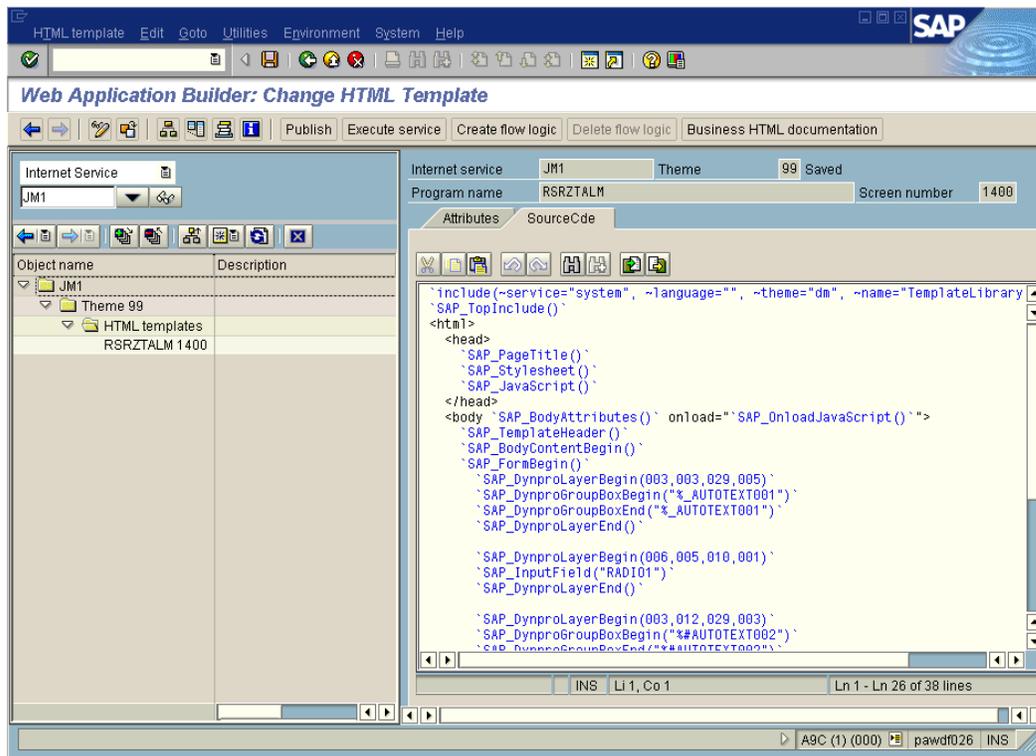
The *Create Object Catalog Entry* dialog box appears.

7. Assign the template to a development class and choose .

Result

The generated template appears in the object list under *Templates*. The generated contents of the template are displayed in the Editor. Only the static screen information is evaluated - an HTML^{Business} function is inserted in the template for each screen element. These are highlighted in blue. You can now change the contents of the template using standard HTML and HTML^{Business}.

Creating HTML Templates

**See also:**[Extending HTML Templates \[Page 74\]](#)[Adding MIME Objects \[Page 75\]](#)[Publishing Services \[Page 50\]](#)

Extending HTML Templates

Extending HTML Templates

Once you have created an HTML template, you can change the generated source code.

To do this, you must be familiar with the basics of HTML and HTML^{Business}.



HTML^{Business} is an extension of standard HTML developed by SAP to allow R/3 screen data to be merged dynamically with information on HTML templates and to make it easier for the ITS to exchange data between the R/3 System and the Web Server.

For further information, refer to [HTMLBusiness Reference \[Ext.\]](#).

Example

This example sets a hyperlink to a particular position on an HTML page:

```
Internet service JM1 Theme 99 Saved
Program name RSRZTALM Screen number 1400
Attributes SourceCode
`SAP_DynproLayerBegin(006,005,010,001)`
`SAP_InputField("RADIO1")`
`SAP_DynproLayerEnd()`

`SAP_DynproLayerBegin(003,012,029,003)`
`SAP_DynproGroupBoxBegin("%#AUTOTEXT002")`
`SAP_DynproGroupBoxEnd("%#AUTOTEXT002")`
`SAP_DynproLayerEnd()`

`SAP_DynproLayerBegin(006,013,009,001)`
`SAP_Button("PUSH1")`
`SAP_DynproLayerEnd()`

`SAP_DynproLayerBegin(020,013,008,001)`
`SAP_Button("PUSH2")`
`SAP_DynproLayerEnd()`

`SAP_FormEnd()`
`SAP_ContentTypeEnd()`

`SAP_DynproLayerBegin(051,002,015,001)`
`<a href="http://workbench:1080" style="color: rgb(187,0,0)">Workbench Ne`
`SAP_DynproLayerEnd()`
* INS Li 38, Co 5 - Li 41, Co 1 Ln 18 - Ln 42 of 42 lines
```

See also:

[Publishing a Service \[Page 50\]](#)

[Adding MIME Objects \[Page 75\]](#)

Adding MIME Objects

Use

You can use MIME objects (icons, graphics, audio files, animations...) to improve the layout of your Web applications.

Prerequisites

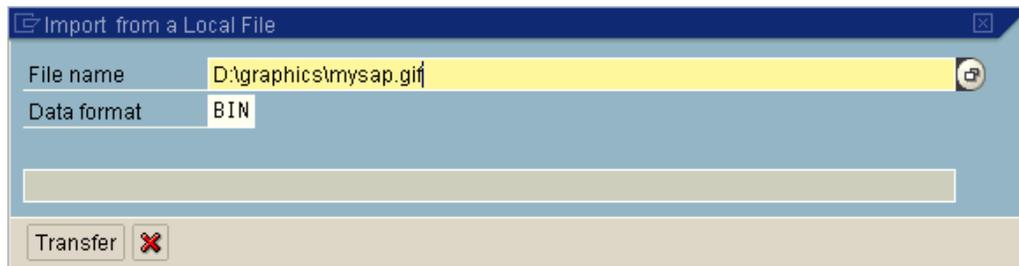
You must already have created an Internet service.

Procedure

To add a MIME object to an Internet service from the object list:

8. Right-click the relevant service.
9. In the context menu, choose *Create* → *Mime*.

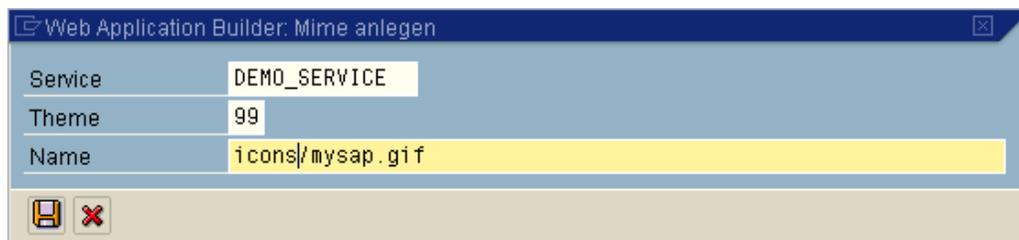
The *Read from Local File* dialog box appears:



10. Enter the path name of the file you want to import, and ensure that the file format is correct.
11. Choose *Import*.

The *Create Mime* dialog box appears.

12. Enter the theme and the name for the MIME object.
13. In the Name field, you can create a subdirectory, separated from the name of the MIME object by a forward slash ("/").



14. Choose  to continue.

The *Create Object Catalog Entry* dialog box appears.

15. Assign the MIME object to a development class and choose .

Adding MIME Objects

Result

The MIME object has been inserted in the R/3 Repository as a standalone object. It appears under Mimes in the object list display, and, if it is a graphic, its contents are displayed.

You can now use this object in your interface design.



When you publish the service, the MIME objects are not stored in an ITS directory. Instead, they are stored on the HTTP server under the name and subdirectory you specified in step 5 above.

See also:

[Publishing Web Applications \[Page 50\]](#)

Entering An Internal Service Request

Use

You want to apply for a business transaction (such as a service or change to data in the SAP System), or you are looking for a solution to a problem.

Integration

The internal service request represents a flow between the intranet and the SAP System.

Intranet	SAP System
Entered by	Person who processes/approves the request

Prerequisites

You are working in the employee work place, or you can call up a special internal service request from the SAP System.

Features

The internal service request covers the following business processes:

- [Searching for a solution \[Page 78\]](#)
- [Requests that do not require approval \[Page 81\]](#)
- [Requests that require approval \[Page 84\]](#)

Activities

You enter your internal service request in the employee work place.

The initial screen has the following structure:

Searching For Solutions: Flow

The screenshot displays a user interface for searching for solutions. At the top, a light blue box contains the text "Jeanette, what can I do for you?". Below this, there are two main sections. The first section, titled "Initiated by", features two input fields and a "Change" button. The second section, titled "My request", includes a small input field, a larger "Request or problem" input field, and three buttons at the bottom: "Delete", "Status", and "Solutions".

Searching For Solutions: Flow

Use

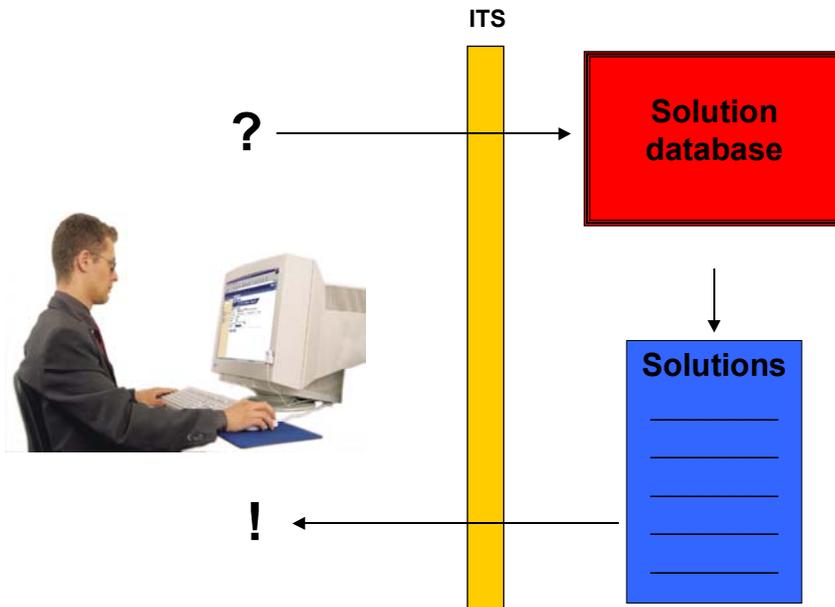
You are looking for a solution to a certain problem.

Prerequisites

You are working in the mySAP.com Workplace.

You have called up the [internal service request \[Ext.\]](#) from your launchpad in the Workplace via *Create internal service request*.

Flow

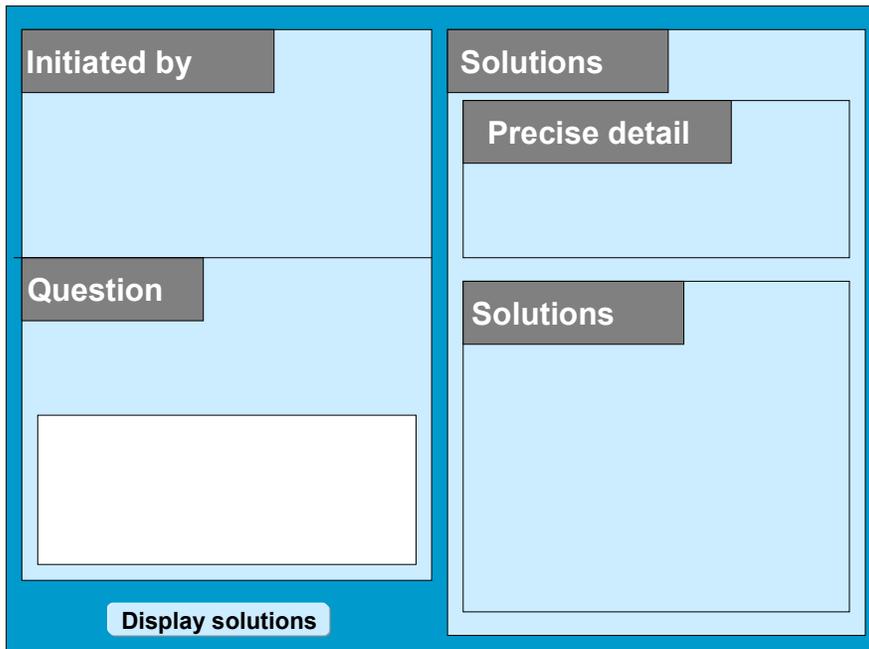


Activities

Enter a problem description. The system uses a search engine to find you solutions to this problem. You do not need to restrict your text to key words when you enter it. The search engine does not just filter the text using the words, but also includes their grammar, and uses word families for the search.

For more information on the search machine, see the SAP Library under *Financials* → *Controlling* → *Cost Center Accounting* → *Information System* → *Interactive Information System*, then [search for report documentation \[Ext.\]](#) under the *Document Search Using The Retrieval System*.

Searching For Solutions: Flow



Result

You have found solutions for the problem.

Request Without Approval Requirement: Flow

Use

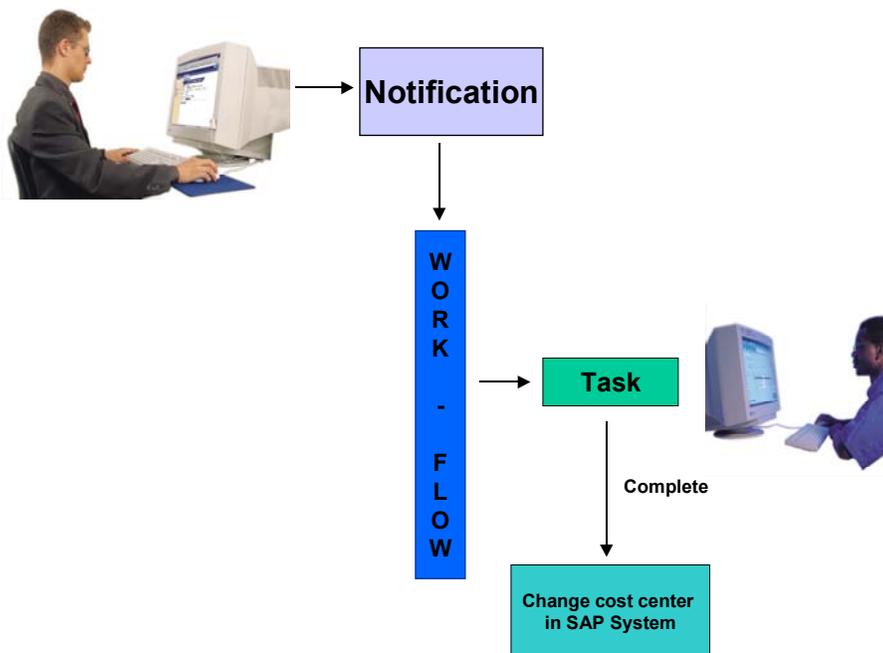
You want to apply for a business transaction (such as a service or change to data in the SAP System), or you are looking for a solution to a problem.

Prerequisites

You are working in the mySAP.com Workplace.

You have called up the [internal service request \[Ext.\]](#) from your launchpad in the Workplace via *Create internal service request*.

Flow



Activities

You are looking for a form suitable for your request.

Request Without Approval Requirement: Flow

The screenshot shows a web interface for creating an internal service request. At the top, a light blue box contains the text "Jeanette, what can I do for you?". Below this is a section titled "Initiated by" with two input fields and a "Change" button. The next section is titled "My request" and contains a single input field. Below that is a section titled "Request or problem" with a larger text area. At the bottom, there are three buttons: "Delete", "Status", and "Solutions".

You select a suitable request or the system suggests a form when you enter a free text describing your request. The system uses a search engine to find you solutions to this problem. You do not need to restrict your text to key words when you enter it. The search engine does not just filter the text using the words, but also includes their grammar, and uses word families for the search.

For more information on the search machine, see the SAP Library under *Financials* → *Controlling* → *Cost Center Accounting* → *Information System* → *Interactive Information System*, then in *Report Documentation*, search under the *Document Search Using The Retrieval System*.

The system calls up the selected request.

Request Without Approval Requirement: Flow

The image shows a screenshot of a request form. The form is set against a blue background. It features three input fields: 'Initiated by' (top left), 'Entered by' (top right), and 'Remarks' (bottom). Below the fields are two buttons: 'Check' and 'Send request'.

Example of a request form

The system has already provided the *Entered by* field with your data.

If required, you enter the initiator name (if you are making an entry for a colleague for example)

The form displays all the fields that are important for your requirements, in which you make your entries.

If required, enter an explanatory free text.

Now you send the request, and

the system finds the correct processor for your request and informs that person.

You can check your request status at any time.

See also:

[Processing Notifications \[Page 96\]](#)

[Processing Tasks \[Page 108\]](#)

[Status Query \[Page 117\]](#)

[Example For Entering And Processing An Internal Service Request \[Page 128\]](#)

Request With Approval Requirement: Flow

Request With Approval Requirement: Flow

Use

You want to apply for a business transaction (such as a service or change to data in the SAP System), or you are looking for a solution to a problem.

Prerequisites

You are working in the mySAP.com Workplace.

You have called up the [internal service request \[Ext.\]](#) from your launchpad in the Workplace via *Create internal service request*.

Flow



Activities

You are looking for a form suitable for your request.

Request With Approval Requirement: Flow

The screenshot shows a web interface for creating an internal service request. At the top, a light blue box contains the text "Jeanette, what can I do for you?". Below this is a section titled "Initiated by" with two input fields and a "Change" button. The next section is titled "My request" and contains a single input field. Below that is a section titled "Request or problem" with a larger text area. At the bottom of the form are three buttons: "Delete", "Status", and "Solutions".

You select a suitable request or the system suggests a form when you enter a free text describing your request. The system uses a search engine to find you solutions to this problem. You do not need to restrict your text to key words when you enter it. The search engine does not just filter the text using the words, but also includes their grammar, and uses word families for the search.

For more information on the search machine, see the SAP Library under *Financials* → *Controlling* → *Cost Center Accounting* → *Information System* → *Interactive Information System*, then in *Report Documentation*, search under the *Document Search Using The Retrieval System*.

The system calls up the selected request.

Request With Approval Requirement: Flow

The screenshot shows a SAP form for creating an internal service request with an approval requirement. The form is set against a blue background and contains several input fields and buttons. At the top, there are two fields: 'Initiated by' and 'Entered by'. Below these is a 'Budget request' section containing 'Budget consumer' and 'Budget object' fields. An 'Approved by' field is located below the 'Budget object' field. At the bottom of the form is a large 'Remarks' text area. Two buttons, 'Check' and 'Send request', are positioned at the very bottom of the form.

Example of a request form

The system has already provided the *Entered by* field with your data.

If required, you enter the initiator name (if you are making an entry for a colleague for example)

The form displays all the fields that are necessary for your requirements, in which you make your entries.

If required, enter an explanatory free text.

Now you send the request.

As it requires approval, it is sent to the approving manager that you specified.

This manager approves the request (possibly after calculating the costs).

The system finds the correct processor for your request and informs that person.

You can check your request status at any time.

See also:

[Approval Requirement For Notifications \[Page 87\]](#)

[Processing Notifications \[Page 96\]](#)

[Processing Tasks \[Page 108\]](#)

[Status Query \[Page 117\]](#)

[Example For Entering And Processing An Internal Service Request \[Page 128\]](#)

Approval Requirement for Notifications

Use

For optimal notification processing in your company, you can active an approval step **for each notification type** in Customizing, which controls whether a newly created notification can be processed further. If an approval requirement exists for a newly created notification, it initially receives the status *Submit for approval* and it cannot be processed further at this point. When this status is set, a person or department responsible must explicitly make a decision about the approval requirement. For this purpose, the *SAP Business Workflow* starts a workflow that automatically informs this person or department that a notification is pending for approval. The following processing options are possible, depending on the decision that is made:

- The person or department responsible gives an approval.
In this case, the notification is released for further processing (this means that the business transaction *Give approval* is executed, which simultaneously sets the statuses *Approval OK* and *Notification in process* in the notification). You can then process the notification in the usual manner (for example, tasks can be created).
- The person or department responsible **does not** give an approval.
In this case, the notification is not released for further processing (this means that the business transaction *Refuse approval* is executed, which simultaneously sets the statuses *Approval refused* and *Notification completed* in the notification).

In this way, the person who created the notification can monitor the current notification status and the decision (in case of an approval requirement) from his or her workplace.

Prerequisites

In Customizing for *Cross-Application Components* under *Notification* → *Notification Creation* → *Partner* → *Define Partner Determination Procedure (Assign Partner Functions for Notification Type)*, the *Approval required* indicator is set for the desired **notification type**.

Scenarios Provided

Scenarios Provided

The scenarios provided not only differ in their purpose, but also whether they incur costs or not.

You can copy these scenarios and adjust them to your requirements.

General Request

Purpose

You should only use the general request if you have not included a suitable scenario for your special [internal service request \[Ext.\]](#). This could happen, for example, for unusual requests for which you do not have a user-defined scenario in Customizing.

Special Features

There are no request-specific fields in the general request. You can only enter your service request in a text.

See also:

[Example For Entering And Processing An Internal Service Request \[Page 128\]](#)

Request For Office Removal

Request For Office Removal

Purpose

You use the removal request if you want to order a removal.

Special Features

As carrying out a removal request incurs costs, you can use a template for the cost calculation.

See also:

[Example For Entering And Processing An Internal Service Request \[Page 128\]](#)

Request For Budget Increase

Purpose

You want to apply for a budget increase for an order, network, network activity, or a WBS element.

You do **not** know who approves this operation.

Special Features

There are two different scenarios:

- *Request for budget increase* (scenario SR61)
The form is sent to a processor, who changes the budget, and executes the budget transactions in the R/3 System.
- *Request for budget with approval* (scenario SR62)
The form is sent to someone for approval.
The manager approves the request. The request processor receives a message via the role determined in Customizing, and via workflow.
For more information on Customizing scenarios, see the implementation guide (IMG) under *Cross-Application Components* → *Internet / Intranet Services* → *Internal Service Request* → *Scenario Definition* → [Define Scenarios \[Ext.\]](#) or in the *SAP-Library* under *Cross-Application Components* → [Internal Service Request \[Ext.\]](#).

For more information on requests for budget increases, see the *SAP Library* under *Financials* → *Controlling* → *Internal Orders* → *Budget Management* → [Request For Budget Increase \[Ext.\]](#).

Request For Master Data Change

Request For Master Data Change

Purpose

If you establish that certain master data needs to be changed, but do not know who the responsible processor is, you can make a request for master data changes. The request is transferred to the SAP R/3 System where it is automatically sent to the responsible processor.

You can request the following changes to master data:

Object	Type of Change
Cost centers [Ext.]	Create, change, lock
Internal orders [Ext.]	Change
G/L accounts [Ext.]	Create, change
Vendors [Ext.]	Create, change, delete
Customers [Ext.]	Create, change, delete

Special Features

You can call up the request form from the work place, and from different SAP applications (depending on the object).

Request Flow

1. You call up the request form and enter the required master data change.
2. Then you send the form.

The request is then transferred to the SAP R/3 System.

3. The responsible processor receives the request in their worklist or workflow inbox.

The processor can reject the request, forward it, or make the requested changes.

For more information, see: [Request For Master Data Changes In The Intranet/Internet \[Ext.\]](#).

Request For An Adjustment Posting

Purpose

If you establish that an error occurred during a posting, but you do not know who the responsible processor is, you can request an adjustment posting. The request is transferred to the SAP R/3 System where it is automatically sent to the responsible processor.

Special Features

You can call up the request form from the work place, and from the following SAP applications:

- Line item report (CO)
- General document display in Controlling (CO)
- General document display in Financial Accounting (FI)

Request Flow

4. You call up the request form and enter the required adjustment posting.
5. Then you send the form.

The request is then transferred to the SAP R/3 System.

6. The responsible processor receives the request in their worklist or workflow inbox.

The processor can reject the request, forward it, or make the requested changes.

For more information, see: [Request An Adjustment Posting In The Intranet \[Ext.\]](#).

Change Notification

Change Notification

Purpose

With a change notification, you arrange the changes to product data that is controlled by Engineering Change Management.

You create a change notification for example, if you want to inform the engineer of a design error or if you want to suggest a product improvement.

Special Features

The change notification:

- has to be approved
- does **not** carry costs

Order Process up to Execution

1. You create the subject as simple text in the change notification.
In addition a little knowledge of the Engineering Change Management or of the work processes in design or work scheduling is essential.
2. The person responsible gets the notification via a workflow or in his worklist and then checks the subject.
3. The processor either approves or rejects the query.
If the query is rejected, the process ends.
4. If the processor has approved the query, he carries out one or more of the following activities:
 - Create change notification (as follow-up notification)
 - Assign ECR
 - Create ECR

The person, who created the ECR, defines the necessary settings such as the [change type \[Ext.\]](#). The other processes in Engineering Change Management are controlled by the change type.

See also:

[Activities for Change Notifications \[Ext.\]](#)

[ECR / ECO \[Ext.\]](#)

Processing The Internal Service Request

After you entered an [internal service request \[Ext.\]](#), the system found the responsible processors using the details given in the request and their definitions in Customizing.

To process the internal service request, choose:

	With Worklist	Without Worklist
For Notifications	<i>Office → Notification → Worklist → Notifications.</i>	<i>Office → Notification → Change</i>
For Tasks	<i>Office → Notification → Worklist → Tasks.</i>	<i>Office → Notification → Task → Process</i>

Processing Notifications

Processing Notifications

To process the [internal service request \[Ext.\]](#), you can do the following:

- [Processing notifications \[Page 97\]](#)
- [Processing notifications using the worklist \[Page 99\]](#)

You call up processing via:

- *Office* → *Notification* → *Worklist* → *Notifications* or
- *Office* → *Notification* → *Change*

Process Notification

Use

Put notification in process

Once you have created a notification and entered all relevant information, you can put the notification in process. When you put a notification in process, you release the notification for processing.

A notification can be put in process in the following ways:

- You can put a notification in process manually.
- The system puts a notification in process automatically if an order was created for the notification.

When a notification is put in process, the system changes the status of the notification from “outstanding” (*OSNO*) to “notification in process” (*NOPR*).

Postpone notification

In some cases, it may be necessary to delay the processing of a notification (for example, if the assigned tasks cannot be carried out yet).

Using this function, you can postpone the processing of a notification. When you postpone a notification, it receives the status *NOPO* (*notification postponed*).



You cannot postpone a notification that has already been put in process.

Complete notification

You complete a notification when the problem has been solved and no further processing is necessary. Once you complete a notification, you can no longer change any data in the notification. You can then only display the notification.

When a notification is completed, it receives the status *NOCO* (*notification completed*).



You cannot complete a notification if outstanding tasks still exist in the notification.

Categorizing a notification problem

When you complete a notification, the system displays a dialog box with the current date and time. You can accept the displayed date and time as the completion time or you can change it. In this dialog box, you must specify whether the problem arose because of an internal or external cause. For example, if a vendor delivers a defective product, you are dealing with an externally caused problem.

Put notification in process again

You can put a completed notification in process again, if you determine that additional processing is necessary.

Archive completed notification

Process Notification

Completed notifications are automatically archived and are deleted from the database after a specific, predefined time interval. This time interval is defined in Customizing. Therefore, make sure you have finished processing the notification before you complete it.

Delete notification

In some cases, you may want to delete a notification. You cannot, however, delete a notification directly. You must set a deletion flag in the notification which identifies it as one that can be deleted.

A company typically runs an archiving program at regular intervals. This program searches for the notifications with deletion flags, converts the deletion flags to deletion indicators, and copies the notifications to an archive. The system then deletes the notifications from the database.



Before you delete a notification, make sure that it is no longer needed.

Once you set a deletion flag for a notification, you can no longer change the data in the notification. It then receives the status *Deletion flag* and *Completed*. You can now only display the notification. For more information about notification statuses, see [Status Management \[Page 118\]](#).

If you set a deletion flag in the notification header, the system also sets deletion flags for the individual notification tasks.

You can cancel the deletion flag as long as it has not been converted to a deletion indicator. You can then put the notification back in process and change or display it in the usual manner.

Procedure

Desired processing type	What you should know
Put notification in process	Call up the notification in the create or change mode and then choose <i>Notification</i> → <i>Functions</i> → <i>Put in process</i> .
Postpone notification	Call up the notification in the change mode, choose <i>Notification</i> → <i>Functions</i> → <i>Postpone</i> , and then save the notification.
Complete notification	Call up a notification in the change mode and choose <i>Notification</i> → <i>Functions</i> → <i>Complete</i> . Specify the completion date/time and the cause of the problem; then close the dialog box and save the notification.
Delete notification	To set a deletion flag, call up the notification in change mode and then choose <i>Notification</i> → <i>Functions</i> → <i>Deletion flag</i> → <i>Set</i> . To reset the deletion flag, choose <i>Notification</i> → <i>Functions</i> → <i>Deletion flag</i> → <i>Deselect</i> .

Processing Notifications Using Worklist

Use

Using the worklist for notifications, you can select notifications on the basis of different criteria and process them. You can select the notifications as follows:

- You select the notifications using the *My worklist* function. The system then displays the following notifications for further selection and processing:
 - All notifications to be processed by you as the "person responsible" or that fall within your department's area of responsibility
 - All notifications to be processed by you as a *<Partner function>*
 - All notifications you created
 - All notifications that were last processed by you
- You select the notifications using the *General selection* function. The system then displays all notifications for further selection and processing. Make sure the preset selection criteria contain the values you want (for example, notification status, selection period).

Prerequisites

If you use the *General selection* function and you want to process notifications for which you are not responsible or which you did not create, you must have the corresponding authorization.

Activities

You choose *Logistics* → *Central Functions* → *Notification* → *Worklist* → *Notifications* to call up the worklist for notifications.

You enter the required data and choose *Execute*.

See also:

[Select and Process Notifications Using Worklist \[Page 100\]](#)

Select and Process Notifications Using Worklist

Select and Process Notifications Using Worklist

Use

You want to use the worklist for notifications to select and display a list of notifications. For example:

- You want to display or change a specific notification, but you do not know the number of the notification.
- You want to display or change several notifications with certain attributes.
- You want to perform certain functions simultaneously for several notifications with particular attributes.
- You want to select the notifications that you are responsible for processing.

Prerequisites

If you want to select notifications on the basis of partner functions:

- Partner functions must be defined in Customizing
- Users or departments must be assigned to the partner functions

Procedure

1. Choose *Logistics* → *Central Functions* → *Notification* → *Worklist* → *Notifications*.

The dialog box *Worklist: Notifications* appears.

2. If you want to create your own **personal** worklist, choose the *My worklist* tab page.

If you want to create a list of **all** notifications that meet certain selection criteria, choose the *General selection* tab page.

3. Enter the desired selection criteria.

If you want to	Then
Select only the notifications that are assigned to you for processing	Set the <i>For processing by me</i> indicator on the <i>My worklist</i> tab page.
Select only the notifications that were created by you	Set the <i>Created by me</i> indicator on the <i>My worklist</i> tab page.
Select only the notifications that were last processed by you	Set the <i>Last processed by me</i> indicator on the <i>My worklist</i> tab page.
Select only the notifications assigned to you as a specific partner (for example, in your capacity as an "author")	Set the <i>In my role as</i> indicator on the on the <i>My worklist</i> tab page and choose a partner function in the list field.

Select and Process Notifications Using Worklist

<p>Select notifications that were created within a specific time period</p>	<p>Set the indicator:</p> <ul style="list-style-type: none"> • <i>1 week</i>, if you want to select notifications that were created up to one week before the current date. • <i>1 month</i>, if you want to select notifications that were created up to one month before the current date. • <i>3 months</i>, if you want to select notifications that were created up to three months before the current date. • <i>1 year</i>, if you want to select notifications that were created up to one year before the current date. • <i>Defined by me</i> and enter the desired time interval if you want to select notifications that were created within a specific time period.
<p>Select a specific notification or several notifications of a specific notification type</p>	<p>Enter the desired notification number or notification type.</p>
<p>Format the list display to meet your own specific requirements</p>	<p>Choose the desired layout in the <i>Layout</i> list field.</p>  <p>You can store the following information in a layout:</p> <ul style="list-style-type: none"> • List column structure • Sort criteria • Filter conditions

Select and Process Notifications Using Worklist

Specify which criterion the monitor column in the list of notifications references	<p>In the list field <i>Reference field for monitor</i>, choose a reference field. The monitor column is then displayed with the corresponding status.</p>  <p>If you do not choose a reference field, the monitor column is not displayed.</p>  <p>For example, if you chose the desired start/end date as a reference field, the colors in the monitor column have the following meaning:</p> <ul style="list-style-type: none"> • Green: the desired start date has not been reached • Yellow: the desired start date has been reached, but the desired end date has not been reached • Red: the desired end date has been surpassed. <p>The notifications displayed in red are critical and should be processed with the highest priority.</p>
Select notifications with a specific status	Select the corresponding indicator on the <i>General selection</i> tab page or enter the desired selection profile.
Select notifications that have been assigned specific partners	In the <i>Partner</i> area of the <i>General selection</i> tab page, choose the desired partner function(s) and/or enter the partner number.



If you always want to use the same selection criteria in your worklist, enter the desired data and choose *My defaults*.

4. Choose *Execute*.

The system displays a list of notifications that meet your selection criteria.

5. You can use the functions described below to process selected objects in the list further.

Function	Procedure
Switch between the <i>Display notification</i> and <i>Change notification</i> transactions	Choose the <i>Change <-> Display</i> pushbutton.
Display a graphic (for example, how many notifications were created on a particular date or for a specific material)	Position the cursor on a column header text, select the desired objects in the list, and then choose <i>Graphic</i> .
Export notification list to MS Excel	Choose <i>Spreadsheet</i> .

Select and Process Notifications Using Worklist

Display document flow for one or more notifications	Select one or more objects in the list and then choose <i>Display document flow</i> .
Initiate an outgoing telephone call (<i>SAPphone</i> component must be active)	Select a notification and then choose <i>Telephone call</i> .
Display notification long text	Select a notification and then choose <i>Long text</i> .
Display or change notifications (branch to notification header)	Select one or more objects in the list and then choose <i>Notification</i> .
Display a portfolio	Position the cursor on a column header and then choose <i>List → Portfolio</i> .
Compare notification processing dates (requested start and end dates)	Select one or more objects in the list and then choose <i>Goto → Schedule graphic</i> .
Display report documentation	Choose <i>Goto → Report documentation</i>
Print one or more notifications	Select one or more objects in the list and then choose <i>Notification → Print notification</i> .



If you selected more than one object in the list, the system displays the first selected object in the list (in the display or change mode). When you have finished processing the first object, choose *Goto → Back*. The next selected object is automatically displayed.

Processing Notifications Using Workflow

Processing Notifications Using Workflow

Use

If a notification is created or put in process in the system, the system can automatically notify the person(s) or department(s) responsible via the *SAP Business Workflow* that:

- The notification must be processed or tasks need to be defined
- Defined tasks need to be executed
- All notification tasks have been completed and the notification must either be completed or additional tasks must be defined for the notification

You receive the information about the notification that needs to be processed as a *work item* in your inbox (*Business Workplace*). Your inbox provides an overview of all notifications that you are responsible for processing. You can begin processing the work item directly from your business workplace.

Integration

You can access your business workplace by choosing *Office* → *Workplace* from the SAP menu. Then choose *Inbox* → *Workflow* to display any work items that may be present.

Prerequisites

The required system settings to activate the *SAP Business Workflow* have been made.

Activities

If there is a work item in your inbox, proceed as follows to begin processing the work item:

- You double click the work item to display a description of the notification.
- You choose *Execute* to process the work item. The system calls up the transaction to process a notification.

Partners in Notification Processing

Definition

A **business partner** is a business object that consists of a person, organization, group of persons, or group of organizations in which a company has a business interest. A business partner may also be a person, organization, or group within a company.

A business partner can have several **functions** at one time. Functions are characterized by function categories. A function category, for example, may be a sold-to party, interested person, payer, or payee.

A business partner can have relationships with other business partners. Relationships are characterized by **relationship types**. For example, relationship types can include the following: "is a subsidiary of," "is the contact person for," or "is identical to." A business partner can have several addresses. Addresses are defined by address types. An address type, for example, can consist of a delivery address or correspondence address.

A business partner can be assigned other business partners using the function determination. These business partners can then act on behalf of the original partner in their respective roles. For example, a business partner can be defined as a payee for another business partner.

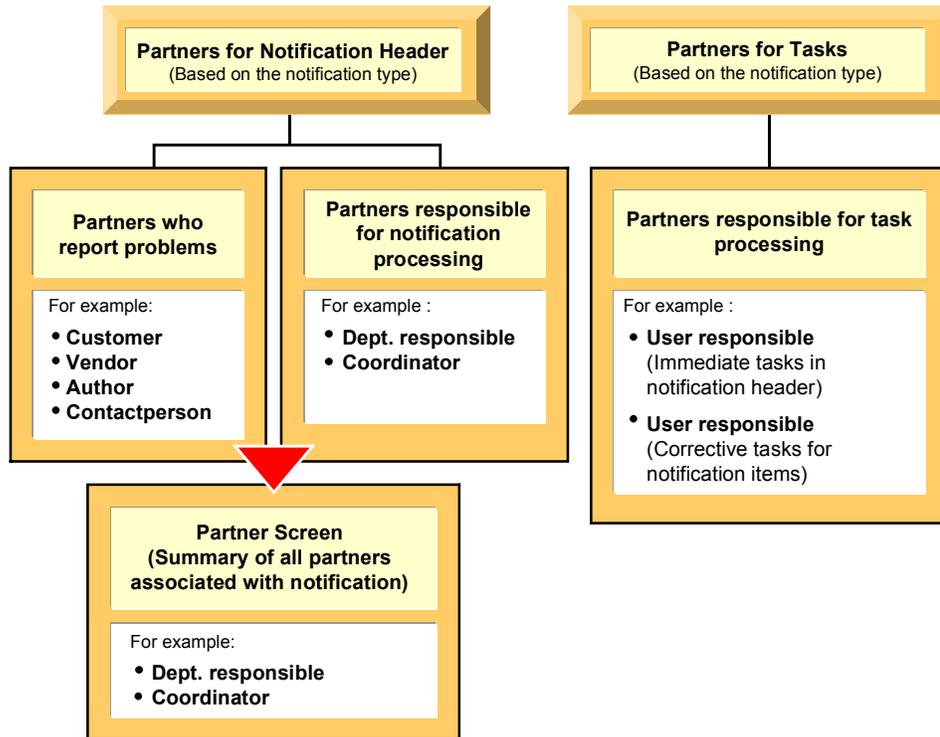
For more information about business partners, refer to the documentation *Basic Functions and Master Data in Sales and Distribution*.

Use

In a notification, you can record all partners who are involved with subject matter/problem described in the notification. This includes the person who reported the subject matter/problem, the persons responsible for processing the notification, and those responsible for processing individual notification tasks. The partner data that you can maintain in a notification is predefined in Customizing for each notification type.

The following graphic shows how partners are used for notification processing.

Partners in Notification Processing



Structure

Partner type

Partners can be:

- Internal partners (for example, internal employees or partners in the *Human Resources* component)
- External partners (for example, customers, vendors, or manufacturers)

Partner function

The various partner types can be subdivided into **partner functions**. The partner function defines the rights, responsibilities and tasks of each partner when a business transaction is processed.

Partner determination procedure

The partner determination procedure specifies whether partners can be assigned to a notification and which partner functions can be used to process various business transactions. Your system administrator defines the partner determination procedure in Customizing.

Processing the Partner Information

Prerequisites

You can only display the partner data screen if a partner determination procedure has been defined in Customizing for *Cross-Application Components* under *Notification → Notification Creation → Partner*.

Procedure

1. In a notification, choose *Goto → Partner*.

The screen for the partner data appears. If partner data already exists for the notification, it will be displayed in a table. In this screen, you can use the following functions:

Function	Follow-on menu path
Display partner data	Select a partner and choose <i>Environment → Display object</i> .
Delete a partner	Select a partner and choose <i>Delete partner assignment</i> .
Phone a partner	Select a partner and choose <i>Phone partner</i> .
Contact partner via paging function	Select a partner and choose <i>Send message to partner</i> .
Sort partners by function	Choose <i>Edit → Sort by partner function</i> .
Display, change, or delete partner addresses	Select a partner and choose the corresponding pushbutton

If you want to enter new partner data for the notification, enter the data directly in the table.

2. Go back to the notification header and save the new or changed partner data.

Task Processing

Task Processing

Purpose

Using the functions for task processing, you can change or display the [tasks \[Page 30\]](#) in a notification independently of the notification itself. In this way, you can:

- Process tasks quickly and easily
- Execute tasks using the functions of the action box
- Access the information in the associated notification at any time

Process Flow

1. You process the tasks using one of the following options, depending on your job duties or method of working:
 - [Worklist \[Page 111\]](#)
You call up the worklist for tasks to determine whether tasks exist that you must process.
 - [Workflow \[Page 116\]](#)
You check your workflow inbox determine whether tasks exist that you must process.
 - [Functions for Task Processing \[Page 109\]](#)
If you want to process a specific task in a specific notification, use the transactions for processing or displaying a task.
2. After you have selected and executed one of the above options, you process a task by either:
 - Forwarding the task to another person, if you cannot process the task yourself
 - Carrying out the instructions specified in the task
 - Documenting the information for a task that has just been executed; for example, as a task long text or via a follow-up function in the action box (internal remark)
 - Setting the status of the task to "completed," once you have carried out the instructions specified in the task
 - Executing additional follow-up functions in the action box, if additional tasks need to be implemented

Result

Once you have processed and completed a task, you can complete the notification (provided there are no other tasks in the notification that need to be processed).

Process Specific Task in a Notification

Use

Using this function, you can process a specific task in a notification. You can process tasks for the notification header (immediate tasks) as well as tasks for individual items (corrective tasks). On the initial screen for this transaction, for example, you can:

- Call up the task to be processed directly by entering the notification number, item number, and task number
- Display a selection of tasks using the input help for the notification number and then select a specific task (for example, if you do not know the notification number, item number, and task number)

Procedure

Notification, item, and task numbers are known

1. Choose *Logistics* → *Central Functions* → *Notification* → *Task* → *Process*.

The initial screen for processing a task is displayed.

2. Enter the notification, item, and task numbers and choose *Enter*.

The screen for processing a task is displayed.

3. Use the functions listed in the table below to process the task.

Notification, item, and task numbers are unknown

1. Choose *Logistics* → *Central Functions* → *Notification* → *Task* → *Process*.

The initial screen for processing a task is displayed.

2. Choose the input help for the *Notification* field and enter your selection criteria for the task in the dialog box.

For example, you can select a task on the basis of the notification type, the person responsible, or various other views (for example, vendor, material, WBS element/project, etc.). If you do not enter any selection criteria, the system will select all tasks in the system.

3. Choose *Continue*.

The system displays a list of tasks sorted by notification and item numbers.



Tasks with an item number "0000" are immediate tasks. Tasks with an item number greater than "0000" are corrective tasks.

4. Select a task and then choose *Copy*.

The initial screen reappears.

5. Choose *Enter*.

Process Specific Task in a Notification

The screen for processing a task is displayed. Use the functions listed in the table below to process the task.

Functions on the Task Screen		
Function	Menu path/Pushbutton	What you should know
Release task	<i>Release task</i> pushbutton	You release the task so it can be executed. For more information, see Status Management for Tasks [Page 123] .
Complete task	<i>Complete task</i> pushbutton	You confirm that the task was executed. For more information, see Status Management for Tasks [Page 123] .
Denote task as being successful	<i>Task successful</i> pushbutton	You confirm that the task solved the problem successfully. For more information, see Status Management for Tasks [Page 123] .
Set user status	<i>Set status</i> pushbutton	If a user status was defined in a status profile, you can use this function to set the user status. For more information, see Status Management for Notifications [Page 118] .
Display associated notification	<i>Notification (extended)</i> pushbutton	You display the notification that contains the task. Choose <i>Back</i> to return to the task processing screen.
Display object	<i>Display object</i> pushbutton	You display reference objects or reference documents that may be specified in the notification.
Display action log	<i>Action log</i> pushbutton	You display changes in the notification via the action log [Ext.] .
Expand all screen areas	<i>Expand all (Expand all screen areas)</i> pushbutton	You expand all screen areas in the task processing screen.
Collapse all screen areas	<i>Collapse all (Collapse all screen areas)</i> pushbutton	You collapse all screen areas in the task processing screen.
Display notification	<i>Notifctn (Open)</i> pushbutton	You display a summary of the notification data, including short and long texts.
Expanding the individual screen areas	<i>Basic data, Planning and Item</i> pushbuttons	You expand the individual screen areas to display and process the corresponding task data.
Action box		You execute follow-up functions in the action box. For more information, see Action Box [Page 33] .

Processing Tasks Using Worklist

Use

Using the worklist for tasks, you can select notification tasks on the basis of different criteria and process them. You can select the tasks as follows:

- You select the tasks using the *My worklist* function. The system then displays the following tasks for further selection and processing:
 - All outstanding tasks to be processed by you as the "person responsible"
 - All tasks to be processed by you as a *<Partner function>*
 - All tasks you created
 - All tasks you completed
- You select the tasks using the *General selection* function. The system then displays all tasks for further selection and processing. Make sure the preset selection criteria contain the values you want (for example, selection period, outstanding tasks only).

Prerequisites

If you use the *General selection* function and you want to process tasks for which you are not responsible or which you did not create, you must have the corresponding authorization.

Activities

You choose *Logistics* → *Central functions* → *Notification* → *Worklist* → *Tasks* to call up the worklist for tasks.

You enter the required data and choose *Execute*.

See also:

[Select and Process Tasks Using the Worklist \[Page 112\]](#)

Select and Process Tasks Using Worklist

Select and Process Tasks Using Worklist

Use

You can use the worklist for tasks to select and display a list of tasks. For example:

- You want to obtain an overview of all notification tasks that fulfill certain criteria.
- You want to select the tasks that you are responsible for processing.

Prerequisites

If you want to select notifications on the basis of partner functions:

- im Customizing Partnerrollen definiert sein
- Users or departments must be assigned to the partner functions

Procedure

2. Choose *Logistics* → *Central Functions* → *Notification* → *Worklist* → *Tasks*.

The dialog box *Worklist: Tasks* appears.

3. If you want to create your own **personal** worklist, choose the *My worklist* tab page.

If you want to create a list of **all** tasks that meet certain selection criteria, choose the *General selection* tab page.

4. Enter the desired selection criteria.

If you want to	Then
Select only the tasks that are assigned to you for processing	Set the <i>For processing by me</i> indicator on the <i>My worklist</i> tab page.
Select only the tasks that were created by you	Set the <i>Created by me</i> indicator on the <i>My worklist</i> tab page.
Select only the tasks that were completed by you	Set the <i>Completed by me</i> indicator on the <i>My worklist</i> tab page.
Select only the tasks assigned to you as a specific partner (for example, in your capacity as an "author")	Set the <i>In my role as</i> indicator on the on the <i>My worklist</i> tab page and choose a partner function in the list field.

Select and Process Tasks Using Worklist

<p>Select tasks for notifications that were created within a specific time period</p>	<p>Set the indicator:</p> <ul style="list-style-type: none"> • <i>1 week</i>, if you want to select notifications that were created up to one week before the current date. • <i>1 month</i>, if you want to select notifications that were created up to one month before the current date. • <i>3 months</i>, if you want to select notifications that were created up to three months before the current date. • <i>1 year</i>, if you want to select notifications that were created up to one year before the current date. • <i>Defined by me</i> and enter the desired time interval if you want to select notifications that were created within a specific time period.
<p>Select tasks on the basis of a specific task code group, task code, or a selection profile</p>	<p>Enter the desired code group, task code, or selection profile.</p>
<p>Select tasks for specific notifications</p>	<p>Enter the desired notification number(s) or notification type(s).</p>
<p>Format the list display to meet your own specific requirements</p>	<p>Choose the desired layout in the <i>Layout</i> list field.</p>  <p>You can store the following information in a layout:</p> <ul style="list-style-type: none"> • List column structure • Sort criteria • Filter conditions

Select and Process Tasks Using Worklist

Specify which criterion the monitor column in the list of tasks references	<p>In the list field <i>Reference field for monitor</i>, choose a reference field. The monitor column is then displayed with the corresponding status.</p>  <p>If you do not choose a reference field, the monitor column is not displayed.</p>  <p>For example, if you chose the desired start/end date as a reference field, the colors in the monitor column have the following meaning:</p> <ul style="list-style-type: none"> • Green: the desired start date has not been reached • Yellow: the desired start date has been reached, but the desired end date has not been reached • Red: the desired end date has been surpassed.
Select only the outstanding tasks	Select the corresponding indicator on the <i>General selection</i> tab page.
Select tasks that have been assigned specific partners	Select the corresponding partner function in the list field on the <i>General selection</i> tab page.
Select notifications that have been assigned specific partners	In the <i>Partner</i> area of the <i>General selection</i> tab page, choose the desired partner function(s) and/or enter the partner number.



If you always want to use the same selection criteria in your worklist, enter the desired data and choose *My defaults*.

5. Choose *Execute*.

The system displays a list of notifications that meet your selection criteria.

6. You can use the functions described below to process selected objects in the list further.

Function	Procedure
Switch between the <i>Display notification</i> and <i>Change notification</i> transactions, as well as the <i>Display task</i> and <i>Change task</i> transactions	Choose the <i>Display <-> Change</i> pushbutton.
Export list of tasks to MS Excel	Choose <i>Spreadsheet</i> .
Display graphic	Position the cursor on a column header text, select the desired objects in the list, and then choose <i>Graphic</i> .

Select and Process Tasks Using Worklist

Initiate an outgoing telephone call (<i>SAPphone</i> component must be active)	Select a task and then choose <i>Telephone call</i> .
Display long text	Select a task and then choose <i>Long text</i> .
Display or change notifications (branch to notification header)	Select one or more objects in the list and then choose <i>Notification</i> .
Display or change tasks (branch to notification header)	Select one or more objects in the list and then choose <i>Task</i> .
Display a portfolio	Position the cursor on a column header and then choose <i>List → Portfolio</i> .
Compare task processing times (requested start and end times)	Select one or more objects in the list and then choose <i>Goto → Schedule graphic</i> .
Display report documentation	Choose <i>Goto → Report documentation for...</i>
Print one or more notifications	Select one or more objects in the list and then choose <i>Notification → Print notification</i> .



If you selected more than one object in the list, the system displays the first selected object in the list (in the display or change mode). When you have finished processing the first object, choose *Goto → Back*. The next selected object is automatically displayed.

Processing Tasks Using Workflow

Processing Tasks Using Workflow

Use

If a task is created in a notification or if the person responsible for a task is changed, the system can automatically notify the person or department responsible via the *SAP Business Workflow* that a task needs to be processed.

You receive the information about the task that needs to be processed as a *work item* in your inbox (*Business Workplace*). Your inbox provides an overview of all tasks that you are responsible for processing. You can begin processing the work item directly from your business workplace.

Integration

You can access your business workplace by choosing *Office* → *Workplace* from the SAP menu. Then choose *Inbox* → *Workflow* to display any work items that may be present.

Prerequisites

The required system settings to activate the *SAP Business Workflow* have been made.

Activities

If there is a work item in your inbox, proceed as follows to begin processing the work item:

- You double click the work item to display a description of the task to be executed.
- You choose *Execute* to process the work item. The system calls up the transaction to process the task.

Status Query

During the whole runtime of an [internal service request \[Ext.\]](#), you can obtain the current status.

See also:

- [Status Management For Notifications \[Page 118\]](#)
- [Display Status Information \[Page 119\]](#)
- [Assigning System Statuses \[Page 121\]](#)
- [Assigning And Changing User Statuses \[Page 122\]](#)
- [Status Management For Tasks \[Page 123\]](#)

Status Management for Notifications

Use

Notifications are supported by the SAP System's status management function. Whenever you create and process a notification, you execute certain business transactions for the notification. These business transactions are documented in the SAP System by a status. The current status of a notification tells you which processing stage the notification has reached and which steps have yet to be completed.

Of the available statuses in the system, only a few need to be set at any given time. Usually, a notification must have a certain status before you can process the it further. If such a status is missing, the system displays a message to inform you of this.

Features

A notification can have the following types of statuses:

- **System statuses**

System statuses are set when you execute certain functions for a notification. For example, when you print a notification, the system sets the status "printed" (*NOPT*). The system status tells you that a business transaction was carried out for the notification.

The system statuses for notifications are predefined in the SAP System and cannot be changed.

- **User statuses**

User statuses allow you to further restrict the functions that are allowed by a system status.

Your system administrator defines user statuses in a status profile (see Customizing for *Cross-Application Components* under *Notification Processing* → *Status Management* → *Define Status Profile*). You can assign and delete user statuses if you have the necessary authorization.

See also:

[Assigning and Changing User Statuses \[Page 122\]](#)

Displaying Status Information

Procedure

To use the functions in the following table, call up the notification in the create or change mode.

Function	Tab page / Pushbutton / Menu path	What you should know
Display status information in the notification header		<p>The current system statuses are displayed in the <i>Status</i> field of the notification header. If user statuses also exist for the notification, they are also displayed.</p> <p>The <i>Status</i> field also displays the task status. If at least one notification task is still outstanding or released, the status "task outstanding" (TSOS) is displayed. As long as this status is set, you cannot complete the notification.</p>
Display status information for individual tasks	<i>Tasks</i> tab page	The task overview contains the status of all recorded tasks.
Display system and user statuses with short texts	<i>Status detail</i> pushbutton	All active system and user statuses for the notification are displayed on the <i>Status</i> tab page.
Display general status overview	Choose <i>Status detail</i> pushbutton and then <i>Extras</i> → <i>Overview</i>	<p>You can see which processing steps have already been performed.</p> <p>All active and inactive system and user statuses for the notification are displayed.</p>
Display statuses that can still be assigned	Choose <i>Status detail</i> and then the <i>Business processes</i> tab page	<p>You can see which processing steps are currently allowed or not allowed.</p> <p>If you choose <i>Transaction analysis</i>, you can see which business transactions can still be executed for the notification (green traffic light) and which ones can no longer be assigned (red traffic light).</p>

Displaying Status Information

Assignment of System Statuses

Use

Whenever you execute a function in a notification that is influenced by the SAP System's status management function, the system automatically assigns a corresponding system status to the notification. On the basis of the current system status, you can determine what other functions can be executed for the notification.

Features

The system assigns a system status to a notification when you:

- Create a notification
- Postpone a notification
- Put a notification in process
- Assign an order to a notification
- Complete a notification
- Print a notification
- Create a task
- Release a task
- Complete a task
- Complete a task successfully
- Complete all tasks
- Activate an approval requirement for a notification
- Give an approval to a notification
- Refuse an approval for a notification
- Create a costing for a claim

Assigning and Changing User Statuses

Assigning and Changing User Statuses

Prerequisites

To be able to assign user statuses to a notification, the following conditions must be fulfilled in Customizing for *Cross-Application Components* under *Notification Processing* → *Status Management* → *Define Status Profile*:

- User statuses are defined in a status profile
- A status profile is assigned to the notification type to be processed.

You can only change a user status if the **status order numbers** assigned to the user status in Customizing allow it:

- The status order numbers defined for individual user statuses in the status profile determine the sequence in which the user statuses can be activated.
- If a user status does not have an order number assigned to it, the user status can always be set.
- It is not possible for several statuses with status order numbers to be active at the same time.
- You cannot delete the initial user status that has been set for a notification.

Procedure

1. Call up a notification in the change mode.
2. In the notification header, choose *Status detail*.
The status information is displayed.
3. In the column for user statuses (right column), activate the desired user status and then choose *Back*.
The notification header with the chosen user status is displayed.
4. Save the notification.

Status Management for Tasks

Use

A task status provides information about a task's current processing stage. The system manages the following statuses for tasks:

Status	Meaning
Task outstanding	The task was created in the notification.
Task released	The task can be executed.
Task completed	The task was executed.
Task successful	The result of the task was successful (for example, the problem was solved or damage was repaired).

The system does not distinguish between notification header tasks (immediate tasks) and item tasks (corrective tasks). If the notification status information indicates that outstanding tasks exist in the notification, these tasks can consist of both immediate tasks and corrective tasks.



A notification retains the status "outstanding tasks" as long as a task has not been completed (even if the task has already been released).

Integration

The various notification and task statuses, for the most part, are independent of one another. This means, for example, that you can put a notification in process even though it contains outstanding, unreleased tasks. As long as a notification still contains outstanding tasks, however, you cannot complete the notification.

For more information about the general status management function, see [Status Management \[Ext.\]](#).

Features

You can use the status management function as follows:

- **Assign a status to a task**

You assign a status to a task by choosing the corresponding function in the task overview or task detail screens. You must assign a status to each task individually. If you select several tasks in the overview screen and choose a status, the system assigns the status to all of the selected tasks.
- **Status information**

Once you have assigned a status to a task, the corresponding status is displayed in the task overview screen and task detail screen. If you want to display the text for the status abbreviations, choose *Status detail*.
- **Date changes as a result of status assignments**

When you assign a status to a task, the system automatically changes the relevant dates as follows:

Status Management for Tasks

Status	The system fills the following fields:
Released	<i>Planned start</i> with the current date/time <i>Planned end</i> with either the current date/time or desired notification end date/time. You can overwrite both fields.
Completed	<i>Date completed <date> by <user name></i> with the current date/time and current user name. You cannot overwrite these fields.
Successful	None



The planned dates for a task must fall within the desired start and end dates of the notification. If you specify other dates for the task, the system displays a message to inform you of this.

Document Flow

Use

You can use this function to display the document flow for a notification as a list or network graphic. The document flow allows you to identify:

- Preceding documents (documents from which the current notification originated)
- Subsequent documents (documents that originated from the current notification)
- "To-and-from" references between documents

Each document contains the following information:

- Document number
- Document number
- Additional data (if available)
- Logical system (if any of the other documents was created in a logical system other than the one in which the original notification was created)

You can also display the detailed information for each document.

Integration

The document flow is integrated in the following processes:

- *Materials Management (MM)*
- *Production Planning and Control (PP)*
- *Sales and Distribution (SD)*
- *Customer Service (CS)*
- *Plant Maintenance (PM)*
- *Financials (FI)*
- *Controlling (CO)*
- *Project System (PS)*

Features

The document flow, for example, can display the following objects (among others):

- Quality notification
- Service notification
- Maintenance notification
- General notification
- Claim
- Service order

Document Flow

- Sales order
- Production order
- Run schedule header
- Goods movement
- Purchase order
- Inspection lot

Displaying the Document Flow for a Notification

Procedure

1. Call up a notification in the create or change mode.
2. Choose one of the following menu paths, depending on whether you want to display the document flow as a list or graphic:
 - *Extras* → *Notification documents* → *Document flow* → *List*
 - *Extras* → *Notification documents* → *Document flow* → *Graphic*



The notifications that reference other subsequent documents in the list display are labeled with a (*Ref.*) designation.

3. If you displayed the document flow as a graphic, select a document and choose *Goto* → *Display document*.

The detailed information for the selected document is displayed.

Example: Entering And Processing An Internal Service Request**Example: Entering And Processing An Internal Service Request**

To make an application for a removal in the intranet, you choose the scenario *Removal request*. In Customizing this request category is specified as requiring approval. Therefore once you send this notification, the approval process is triggered. When the removal is carried out, costs are posted to the requesting cost center.

The Environment

Intranet	SAP System
Person who makes the entry	Person who processes/approves the request

The Removal Request Form

The removal request in the intranet is structured as follows:

The title contains the request name, in this case *Removal Request*, and the company logo. Underneath the request creation function is the request number.

Under this, on the right-hand side is the *Entry data*, such as *Name* and *Telephone number* of the person who makes the entry, and the *Date entered*.

On the left-hand side is the *Removal data* with the following fields:

- *Who is moving? (initiated by)*
- *From building*
- *From room*
- *To building*
- *To room*
- *Removal date*
- *Number of PCs to be moved*
- *Change telephone?*
- *Number of packing boxes required*

Underneath this is a free text field called *Remarks*.

Next to the fields for the building or room, you may have included a pushbutton when you defined the request form. This pushbutton takes you to the building plan.

Creating The Removal Request

You choose *Internal Service Request* in the launchpad for your work place.

Example: Entering And Processing An Internal Service Request

You choose the scenario *Removal request*.
The entry form for the removal request is displayed.

The fields for the person entering data are derived from the user data, and cannot be changed.

Example: Entering And Processing An Internal Service Request

As a standard setting the field *Initiated by* is defaulted with the name of the person making the entries, and can be changed.

The fields for the removal details are mandatory.

Once you have entered your data, you can choose from the following two options:

- *Check request*
- *Send request*

After you send the request, you receive a success message containing the notification number.

After The Removal Request Has Been Sent

The fields *Who is moving?* and *Removal date* are checked according to the entry format.

At the same time, the *Requesting controlling area* and *Requesting cost center* are derived from the initiator data.

A notification is created in the SAP R/3 System for the intranet request. The fields in the HTML form are transferred to the notification.

Three tasks are created for the notification.

Task	Partner Role	Type of Item	Costing Item
Move PC	Task processor	Activity item	<p>If the number of PCs is greater than zero, an activity item is created with the:</p> <ul style="list-style-type: none"> • <i>Sending cost center</i> = PC cost center • <i>Activity type</i> = PCMOVE • <i>Receiver</i> = Requesting cost center • <i>Quantity</i> = Number of PCs <p>The requesting cost center and the quantity are transferred from the form.</p> <p>The service provider (the cost center and activity type) is stored in the Customizing (see below).</p>

Example: Entering And Processing An Internal Service Request

Change telephone	Task processor	Activity item	<p>If the number of telephones is greater than zero, an activity item is created with the:</p> <ul style="list-style-type: none"> • <i>Sending cost center</i> = Telephone cost center • <i>Activity type</i> = PHONEMOVE • <i>Receiver</i> = Requesting cost center • <i>Quantity</i> = Fixed quantity one <p>The requesting cost center and the quantity are transferred from the form. The service provider (the cost center and activity type) is stored in the Customizing (see below).</p>
Move work place	Task processor	External activity	The <i>removal</i> service is provided by an external company.

The notification is automatically costed using three costing items.

The name of the person responsible for the notification is the cost center manager of the employee moving (this is a standard setting).

The notification is forwarded to the manager for approval.

Customizing Required For Automatically Transferring The Removal Request

You need to store certain data in Customizing to allow the system to forward the removal request to an SAP R/3 notification with notification items.

- You specified whether the request requires approval, and who is to approve it for the notification type *Removal request*.
- For each field from which a notification item is to be generated (for example, in this case, the *number of PCs*), the following was stored in Customizing independently of the requesting cost center:

Type of Item	Stored Data
Activity item	<ul style="list-style-type: none"> • Sending cost center • Activity type • Quantity
Material item	<ul style="list-style-type: none"> • Material number • Storage location • Plant • Quantity

Example: Entering And Processing An Internal Service Request

External activity	<ul style="list-style-type: none">• Material number• Vendor• Purchasing organization• Quantity
-------------------	---

- These items are assigned to tasks.

The system displays the basic data for the request to the processors for the notification and task in the *Extended view* in the processing type.

Status Query By The Person Who Made The Request

If you make, enter, or initiate a request, you can check its status in the intranet. The system displays (for example) the notification short text, who it was entered by, and on what date. You can see the notification priority and number for example, by using the worklist.

You can go from here to the notifications detail screen, or add a note.

In the *detail screen* for each notification, you can see the request form, which cannot be changed.

In addition, the following fields are displayed underneath:

- *Status*
- *Approved by*

Under this, there is a free text where you can enter more remarks.

Approval Of The Removal Request

You (the approving manager) also use the intranet. You receive a list of all service requests (regardless of request type) for approval.

The list contains the following fields:

- *Description of notification type, such as removal request*
- *Entered by*
- *Initiated by*
- *Date entered*
- *Required date*
- *Status*

You can go from here to the detail screen for a notification or to approve a notification.

In the *detail screen* for each notification you can see the request form, which cannot be changed.

Example: Entering And Processing An Internal Service Request

The screenshot shows a SAP form for entering and processing an internal service request. The form is contained within a blue border and is divided into several sections. At the top, there are two input fields: 'Initiated by' on the left and 'Entered by' on the right. Below these are two more input fields: 'Request data' on the left and 'Costs' on the right. At the bottom of the form is a large text area labeled 'Remarks'. Below the form are three buttons: 'Check entry', 'Reject request', and 'Approve request'.

In addition, the following fields are displayed underneath:

- *Debited cost center*
- *Estimated costs*

Underneath this there is a free text in which you can store more remarks, for example, if you reject the request.

Once you have entered your data, you can choose from the following options:

- *Reject request*
- *Approve request*

After you approve the service request, the system sets the status to *Approved*. Each item is forward to the processor who is determined using the settings in Customizing.

Carrying Out The Removal

Each task has a different processor.

You (the processor) receive a list of all the tasks that are to be processed.

The list contains the following fields:

- *Service request number*
- *Required date*
- *Initiated by*
- *Date entered*
- As well as more data from the form

You can go from here to the detail screen for a task, or complete a task.

Example: Entering And Processing An Internal Service Request

In the *detail screen* for each notification you can see the request form, which cannot be changed. Underneath this there is also a free text in which you can store more remarks, such as when a material no longer exists and is replaced by another.

Once you have entered your data, you can choose from the following two options:

- *Forward*
Forwarding is useful if you are not responsible for this notification. You can enter the processor who is responsible.
- *Complete*
There are two different situations for completing a notification that you can choose in Customizing for the service request type:
 - *Non-Integrated R/3 Posting*
You complete your work in and outside of the SAP System (for example, you carry boxes and post the activity allocation manually in the SAP System using confirmation). In this case, only *Completed* is available. This only closes the task in R/3.
 - *Integrated R/3 Posting*
You complete your work outside of the SAP System and want to make an integrated posting of the R/3 data upon completing the service request. In this case, you can use the action box for *Execution-Services*. A simple screen appears for the current application (such as activity entry, or for material issue) in which the relevant fields are defaulted with values from the plan data for the notification, and which you can change. You make your posting to the account assignment object stored in Customizing. The postings are integrated in R/3. The system prevents accidental double postings by issuing a warning, and preventing you from going to the document flow.

If all the notification tasks are completed, it receives the *Completed* status.

Cost Analysis By The Approver

You (the approver) can analyze the costs of the service request using a report for your cost center.

You go from the totals records report to the line item report. In the line items, the system displays the service request number if the line item was posted integrally from a service request. You can go from the line items of a service request to the service request itself.