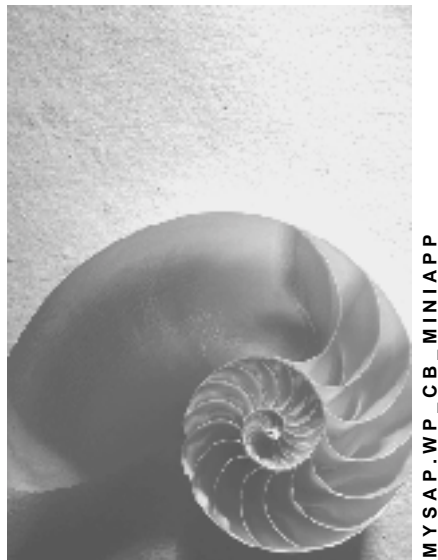


MiniApps



Release 2.11



Copyright

© Copyright 2000 SAP AG. All rights reserved.

No part of this brochure 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® 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, California, USA.

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, Laboratory for Computer Science NE43-358, Massachusetts Institute of Technology, 545 Technology Square, Cambridge, MA 02139.

JAVA® is a registered trademark of Sun Microsystems, Inc. , 901 San Antonio Road, Palo Alto, CA 94303 USA.

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

SAP, SAP Logo, mySAP.com, mySAP.com Marketplace, mySAP.com Workplace, mySAP.com Business Scenarios, mySAP.com Application Hosting, WebFlow, R/2, R/3, RIVA, ABAP, SAP Business Workflow, SAP EarlyWatch, SAP ArchiveLink, BAPI, SAPPHIRE, Management Cockpit, SEM, 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
	Tip

MiniApps	6
Suitable Applications for MiniApps	7
MiniApps: Where Do They Come From?	8
MiniApp Community.....	9
More About Creating MiniApps.....	11
Assigning MiniApps to a Role.....	11
MiniApp Maintenance	13
MiniApp Maintenance in the ABAP Workbench.....	13
Creating MiniApp Objects.....	14
Maintaining Entries for MiniApps.....	15
Maintaining MiniApps Using Tables.....	17
Changing MiniApps (Old Procedure).....	19
URL Templates.....	20
Special Characters in URL Templates.....	22
MiniApp Size.....	24
Personalization of MiniApps.....	24
Drag&Relate with MiniApps.....	25
Web Reporting in the Business Information Warehouse (BW).....	25
Web Reporting Process.....	26
Business Explorer Browser on the Web.....	28
Workflow Inbox MiniApp.....	28
Installation of the Workflow Inbox MiniApp.....	29
Configuration of the Workflow Inbox MiniApp.....	29
MiniALV	31
Overview.....	33
Functionality.....	34
Service Parameters	35
The MiniALV Trace.....	37
Implementing the Proxy Report.....	39
Displaying Lists Using REUSE_MINIALV_LIST_DISPLAY.....	39
First Steps.....	40
Interface Description	41
Application Scenarios.....	43
Defining Hyperlinks.....	43
Using Checkboxes.....	44
Coloring Columns, Rows or Cells.....	45
Using Exceptions.....	47
Displaying Values with Currency/Quantity Units	48
Field Catalog	49

Structure of the Field Catalog.....	50
Generating the Field Catalog Semi-Automatically.....	50
Generating a Field Catalog Manually	51
Fields of the Field Catalog.....	51
Texts in the Field Catalog.....	56
Layout Structure	57

MiniApps

Purpose

MiniApps are intuitive, easy-to-use Web applications or Web documents. After the user starts the mySAP.com Workplace, MiniApps provide the user with an overview of and access to his or her most important data on the initial screen.

The mySAP.com Workplace Server displays the MiniApps using a URL (Uniform Resource Locator). It does not matter where they reside.

Which MiniApps are seen in the mySAP.com Workplace depends on the user's role. The user can also [personalize the display and order of MiniApps \[Extern\]](#) and [add other MiniApps \[Extern\]](#) to the WorkSpace.

The Workplace architecture supports various MiniApp technologies and communication with any server, so it is open for third-party software.

Attributes – MiniApps should be:

Self-contained	MiniApps are independently executable objects.
Stateless	<ul style="list-style-type: none"> • No saving of context data to middleware and backend servers. • No permanent connection to middleware or backend servers. • After each request-response cycle, the connection is disconnected between the frontend and the servers again to enable to high number of accesses that are typical for intranet and Internet scenarios.
Simple	Everything should appear on one screen.
Lean	They should have only essential functions.
Access-providing	They should offer access to complex operations.
Direct	Access within a MiniApp to data and functions does not require navigation.
Personalizable	Users should be able to set up MiniApps as they prefer.
Active	MiniApps automatically fetch the data for the users.

Example

Examples of MiniApps:

- Alert
- Report
- E-mail access
- Calendar
- Search engine
- Company news, Web news

- Stock ticker
- Personal picture
- Telephone directory
- Note
- ToDo list

For more information about configuring the mySAP.com Workplace, see [mySAP.com Workplace: Overview \[Extern\]](#).

For configuration procedures, see the [Configuration Guide for the mySAP.com Workplace \[Extern\]](#).

For more information about roles, see [Roles in the mySAP.com Workplace \[Extern\]](#).

Suitable Applications for MiniApps

When you create MiniApps, remember the [Characteristics of MiniApps \[Extern\]](#). Primarily, it is technically critical that MiniApps are self-contained and stateless. From the design angle, direct (no-navigation) access to data is very important. In a MiniApp, it is best to not display too much data, and to display it in a simple way.

A good way to get ideas for new applications is to analyze what is already available, abstract the ideas, make analogies, and apply suitable patterns to other areas.

Examples of good data display in MiniApps:

The image shows a screenshot of a MiniApp interface. At the top, there is a header bar with the title "Confirm goods receipt" in bold black text. Below the header, the background is a light gray color. On the left side, there are two labels: "PO #" and "Date". To the right of each label is a white rectangular input field. At the bottom center of the form, there are two buttons: "Submit" and "Clear", both in blue text and underlined.

New Trip Request	
Destination	<input type="text"/>
Date	<input type="text"/>
Reason	<input type="text"/>
Submit More ...	

If you have a more complex application in mind, consider whether it might be better to implement it as an [Internet Application Component \[Extern\]](#) (IAC).

These transactions are also designed for occasional users and are easy and intuitive to use. They offer a way to use simple applications in the Web. For more information, see [ITS Implementation Models \[Extern\]](#). You can find some examples in [SAPNet](#) (Alias 46C).

MiniApps: Where Do They Come From?



Basically, you can distinguish between SAP System-dependent MiniApps and MiniApps that are completely separate from the SAP System.

You can use various methods to obtain MiniApps for your mySAP.com Workplace.

- Use MiniApps provided by SAP
 - For information about the MiniApps supplied with SAP Release 4.6C (including the [MiniALV \[Seite 31\]](#), for example), see the 4.6C documentation CD (MiniApps directory) and [SAPNet](#) (Alias 46C).
 - You can find SAP System-independent MiniApps (for example, the Outlook MiniApp) in the [SAP download area](#).
- Use MiniApps provided by SAP's partners
 - Contact the consulting partners directly
 - Use the [SAP MiniApp Community \[Seite 9\]](#) homepage (www.sap.com/miniapps). There you can find news tickers and stock tickers under *News Providers*.
 - If you are using the MiniApp Servlet for iViews, you can also implement iViews as MiniApps. For more information, see [MiniApp Servlet for iViews \[Extern\]](#) and [Importing iViews \[Extern\]](#).

- Develop your own MiniApps
 - If you use the SAP Business Information Warehouse (BW) with Release 2.0 or higher, you can use [BW Web Reporting \[Seite 25\]](#) to create MiniApps.
 - If you implement MiniApps based on ITS services, you can use the Web Application Builder in the ABAP Workbench, the SAP@WebStudio, or another editor to create the required [files \[Extern\]](#).
 - The [Web Application Builder \[Extern\]](#) enables translation administration, use of SAP design elements, and change logging. Make sure that you work in the customer namespace.
 - If you do not use the Web Application Builder, you must place the files manually in the directory structure on the [Internet Transaction Server \[Extern\]](#) (ITS) of the mySAP.com Workplace Server and check them into the SAP System. You can do this using the [SAP@WebStudio \[Extern\]](#) or the *Administration of Web Objects* (transaction SIAC1).
 - If you do not want to work with ITS services, use a popular development environment to create the underlying files for the MiniApps (for example, MS Visual Studio or IBM Visual Age). You then store these on a Web server so that they can be accessed via a URL. If you use this method to develop SAP System-dependent MiniApps, the [DCOM Connector \[Extern\]](#) is provided as a connection to the SAP System. For more information see *Development Zone* in the [MiniApps Community \[Seite 9\]](#) (www.sap.com/MiniApps).
 - You can use the [iView Creator \[Extern\]](#) on the [MiniApp Servlet for iViews \[Extern\]](#) to create iViews that are automatically copied to the [MiniApp maintenance transaction \[Seite 13\]](#) on the Workplace Server.
 - The mySAP.com Workplace provides the infrastructure to develop MiniApps for mobile devices and WAP-enabled mobile phones. For current information on this, see [SAPNet](#) (Alias Pervasive).
- See also [More About Creating MiniApps \[Seite 11\]](#).

To integrate MiniApps in the mySAP.com Workplace, you must assign them to the appropriate roles. You can find further information under [Assigning MiniApps of a Role \[Seite 11\]](#) and in the SAP Library under *Basis* → *ABAP Workbench: Tools* → *Integration of Internet-Services* → *Web Application Builder*.

MiniApp Community

www.sap.com/miniapps

Idea

The mySAP.com Workplace is conceived as an open enterprise portal and provides a framework of open Internet APIs for integrating third-party software. These interfaces are published on the MiniApps Community Homepage where you can also find documentation on development technology and examples that provide an easy introduction to creating MiniApps. Using this policy of open information, SAP makes it easier to integrate third-party software and Internet services.

As well as the publication and documentation of the mySAP.com Workplace interface, the idea behind the community is to create an information platform for exchanging information regarding development of MiniApps. By means of this community, SAP customers, SAP

partners, third-party software providers, and Internet developers are invited to write their own MiniApps. The information provided on the homepage supplies the knowledge required for this.

MiniApp Development

You can find explanations of the following approaches to creating MiniApps on the MiniApp Community Homepage:

- Development of SAP MiniApps using [flow logic \[Extern\]](#) and [HTML Business functions \[Extern\]](#).
- Development of MiniApps with generic Internet technology and using the Workplace APIs (personalization, launch event, refresh interface, Drag&Relate, and so on):
 - Server-side programming: HTML, CGI, ASP, JSP, NSAPI/ISAPI, server-side scripting, server-side includes
 - Client-side programming: HTML, DHTML, VBScript, JavaScript, Jscript, Java Applets, Active X, and so on
- Development of MiniApps using generic Internet technology that accesses SAP component systems. Middleware products such as the SAP Business Connector, DCOM Connector, or the RFC Library can be used here.
- Combinations of ITS service and use of generic Internet technology
- Internet services and Internet information can also be used as MiniApps (news, financial information, weather, and so on)

News and Documentation

A newsgroup and a Frequently Asked Questions (FAQ) document allow you to exchange ideas and experiences regarding MiniApp creation. You can place your own questions here and find solutions to certain problems. Everyone is invited to participate in the discussion.

Community participants can include their documents in the documentation area. If you have put together your knowledge and experiences in a tutorial or cookbook and want to spread this knowledge, we would be glad to put the document on the MiniApps Community Homepage. Use the *Contact Us* function on the MiniApps Community homepage.

Download

The MiniApps Community homepage contains a download page and a link page for MiniApps. SAP customers, SAP partners, third-party software providers, Internet content providers and anyone else who wants to provide MiniApps can put a link to their own homepage on the link page. The MiniApp is then available for downloading on the provider's homepage. As well as providing MiniApps commercially, you should also provide a version of the MiniApps free of charge, for example a demo version or shareware.

Advantages for Participants

- Access to the current documentation
- A large pool of information that goes significantly beyond the current official documentation and also covers the topic of integration of third-party software and Internet services in MiniApps.
- The discussion forum allows exchange of information, questions, and experiences.
- The download area (or link page) provides a fast shipment track to the customer.
- All participants can add their MiniApps and documentation to increase the content value of the page, thus helping to advance the community and making their own use of the page.

www.sap.com/miniapps

More About Creating MiniApps

For general information on these topics, see the SAP Library documentation or use SAPNet:

- Web Application Builder:
 - [Documentation \[Extern\]](#)
 - [Tutorial: Implementing Web Applications \[Extern\]](#)
 - SAP Library under *Basis* → *ABAP Workbench: Tools* → *Integration of Internet-Services* → *Web Application Builder*
- Flow logic:
 - [Developing Internet Applications with Flow Files: Reference Handbook \[Extern\]](#)
 - [Developing Internet Applications with Flow Files: Tutorial \[Extern\]](#)
- Internet Transaction Server (ITS) and WebRFC programming:
 - *Basis* → *Frontend Services* → *ITS/SAP@Web Studio* → [ITS Management \[Extern\]](#)
 - *Basis* → *Frontend Services* → *ITS/SAP@Web Studio* → *IAC Implementation* → [WebRFC Programming \[Extern\]](#)
 - *Basis* → *Basis Services/Communication Interfaces* → *Remote Communications* → [RFC Programming in ABAP \[Extern\]](#) and [DCOM Connector Logon Component \[Extern\]](#)
- Creating MiniApps using BW Web Reporting:
 - You can find further information in SAPNet under [BW Documentation Enhancements](#)
- Business Application Programming Interfaces (BAPIs):
 - *Basis* → *Middleware* → *BAPI* → [BAPI User Guide \[Extern\]](#)
- HTML Business
 - [Language Description \[Extern\]](#)

Assigning MiniApps to a Role

Use

The assignment of MiniApps to a role establishes which MiniApps the user can see in his or her WorkSpace.



If you want to assign MiniApps to a role, use the procedure described below.

If you want to make changes to existing MiniApps (for example, to the height or the Web address (URL)) that you assigned to a role using the [old procedure \[Seite 19\]](#), you can still do this using the old procedure. However, SAP recommends converting the MiniApps to the new procedure.


Prerequisites

You have created MiniApps and integrated them in the [MiniApp Maintenance Transaction \[Seite 13\]](#).



The concept of the [MiniApp Maintenance Transaction \[Seite 13\]](#) is that you define and store each MiniApp once only in a central location in the system. Therefore you only have to make changes to MiniApps that you have integrated in the MiniApp maintenance transaction there.

Procedure


1. In the role maintenance transaction (transaction *PFCG*), choose the role to which you want to assign MiniApps and go to change mode by choosing .



Note that you can only assign MiniApps to single roles.

2. Choose the *MiniApps* tab page.
3. Enter the names of the MiniApps that you require or choose one or more MiniApps using the possible entries help. You go directly to [MiniApp maintenance \[Seite 13\]](#) by double-clicking on a MiniApp.



You can create a MiniApp by going to the [Web Application Builder \[Extern\]](#) from here with the  *Create MiniApp* button.

4. Define the priority (*high*, *medium*, or *low*) for the MiniApps specified. This allows you to influence where the MiniApp is displayed in the WorkSpace.
5. Once you have entered all the desired MiniApps, save your input.

Result

You have assigned MiniApps to a role. When you use the role in the mySAP.com Workplace, these MiniApps appear on the right-hand side of the screen, in the WorkSpace.



If you want [newly created MiniApps \[Seite 14\]](#) to be displayed in the mySAP.com Workplace, you must publish them on the mySAP.com Workplace Server after you have created them in a component system so that the URL can be generated. For this, you must call transaction **S_WP_CACHE_RELOAD**. For more information, see [Transferring Data to the mySAP.com Workplace Server \[Extern\]](#).



You must activate the Workflow Inbox MiniApp using *Goto* → *Workflow MiniApp*. For more information, see [Workflow Inbox MiniApp \[Seite 28\]](#).

MiniApp Maintenance

Definition

The concept of the MiniApp maintenance transaction is that you define and store the MiniApp once only in a central location in the system. Therefore you only have to make changes to MiniApps that you have integrated in the MiniApp maintenance transaction there.

Use

To add MiniApps to a role in the role maintenance transaction, you must first add the MiniApps to the MiniApp maintenance transaction. There are different ways to do this depending on the Basis release level of the component system:

- For component systems with a Basis release level of 4.6D or higher, choose a development class in the ABAP Workbench (transaction *SE80*) and choose *Create* → *MiniApp* from the context menu (right mouse button). On the following screen, enter either the service file (for SAP System-dependent MiniApps) or a URL (this can also be a URL generated using [BW WebReporting \[Seite 25\]](#), for example).
- For component systems with a Basis release level up to and including 4.6C, you must make the entry in the MiniApp maintenance transaction using the corresponding tables.

For more information, see:

Basis Release Level of the Component System	MiniApp Maintenance
As of Release 4.6D	Using the MiniApp Maintenance Transaction in the ABAP Workbench [Seite 13]
Up to and including Release 4.6C	Using table maintenance [Seite 17]



If you want [newly created MiniApps \[Seite 14\]](#) to be displayed in the mySAP.com Workplace, you must publish them on the mySAP.com Workplace Server after you have created them in a component system so that the URL can be generated. For this, you must call transaction **S_WP_CACHE_RELOAD**. For more information, see [Transferring Data to the mySAP.com Workplace Server \[Extern\]](#).

MiniApp Maintenance in the ABAP Workbench

Use

You can integrate MiniApps from component systems with **Basis release 4.6D or higher** directly in the MiniApp maintenance environment in the ABAP Workbench.

If you want to integrate MiniApps that are located in component systems with a **Basis release level earlier than Release 4.6D** in the MiniApp maintenance transaction, you must use table entries. For more information, see [Maintaining MiniApps Using Tables \[Seite 17\]](#).

The MiniApps maintenance environment allows you to create *MiniApp* objects as development objects in the ABAP Workbench. You can also maintain the administrative data used to connect to the MiniApp Framework, which you use to make generic Workplace functions available to the MiniApp. You then define a set of MiniApp parameters to transfer values from the MiniApp to the Internet Service assigned to it.

Integration

The MiniApp maintenance transaction is fully integrated into the ABAP Workbench as part of the [Web Application Builder \[Extern\]](#). The objects of type *MiniApp* created in this tool are stored in the SAP Repository and as such are connected to the Transport Organizer.

You call the MiniApp maintenance transaction from the Object Navigator (transaction *SE80*) by choosing the required development class and choosing *Create* → *MiniApp* from the context menu (right mouse button).



If you want [newly created MiniApps \[Seite 14\]](#) to be displayed in the mySAP.com Workplace, you must publish them on the mySAP.com Workplace Server after you have created them in a component system so that the URL can be generated. For this, you must call transaction *S_WP_CACHE_RELOAD*. For more information, see [Transferring Data to the mySAP.com Workplace Server \[Extern\]](#).

Prerequisites

The release level of the component system is Basis release 46D (or higher).

Features

- Creating MiniApp objects
- Assigning MiniApp objects to Internet Services and URLs
- Maintaining other entries needed to integrate a MiniApp in the mySAP.com Workplace
- Defining parameters for transferring values between a MiniApp and the service assigned to it

Activities

[Creating MiniApp Objects \[Seite 14\]](#)

[Maintaining Entries for MiniApps \[Seite 15\]](#)

Creating MiniApp Objects

Procedure

To create a MiniApp object:

1. In the Object Navigator (SE80), choose a development class.
2. Choose *Create* → *MiniApp* from the context menu (right mouse button).

The system displays the *Create MiniApp* dialog box.


3. Enter other data for your MiniApp.


Enter a name and a meaningful short description for your MiniApp. Make sure that you work in the customer namespace.



The *Short description* is used by the system as the long text title (or tray header) of the MiniApp. In the mySAP.com Workplace, this long text is used as the MiniApp frame title.

You can give users of your MiniApp the option of personalizing the MiniApp. You have the following options:

- Choose *Personalization MiniApp* in the *Create MiniApp* window. This type of MiniApp allows users to customize data used by the MiniApp and to change their personal settings. You must specify this personalization MiniApp when maintaining the MiniApp that you want to personalize. The Workplace user can then call this MiniApp by choosing .
- You use automatic personalization via parameters. For more information, see the *Creating Parameters* section in [Maintaining Entries for MiniApps \[Seite 15\]](#).

4. Choose  to confirm your entries.

Result

The Web Application Builder displays the attributes of the MiniApp. This new development object of type MiniApp is stored in the SAP Repository and added to the object list of the development class.

After creating the MiniApp object, you can enter data for integrating the MiniApp in the Workplace and for using the generic functions of the MiniApp framework.

See also:

[Maintaining Entries for MiniApps \[Seite 15\]](#)

Maintaining Entries for MiniApps

Prerequisites

- You have [created the corresponding MiniApp object \[Seite 14\]](#).
- If you want to assign the MiniApp object to an Internet service, the service must already exist in the same system. The same applies if you want to assign a MiniApp object to a URL.
- Before you can create MiniApp parameters, these parameters must already exist as variables in the source code of the HTML template (of the assigned service).

Procedure

Maintaining other attributes of the MiniApp

1. Select the required MiniApp object from the object list.
2. Enter a short, medium and long *Description* for the MiniApp.


The description is used by the system as the title (or tray header) of the MiniApp. When the MiniApp is displayed in the mySAP.com Workplace, one of these descriptions will be used as the MiniApp frame title. The description used is determined by the dimensions of the MiniApp in the Workplace frame.

3. Specify the *Implementation type*.

The default option is *ITS Service*. This option assigns the MiniApp to an Internet service in the same system. If you choose this option, you must enter the name of the service from which the MiniApp is to be called in the *Service name* field. This service also contains the implementation of this MiniApp in the HTML template.



You can assign several MiniApps to the same Internet service.

If you choose the *Generic URL* radio button, you must enter an explicit Web address (URL). To do this, choose . In the dialog box that appears, enter a complete, explicit Web address (including all the necessary parameters). This implementation method allows you to attach each MiniApp to any number of URLs.



When you enter the URL, check whether [URL Templates \[Seite 20\]](#) are in use.

4. Specify the *Size* of the MiniApp in the Workspace.

Make sure that all the contents of the MiniApp fit in the frame. The dimensions of the MiniApp **do not** change dynamically when the data is changed. If the content is too big for the size defined, scroll bars appear. However, if possible, you should avoid having scroll bars in MiniApps.

For more information, see [MiniApp Size \[Seite 24\]](#).

5. If you want the MiniApp to be available to other device types (such as WAP-enabled mobile telephones, Palm Pilots, Pocket PCs) as well as in the Workplace, select the *Pervasive enabled* box.

As soon as you select this field, the *Device types* button appears. You can then call a dialog box in which you can assign each device type to a service.

Select the device types you want and enter the name of the calling services.

Choose  to confirm your entries.




Before selecting the *Autorefresh* field, read SAPNet Note **0304438**.

6. If required, enter a *Personalization MiniApp*.

The Workplace user can call this [special MiniApp \[Seite 14\]](#) by choosing  and then has the option of customizing the MiniApp data or changing personal settings.



Another option for providing the user with MiniApp personalization is to use the automatic personalization generation using parameters (see below).

7. Choose  to save your entries.

Creating parameters

1. Choose the *Parameters* tab.
2. Enter all the parameters that must be passed from the MiniApp to the assigned Internet service.



Before you can create MiniApp parameters, these parameters must already exist as variables in the source code of the HTML template (of the assigned service).



The system does not perform any corresponding checks.



Choose *Add service interface* to add all the interface parameters for the service to the list of MiniApp parameters.

If you flag one or more parameters as *Personalizable* in the appropriate column, the system automatically generates a [personalization function for the Workplace user \[Extern\]](#) that is based on the selected parameters. If you have created a *Personalization MiniApp* (see above), this takes priority.

Result

You have created a MiniApp object and entered all the administrative data that is needed to use the generic functions of the MiniApp framework for workplace integration.

You can now assign the MiniApp to one or more single roles. Open the role maintenance transaction (*PFCG*) and follow the procedure described in [Assigning MiniApps to a Role \[Seite 11\]](#).

Maintaining MiniApps Using Tables

Use

To integrate MiniApps that are located in **component systems with a Basis release level earlier than Release 4.6D** in MiniApp maintenance, you must maintain the table entries. You use the following transactions:

- W3WEBAPP: To integrate MiniApps in the MiniApp maintenance transaction (for developers)
- W3CUSX: To change values that the developer preset, such as width and height (for system administrators)
- W3DEVMINI: To change the values that the developer preset to device-specific values for MiniApps (for system administrators)
- W3DEVTRAN: To change the values that the developer preset to device-specific for transactions (for system administrators)



If you want to edit MiniApps in **component systems with a Basis Release starting with 4.6D** using the MiniApp maintenance transaction, you should read [MiniApp Maintenance in the ABAP Workbench \[Seite 13\]](#).

Prerequisites

- The current plug-in for the mySAP.com Workplace 2.10 has been imported into your component systems.
- You have created an ITS service file or have a URL for your MiniApp. For more information, see [MiniApps: Where Do They Come From? \[Seite 8\]](#).

Procedure

To add a MiniApp to the MiniApp maintenance transaction, proceed as follows:

1. Choose transaction W3WEBAPP in your component system.
The cross-client table *Change MiniApps* appears.
2. Choose *New entries*.
3. Give the MiniApp a name and enter the appropriate values in the following fields:
 - *Service/URL*: Enter either an ITS service file or a URL (this can also be a URL generated using [BW Web Reporting \[Seite 25\]](#), for example).



When you enter the URL, check whether [URL Templates \[Seite 20\]](#) are in use.

- *Application type*: Select whether you want to create a MiniApp or a MiniApp for personalization.
- *Implementation type*: Specify whether you are using an ITS service or a URL.
- *MiniApp for Pers.*: Enter the MiniApp for personalization that belongs to the MiniApp (if you have already created one).



Another option for providing the user with MiniApp personalization is to use the automatic personalization generation using parameters (see below).

- *Pervasive enabled*: Set this indicator (X) to show whether the MiniApp can also be used on pervasive devices.



For more information, see [Maintaining Entries for MiniApps \[Seite 15\]](#).


- *Autorefresh/Time*: For current information, see SAPNet-note 0304438.
- Height and width of the MiniApp: Enter the following values here:
 - **F**: for full height (only possible if you use half width) or full width
 - **H**: for half height (only possible if you use half width) or half width
 - If you specify full width, you must enter the value for the height in rows
 For more information, see [MiniApp Size \[Seite 24\]](#).
- *URL continuation*: Continue entering URLs here if they are very long.



For more information, see [Maintaining Entries for MiniApps \[Seite 15\]](#).

4. Save your changes.

Creating parameters

1. Choose *Table view* → *Other view*. Place your cursor in the *Parameters* and choose .
2. Enter all the parameters that must be passed from the MiniApp to the assigned Internet service.



Before you can create MiniApp parameters, these parameters must already exist as variables in the source code of the HTML template (of the assigned service).



The system does not perform any corresponding checks.



Choose *Add service interface* to add all the interface parameters for the service to the list of MiniApp parameters.

If you flag one or more parameters as *Personalizable* in the appropriate column, the system automatically generates a [personalization function for the Workplace user \[Extern\]](#) that is based on the selected parameters. If you have created a *Personalization MiniApp* (see above), this takes priority.

3. Save your changes.

Result

You have added your MiniApp to the MiniApp maintenance transaction. In this case you must [add MiniApps to Roles \[Seite 11\]](#) on the mySAP.com Workplace Server.



If you want [newly created MiniApps \[Seite 14\]](#) to be displayed in the mySAP.com Workplace, you must publish them on the mySAP.com Workplace Server after you have created them in a component system so that the URL can be generated. For this, you must call transaction **S_WP_CACHE_RELOAD**. For more information, see [Transferring Data to the mySAP.com Workplace Server \[Extern\]](#).

Changing MiniApps (Old Procedure)


Use

You should only use the procedure described here if you want to make changes to MiniApps (for example, the height or the URL) that you had already assigned to a role using the old procedure (up to release 2.00 of the mySAP.com Workplace).



For the new procedure (as of Release 2.10 of the mySAP.com Workplace) for assigning MiniApps to a role or changing them, see [Assigning MiniApps to a Role \[Seite 11\]](#).

Procedure

1. In the role maintenance transaction (*PFCG*), choose the required roles and branch to change mode by choosing .
2. Choose *Goto* → *MiniApps until component release 4.6C*.
3. Double-click to go to the *Change MiniApps in Role: Detail* view.



If you had only included one MiniApp, you are already on the details screen for this entry at step two.

You can make the following changes here:

- a. You can specify a title for the MiniApp in the *Header* field.
- b. The *Hght in pixel* field determines the display area for the MiniApp.



Unlike with the new procedure, you cannot display MiniApps in half width in the mySAP.com Workplace using this procedure.

- c. Specify the address of the MiniApp in the *URL* field.



When you enter or change the URL, check whether [URL Templates \[Seite 20\]](#) are in use.

- d. The *Sequence* of MiniApps that have already been included cannot be changed again (except by deleting the required MiniApps from the list and reassigning them to the role).

4. Save your entries.



You can find instructions for extended table maintenance in the system by choosing *Utilities* → *User instructions*.

URL Templates

Use

You should use URL templates to assign Web applications to a single role. In comparison to the use of fixed URLs, this procedure has the advantage of being flexible, for example, in the following scenarios:

- The Web server is still unknown when the roles are defined.
- You should be able to configure protocols (http, https).
- The language is user-specific.
- The role defines the client.

URL templates are character strings that contain tags. A tag is a character string that is delimited using pointed brackets (<>).

The tags are replaced by their actual values when a certain user generates the LaunchPad. The system reads tables to find a value for each tag. These values can depend on the current user and the logical system to which the Web application has been assigned.



```
<web_protcl>://<web_server>/<web_path_prefix>/ITS-  
Service/!/?~language=<language>&~client=<client>
```



You must not use special characters (such as blanks), instead you must replace them accordingly (for example, %20 for blanks), as the URL generation cannot

convert special characters. For more information, see [Special Characters in URL Templates \[Seite 22\]](#).



The URL is automatically generated for system-dependent MiniApps you included in the [MiniApp maintenance \[Seite 13\]](#).

The following tags are available:

URL Tag	Meaning	Where Does the Information Come From?
<logsys>	Logical system for which the role was created	Was defined as target system in the role definition.
<client>	Client in the target system	The target system client is used (from table TWPSRVLOGS).
<wps_client>	Client of the Workplace Server system	
<language>	Logon language	The language used to log on to the mySAP.com Workplace is used.
<web_server>	Name of the Web server (with ITS extension) for SAP GUI for HTML transactions, IACs, etc. that was entered for the logical system in the Customizing	The ITS that is entered for the target <i>Web Server</i> field is used.
<web_protcl>	Web protocol to be used for Web server <web_server>	From field <i>Web protocol</i> of table TWPURLSVR.
<web_path_prefix>	Path for Web server <web_server> with which the ITS services are called (scripts/wgate)	This information is read from table TWPURLSVR (field <i>Web path</i>) or table TWPURLPATH.
<win_server>	Name of the Web server (with ITS extension) for starting SAP GUI for Windows and SAP GUI for JAVA transactions, etc. that was entered for the logical system in the Customizing	The Web server that is entered for the target system in table TWPURLSVR in the <i>GUI Start Server</i> field is used.
<win_protcl>	Web protocol to be used for Web server <win_server>	The protocol that is entered in table TWPURLSVR in the <i>GUI Start Protocol</i> field is used
<win_path_prefix>	Path for Web server <win_server> with which the ITS services are called (scripts/wgate)	This information is read from table TWPURLSVR (field <i>GUI Start Path</i>) or table TWPURLPATH.
<kw_server>	Name of the Web server with Knowledge Warehouse enhancement that is entered for the logical system in Customizing	The Web server that is entered for the target system in table TWPURLSVR in the <i>Know.W.Web Server</i> field is used.
<kw_protcl>	Web protocol to be used for Web server <kw_server>	The protocol that is entered in table TWPURLSVR in the <i>KW Protocol</i> field is used
<kw_path_prefix>	Path with which Knowledge Warehouse services are called	This information is read from table TWPURLSVR (field <i>KW path</i>) or table TWPURLPATH.
<dr_server>	Name of the Web server with Drag&Relate enhancement that is	The Web server that is entered for the target system in table TWPURLSVR in

	entered for the logical system in Customizing	the <i>Drag&Relate Server</i> field is used.
<dr_protcl>	Web protocol to be used for Web server <dr_server>	The protocol that is entered in table TWPURLSVR in the <i>D&R Protocol</i> field is used
<dr_path_prefix>	Path with which Drag&Relate Servlet services are called	This information is read from table TWPURLSVR (field <i>D&R path</i>) or table TWPURLPATH.
<wps_web_server>	Name of the Web server (with ITS extension) for SAP GUI for HTML transactions, IACs etc. that was entered for the mySAP.com Workplace Server in the Customizing	The ITS that is entered for the mySAP.com Workplace Server in field <i>Web Server</i> in table TWPURLSVR is used, whatever the logical system.
<wps_web_protcl>	Web protocol to be used for Web server <wps_web_server>	The Web protocol of the mySAP.com Workplace Server is used from the <i>Web protocol</i> field of table TWPURLSVR, whatever the logical system.
<wps_web_path_prefix>	Path for Web server <wps_web_server> with which the ITS services are called	This information is read from tables TWPURLSVR (<i>Web Path</i> field) and TWPURLPATH. The path entered for the mySAP.com Workplace Server is used, whatever the logical system.
<lp_node_text>	Text of the corresponding LaunchPad entry in the language in which the LaunchPad is displayed	Was defined in the role maintenance (transaction <i>PFCG</i>)

For more information about Customizing tables, refer to [Overview: Customizing Tables for the mySAP.com Workplace \[Extern\]](#) and the [Configuration Guide for the mySAP.com Workplace \[Extern\]](#).

Special Characters in URL Templates

You must use the following schema to mask special characters that are used as part of a parameter value or parameter name in [URL Templates \[Seite 20\]](#):

The masked character consists of three characters:

- The first character is always a percentage character: %.
- The second and third characters are the ASCII code for the character converted into the hexadecimal system.



Special character	Mask for URL Templates
Blank character	%20
%	%25

+	%2B
---	-----

Only the developer of the MiniApp or the role administrator can decide whether a character functions as a special character with a particular function or as part of the name of a parameter. There is no way to make this differentiation during URL generation.

The following special characters have functions in URL templates:

- ? : introduces parameters
- & : separates the individual parameters
- = : separates the name and value of a parameter
- < : start of a tag
- > : end of a tag



If you do **not** want to use these parameters in the functions named above in a URL template, you must mask them as follows:

Special character	Mask
?	%3F
&	%26
=	%3D
<	%3C
>	%3E



Incorrect:

```
<web_protcl>://<web_server>/<web_path_prefix>/my_service/!/?param1=part1=part2
```

In this example, the URL generation cannot resolve which of the two equals signs is used to separate parameter names and values and which belongs to the name or value.

Correct:

- `<web_protcl>://<web_server>/<web_path_prefix>/my_service/!/?param1%3Dpart1=part2`

(The **name** of the parameter is *param1=part1* and the **value** of the parameter is *part2*)

- `<web_protcl>://<web_server>/<web_path_prefix>/my_service/!/?param1=part1%3Dpart2`

(The **name** of the parameter is *param1* and the **value** of the parameter is *part1=part2*)

The specification of how a correct URL must be structured is issued by W3C (WWW consortium, www.w3.org). The rules specified there relate directly to the URL templates described here, with the exception of the tag replacement. You can find details of URL syntax in the "Request for Comments" 2396.

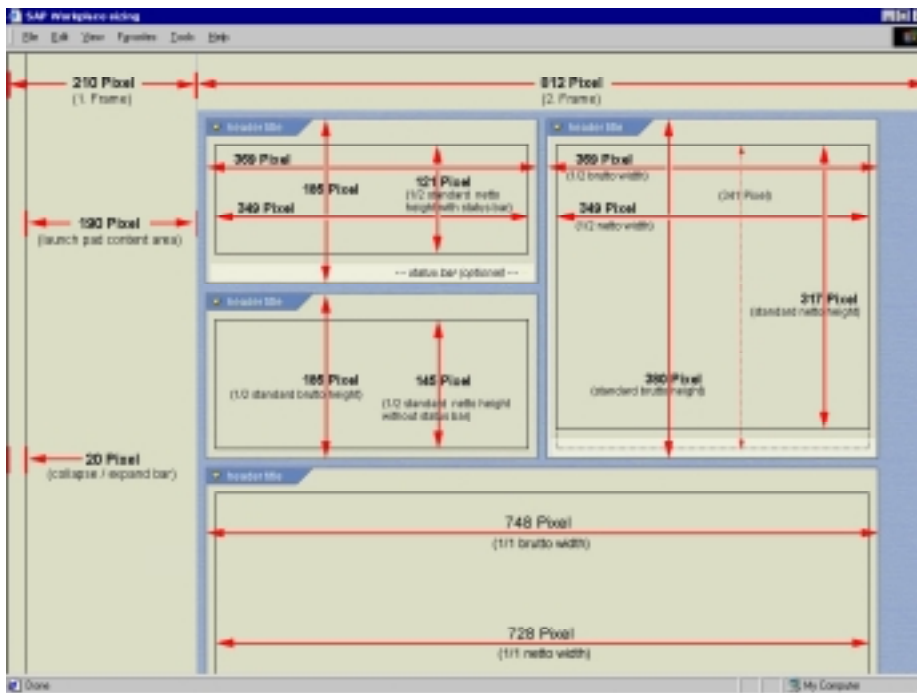
MiniApp Size

You can include MiniApps in the mySAP.com Workplace in two different widths:

- Full-width MiniApps: You can specify any row height for these MiniApps
- Half-height MiniApps: There are two possible heights for these MiniApps:
 - Full standard height (about seventeen rows)
 - Half standard height (about seven rows)

You specify these dimensions when you create the MiniApp in the [MiniApp Maintenance \[Seite 13\]](#) transaction.

One row corresponds to a height of twenty pixels.



Personalization of MiniApps

Users of the mySAP.com Workplace have the option of modifying the display and arrangement of MiniApps on their [Start Page \[Extern\]](#) to meet their personal requirements. For more information, see [Adding Your Own MiniApps \[Extern\]](#) and [Changing and Personalizing How MiniApps Are Displayed \[Extern\]](#).


The developer of the MiniApp must use one of the following options to allow the user to personalize the content of a MiniApp:

- Create a *Personalization MiniApp* in the [Web Application Builder \[Extern\]](#).
- Use the option of automatically creating personalization using parameters. To do this, the developer sets the *Personalizable* indicator on the *Parameters* tab page when creating the MiniApp.

For more information, see:

- [Settings for MiniApps \[Extern\]](#)

- [Maintaining MiniApps Using Tables \[Seite 17\]](#)
- The *Creating Parameters* section in [Maintaining Entries for MiniApps \[Seite 15\]](#).

Users of the mySAP.com Workplace see the  icon in the header row of the MiniApp. They can use this icon to access the dialog for personalization of the content of the MiniApp.

Drag&Relate with MiniApps

You can link objects in MiniApps with SAP transactions or a Web site using *Drag&Relate*:

- You can implement the *Drag&Relate* function between a MiniApp and a site in the World Wide Web. This allows you to drag a business object from the MiniApp to a transaction on an external Web site.
- You can implement the *Drag&Relate* function between a MiniApp and a transaction in a logical system. This allows you to drag a business object from the MiniApp to a transaction in a logical system.

For more information, see:

- [Drag&Relate \[Extern\]](#)
- [Drag&Relate from a MiniApp to the World Wide Web \[Extern\]](#)
- [Drag&Relate from a MiniApp to a Transaction in a Logical System \[Extern\]](#)

Web Reporting in the Business Information Warehouse (BW)

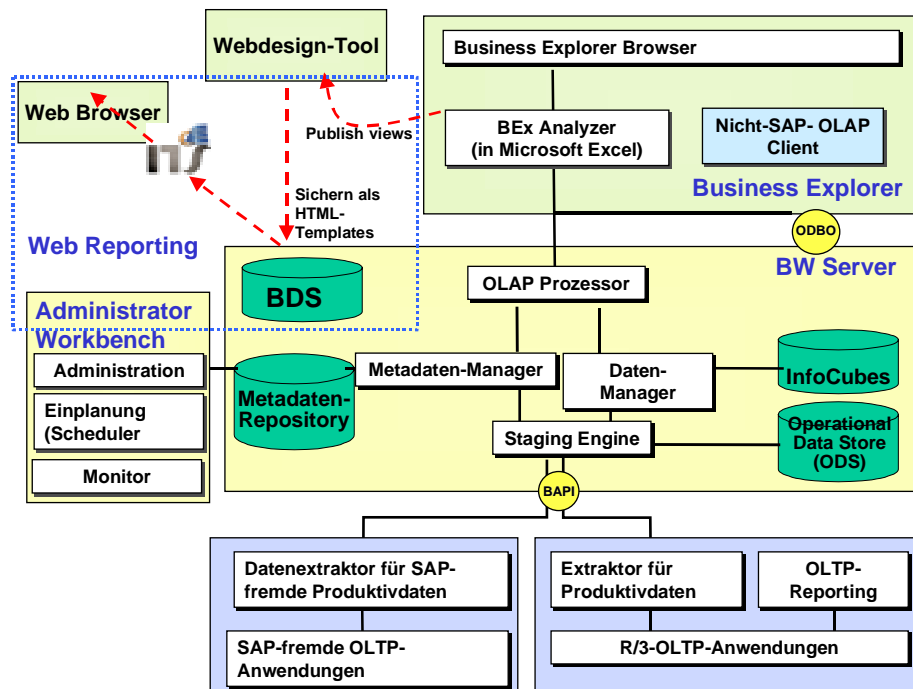
Purpose

You can use Web Reporting to publish queries defined in the Business Explorer Analyzer on an intranet or the Internet. These queries can be included in any HTML page and presented. You can embed various queries in an HTML page and use predefined navigation buttons or graphics to present data (Cockpit). The Web query data can be interpreted by navigation (expanding, refreshing, others).

Implementation Considerations

You can use any HTML editor or Web authoring tool to create HTML files for the presentation of Web queries. You can use the Business Explorer Analyzer *BW Web Publisher* component to save views of defined queries and supply display options for presenting them on the Web. You can open Web queries in any browser because pure HTML is used to display them.

Integration



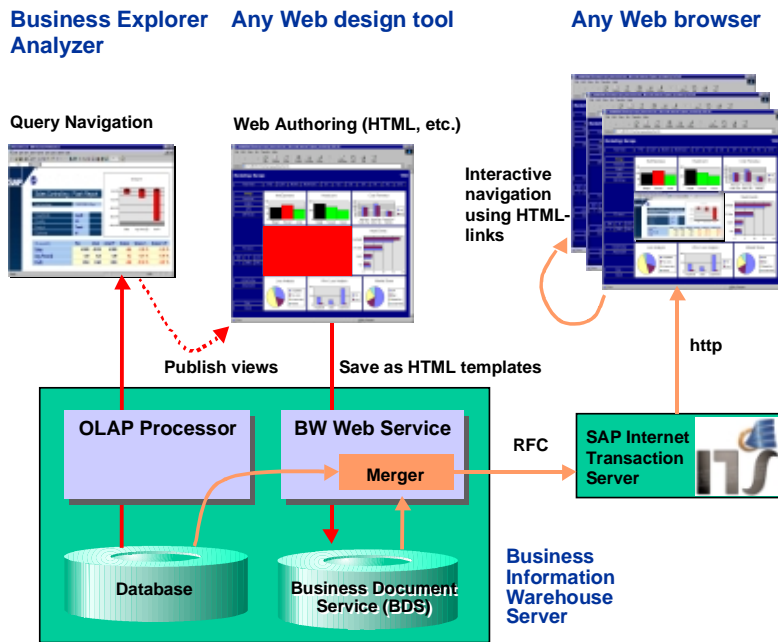
See also:

For more information, see [Web Reporting Process \[Seite 26\]](#) and [Business Explorer Browser on the Web \[Seite 28\]](#), in the Business Information Warehouse documentation.

Also read the documentation on the [WebReporting](#) functionality in SAPNet (Alias BW) under *Documentation* → *Documentation Enhancements*.

Web Reporting Process

This illustration shows the Web Reporting process flow:



1. Create an HTML template in any HTML editor or Web authoring tool. This HTML file can have any structure.
2. In the Business Explorer Analyzer, embed one or more queries in a workbook. Work with the navigation functions to create the required navigational state (view of the data). Save this query view (Choose *Save* → *Save query view*).
3. Then define how the view is presented in the Web. Choose *Tools* → *Publish to Web*.

This starts the BW Web Publisher. In the BW Web Publisher you create elements for views of the query. You can create a new element or edit an existing one. When you create a new element, you name it and select one of the provided element display options (table, pushbutton, others). An element is always defined for a view of a query in a particular workbook. Save the element. This stores it in the Business Document Service. Choose *Copy HTML tag* to put the HTML tag on the clipboard.

4. Switch to your HTML editor or Web authoring tool. Paste the HTML tag into the HTML page. Insert the tag in the required position between the “<Body> ... </Body>” tags. Save the HTML file locally.
5. Switch to the **BW Web Publisher** and import the HTML file into the Business Document Service (BDS). This makes the HTML file part of the document set. The HTML file is given a page ID, which is used later for the call in the URL.

A document set comprises one workbook plus any number HTML files (page IDs) and any number of query views (view IDs), with any number of elements (element IDs) relating to those views.

6. Enter the address of the **Internet Transaction Server (ITS)** in the BW Web Publisher. Maintain the address (file path) of the required **Web browser** as well.

Result

You have created a Web-enabled query. You can open the HTML file and display the current query data in the Web browser.

Business Explorer Browser on the Web

Use

Using the Business Explorer Browser on the Web (BEx Browser on the Web), the content of the BEx browser (roles, tasks, favorites) are displayed in a Web browser.

You can also use the BEx Browser on the Web if no BEx frontend is installed.

You can call HTML pages from the BEx Browser on the Web. These are created using BW Web Reporting and contain one or more Web queries. You can also call any other intranet or Internet pages from the BEx Browser on the Web. You can search for objects within the browser.



If one of your favorites (transaction *SESS*) is a URL called "Home", this Web page is displayed as the default start page in the BEx Browser on the Web.

Constraints

You cannot reposition folders and objects using Drag and Drop. You cannot maintain your own BW favorites in the BEx Browser on the Web.

In contrast to the Business Explorer Browser in the frontend, URLs that are not assigned to a folder are also displayed in the tree structure.

For more information, see the Business Information Warehouse documentation.

Workflow Inbox MiniApp

Integration

The Workflow Inbox MiniApp comprises the following components:

- Server component for the mySAP.com Workplace Server
Is part of the scope of the mySAP.com Workplace Server
- Server component for component systems
Is contained in the plug-in and can be imported into any system running SAP Basis Release 3.1H or higher
- ITS Service BCBMTWFM0001
Is on the ITS installation CD-ROM

Prerequisites

You have installed and configured the mySAP.com Workplace.

Features

If the Workflow Inbox MiniApp is active for a user, data selection is carried out in all logical systems that are:

- Activated globally (*Active* in table SWLIGL)
- Addressed by a role that is associated both with attribute *Read Workflow Inbox* (*Active* in table SWLIAG) and with the user

The Workflow Inbox MiniApp selects all the work items for the current user from these systems. Users can then choose to enter the *Inbox*, the *Outbox*, or *Resubmission*.

- The *Inbox* shows the user all work items that are ready to be processed. Users can execute a work item by clicking on its text. Clicking the *Display* icon displays the work item. Further actions on a work item are available from the work item display. While execution of the work item is always started with the appropriate type of GUI for the background task (transaction) according to the task classification, work item display is always presented in the SAP GUI for HTML. There are also task and work item status filters for the *Inbox*.
- For the *Outbox*, users can choose between various selection periods. They can also switch between categories of items to be presented from all addressed systems:
 - Workflows started
 - Work items executed
 - Work items forwarded
- In *Resubmission*, users find all of the work items for resubmission in the addressed systems. There are task and submission date filters for the work items.



If you use Internet Explorer 4 instead of Internet Explorer 5, the shuffler (filter option) is suppressed as standard in the *Inbox* and *Resubmission*.

- There is also an *Info* tab page, which has details of the current selection, for example, why a requested system could not be reached. In addition, you can specify your workflow substitute for each system there.

Users can update any view at any time by choosing *Refresh*.

For more information, see [Installation of the Workflow Inbox MiniApp \[Seite 29\]](#) and [Configuration of the Workflow Inbox MiniApp. \[Seite 29\]](#)

Installation of the Workflow Inbox MiniApp

The central component of the Workflow Inbox MiniApp is on the mySAP.com Workplace Server. From here all of the Workflow-relevant component systems are addressed. This is why the ITS part of the MiniApp must be on the mySAP.com Workplace Server ITS.

The component plug-in must be imported into all component systems where work items are to be selected.

All users who should see the Workflow Inbox MiniApp in their Workplace must be assigned to a role to which you have assigned the WORKFLOW_INBOX_MINIAPP MiniApp. For more information, see [Assigning MiniApps to a Role \[Seite 11\]](#).

Configuration of the Workflow Inbox MiniApp

On the mySAP.com Workplace Server, you have to define which component systems the service should ask for work items, for which users. You can do this in the following ways:

- By specifying logical systems that are queried for all users who have the MiniApp in their Workplace

You globally activate logical systems in table *SWLIGL* using transaction *SM30*. All logical systems of class *Workflow* that are flagged as *Active* in this table are queried for all users.

- By specifying logical systems that are queried for users with a particular role.

You globally activate logical systems by role in table *SWLIAG*. In this table you can associate a role with attribute *Read Workflow Inbox*. Association with this attribute means that users assigned that role are presented the *Workflow Inbox* from the logical system that the role points to. You can work on table *SWLIAG* in transaction *PFCG Role Maintenance* (choose *Goto* → *Workflow MiniApp*) or in transaction *SM30*.

To address a logical system using the *Workflow Inbox MiniApp*, it is essential that the *Customizing for URL generation* has been done for that logical system. This means that for each logical system, an RFC destination must be available via which the system can be accessed. See *Customizing for URL Generation*.

You can manipulate the appearance and function of the *MiniApp* by entering **parameters**. You can either enter the parameters directly in the service file (*bcbmtwfm0001.srv*, on the *ITS*) or in the [MiniApp maintenance transaction \[Seite 13\]](#). For more information, see [Maintaining Entries for MiniApps \[Seite 15\]](#).

The following parameters are available:

<p>FILTERTASK</p>	<p>Individual task group or individual single-step task or individual workflow, according to which filtering is to be performed</p> <p>Example: FILTERTASK=TSxxx</p>
<p>IT_FILTER_TASKS-TASKID</p>	<ul style="list-style-type: none"> • Task group and/or • Single-step task and/or • Workflows <p>According to which filtering is to be performed</p> <p>Example: IT_FILTER_TASKS-TASKID[1]=TSxxx IT_FILTER_TASKS-TASKID[2]=WSxxx IT_FILTER_TASKS-TASKID[3]=TGxxx</p>
<p>NOSHUFFLER</p>	<p>You can use this parameter to suppress the display of the shuffler in the inbox and in the resubmission folder.</p> <p>Possible value: NOSHUFFLER=X: No shuffler</p> <p>The default setting is that the parameter is not set. The shuffler is then displayed (see below for restrictions).</p>
<p>APPEARANCE</p>	<p>You can use this parameter to change the appearance of the <i>MiniApp</i>.</p> <p>Possible values: APPEARANCE=0 [default]: All tab pages are displayed. APPEARANCE=1: Only the workflow inbox is displayed. There are no</p>

	<p>tab pages.</p> <p>APPEARANCE =2:</p> <p>The <i>Info</i> tab page is suppressed.</p>
NOINPLACE	<p>You can use this parameter to control whether a new browser window is opened when the user executes or displays a work item in the workflow inbox.</p> <p>As standard, this parameter is not set. You execute and display dialogs from the Workflow Inbox directly in the Workplace.</p> <p>Possible values:</p> <p>NOINPLACE=X</p> <p>All dialogs are started outside the Workplace in separate browser windows.</p>

MiniALV

Purpose

The *MiniALV* is designed for displaying lists in a small output area of the screen. You can use the MiniALV both as a stand-alone MiniApp and as part of a MiniApp. Depending on the [style and functionality \[Seite 34\]](#) of the report, the user can scroll through the entries in the list, use combo boxes to call different lists via a MiniALV (for example, other selection variants and layouts), trace hyperlinks, and use checkboxes to mark entries in the MiniALV, for which a specific operation is to be executed.

You can use both checkboxes and hyperlinks in the MiniALV. In lists, you can also display your data in an icon view or as a pie chart or a bar chart. In addition, you can serialize the MiniALV output in order to display your data [on small devices \[Extern\]](#) using the Wireless Application Protocol (WAP).

If you use an Internet service of your own that includes the MiniALV service, you can use HTML^{Business} functions to position pushbuttons below the table output to enable users to perform actions related to table entries. You can also use a special hyperlink syntax to allow transactions to be started directly from within the MiniALV.

Implementation Considerations

The MiniALV is delivered together with SAP standard releases and with the Workplace. This documentation refers to the MiniALV version contained in the Workplace Plug-In, which allows you to display reports from older releases (< 4.6C) in the Web. The Workplace Plug-In comes with the Internet service `MINIALV_ADDON` and the function module `FLOW_LOGIC_MINIALV_CALL` (which is accessed by this service).

The Workplace Plug-In imports the function module `REUSE_MINIALV_LIST_DISPLAY`. You use this function module to implement a report in the SAP System that you want to display in the Web (in the standard SAP System, the ABAP object class `CL_GUI_ALV_GRID` is used, which is, however, only compatible with the MiniALV as of Release 4.6C). You can then display this report in both the SAP System and the Web. To differentiate between the part of the implementation that is required in the SAP System and the part that is required for display on the Web, the report is called the *Proxy Report* in the SAP System. See [Implementing the Proxy Report \[Seite 39\]](#) to find out how to implement the `REUSE_MINIALV_LIST_DISPLAY` function module correctly to create this proxy report. See [Functionality \[Seite 34\]](#) to find out

the options for the **MINIALV_ADDON** Internet service, that is, the Web part of the implementation.



If you are using a system of Release 4.6C level or higher, both the **MINIALV** Internet service (standard) and the **MINIALV_ADDON** (Workplace shipment) are available once you have imported the Workplace Plug-In. You can only use each version separately. [Notes for Web Add-On Applications \[Extern\]](#) provides an overview of the different components in the two versions.

The functions of the MiniALV Internet service that is delivered with the Workplace is in large parts identical with the functions of the standard system. The sections below therefore refer to the standard documentation with special reference to differences, if there are any.

Prerequisites

General Prerequisites

The function module **FLOW_LOGIC_MINIALV_CALL** must exist in the system. For the browser display, you call this module using WebRFC. Therefore, the module must be released in the Web Repository (transaction **SMW0**) for Internet use.



[Releasing Objects for Web Reporting \[Extern\]](#) describes how to use transaction **SMW0**.

The MiniALV in the Workplace Plug-In works via the **MINIALV_ADDON** Internet service that uses the services **SHUFFLER_ADDON** and **WAOFIG**. All these services must be published on your ITS.

Prerequisites for the Report

If you implement the proxy report (that is, the report in the SAP System), it must fulfill the following additional conditions:

1. An authorization group must be assigned to the report. This is the prerequisite for releasing the report for the Internet.
2. You must release the report for the Internet (also in transaction **SMW0**).

The exact procedure for this is identical for both version of the MiniALV and is described in [Prerequisites \[Extern\]](#) in the standard documentation.



In the standard system, you must use the **offline()** method to ensure that no frontend objects are generated. This is unnecessary when using the MiniALV in the Workplace Plug-In, because the **REUSE_MINIALV_LIST_DISPLAY** function module uses the ABAP list processor instead of the controls technology.

Proxy Report Variants

If [variants \[Extern\]](#) are saved for the proxy report, these can be chosen in the MiniALV from a combo box to refresh the data table based on this variant.



Variants are used to save input values of a selection screen so that they can be preassigned later on. Variants are also called *selection variants* or *report variants*.

These variants are only transported together with the report if the following prerequisites are met:

- Variants delivered by SAP to customers must have the prefix "SAP&" in their technical name. This ensures that the system adds them automatically as a transport object to the transport request of the report.
- Variants that are created by customers and are to be transported together with the report, must have the prefix "CUS&" in their technical name.



The ampersand ('&') is used in URLs to separate the value assignments of service parameters from each other. If you use an ampersand ('&') in the variant name and assign your MiniApp to a role [using the old method \[Seite 19\]](#) (by explicitly specifying a URL), you must replace the ampersand with '%26'.

The ampersand is replaced automatically if you:

- Assign the variant name within an Internet service
- Assign the MiniApp to the role [using the new method \[Seite 11\]](#).

Other Documentation

If you want to use the MiniALV, you must familiarize yourself with the flow logic and the development tools for Web applications. The tutorial [Developing Internet Applications with Flow Files \[Extern\]](#) shows you how to use the *SAP@Web Studio* to develop Web applications, and the documentation [Integration of Internet Services \[Extern\]](#) tells you how to implement a MiniApp using the *Web Application Builder* of the ABAP Workbench.

Overview

Architecture

The MiniALV is an Internet service based on flow logic. When started, this service retrieves the data of a report in the SAP System using the remote-enabled RFC function module `FLOW_LOGIC_MINIALV_CALL`. The following components are involved in that process:

- A report in the SAP System that generates the data for the list display in the MiniALV. The report must be created using function module `REUSE_MINIALV_LIST_DISPLAY`. It is also called *proxy report*.
- The function module `FLOW_LOGIC_MINIALV_CALL` that calls the respective proxy report. The module `REUSE_MINIALV_LIST_DISPLAY` passes the data without displaying the list in the SAP GUI.
- The MiniALV service `MINIALV_ADDON` that displays the data on the Internet Transaction Server (ITS) using templates.



See also: [Architecture \[Extern\]](#).

Consequences

To use the MiniALV, you must perform the following steps:

1. You write a proxy report in the SAP System or use an existing report. To do this, you use function module `REUSE_MINIALV_LIST_DISPLAY`. For more information on how to do this and on the prerequisites the report has to meet, see [Implementing the Proxy Report \[Seite 39\]](#).
2. You either use the MiniALV [as MiniApp \[Extern\]](#) or include the MiniALV service to use the MiniALV [as part of your own MiniApp \[Extern\]](#). In this context, you use [service parameters \[Seite 35\]](#) to customize the MiniALV to meet your individual requirements.

3. If you use the MiniALV as part of your own MiniApp, you can use additional HTML commands to influence the appearance of the MiniALV, use the dialog features of the MiniALV, and implement other functions, which you can link to the MiniALV.



For an overview of the functions for the MiniALV service, see [Functionality \[Seite 34\]](#).

4. You [assign your MiniApp to a role \[Seite 11\]](#) to represent it in the Workplace.

If the system returns the message that the RFC did not return data you can determine the cause of the error using the [MiniALV Trace \[Seite 37\]](#).

Functionality

In the simplest case, the MiniALV displays the list with a scrollbar. If selection variants or layouts exist, the user can select them using a combobox provided by the MiniALV to fill the list with new data.

There are different ways of how to configure the MiniALV according to the application requirements, and they depend on whether you

- define a MiniApp that only refers to the MiniALV service ([as MiniApp \[Extern\]](#)) or
- define a MiniApp that refers to your own Internet service, which includes the MiniALV service ([as part of a MiniApp \[Extern\]](#)). In relation to the MiniALV, this MiniApp is then a *MiniApp frame*.

In both cases, you can set the [service parameters \[Seite 35\] \[Extern\]](#) of the MiniALV service to control visual attributes or to offer a dialog.

Possible Enhancements of the MiniALV

Type	If integrated as	Description
Visual Attributes [Extern]	MiniApp/Part of a MiniApp	Only by setting service parameters
Comboboxes in the Shuffler [Extern]	MiniApp/Part of a MiniApp	Using a callback function module. Use the <code>WWW_USER_SETTINGS_FB_TEMPL_WAO</code> function module as a template. This function module is imported with the Workplace Plug-In.
Hyperlinks [Extern]	MiniApp: Only types <code>HTTP</code> , <code>documentation</code> and <code>transaction</code> Part of a MiniApp: All types	Definition of the links using proxy report – interpretation through MiniALV
Checkboxes [Extern]	Part of a MiniApp	Definition in the proxy report, evaluation in the MiniApp frame using JavaScript functions and the flow logic, operations using application-specific function module in the SAP system
Display the MiniALV as a graphic	MiniApp/part of a MiniApp	The MiniALV includes the service <code>WAOPIG</code> (currently only in the add-on as of Workplace 2.10) and calls its function <code>PIG_TemplateChart()</code> if the MiniALV service parameter <code>ma_style</code> is set to

		"graphic".
--	--	------------

Service Parameters

Use

The service parameters of the MiniALV service allow you to control the contents, the personalization and the visual attributes of the MiniALV.

Features

The service parameters of the two versions delivered (Workplace version and standard version) always have the same status. Since the Workplace version has shorter release cycles, some services are available at an earlier date in this version.




The Workplace version of the MiniALV does not allow you to display SAP Query reports in the MiniALV. The service parameters **query**, **usergroup** and **workspace** only work with the standard MiniALV.

The following new MiniALV service parameters are delivered with Workplace version 2.11. In the standard system, they are only available as of Release 5.0A.



For all other parameters, see the [standard documentation \[Extern\]](#).

MiniALV Functions

Parameter (default)	Meaning
ma_sort (on off)	The column headings are displayed as hyperlinks for sorting the column. When the column heading is clicked for the first time, the MiniALV sorts the table in ascending order; when the heading is clicked a second time, the table is sorted in descending order. A symbol in the column heading indicates the sort order applied.
ma_trace (on off)	Set this parameter to 'on' to find out what caused the error "RFC did not return data". See: The MiniALV Trace [Seite 37] .
ma_stateless (on off)	In the default setting, the MiniALV holds a state of the current user session on the ITS. If this parameter is set to "on", the MiniALV becomes stateless and terminates the ITS user session after being called. MiniALV-specific parameters are resent to the ITS using a hidden form (for example, if another variant has been chosen using the shuffler).  We recommend that you set this parameter for all previous and old developments that use the MiniALV in order to reduce the ITS load. The parameter has been introduced for compatibility reasons only.

Visual Attributes

Parameter (default)	Meaning
<code>ma_color</code> (on off)	The colors defined in the proxy report for rows, columns or individual cells are passed on to the MiniALV and displayed there (see Coloring Columns, Rows or Cells [Seite 45]). Set this parameter to 'off' if you want to display the MiniALV with its custom colors.

PIG Graphic

PIG means *Portable Interactive Graphics* and is a service for MiniApps that allows you to embed graphics. The MiniALV includes this service and provides its own parameters to display graphics in the list. The PIG graphic is available for the Workplace as of version 2.10.



For more information, see the PIC documentation under [Portable Internet Graphics \(PIG\) \[Extern\]](#).

(If `ma_style=graphic`)

Parameter (Default)	Meaning
<code>ma_view</code> (pie, bars, columns , lines and other)	Graphic type. Corresponds to parameter <code>type</code> of <code>PIG_TemplateChart</code> .
<code>ma_mode</code> (java igs)	Method used to display the graphic on the frontend: <ul style="list-style-type: none"> • Java applets • IGS (Internet Graphics Server) For advantages and disadvantages see the PIG documentation.
<code>ma_width</code> (320)	Width of the graphic
<code>ma_height</code> (120)	Height of the graphic
<code>ma_textcolumn</code> (1)	Number of the column in the output table for the data of the abscissa (X axis)
<code>ma_valuecolumn</code> (2)	Number of the column in the output table for the data of the ordinate (Y axis)
<code>ma_colors</code> []	Array to determine colors

Activities

In order to display data in the MiniALV, you must at least assign a value to the service parameter `report`. You can also start the MiniALV using a layout or a report variant (with service parameters `layout` or `variant`). If the MiniALV does not display any data, proceed as follows:

1. Ensure that all prerequisites for display are met (see *Prerequisites* section in [MiniALV \[Seite 31\]](#)).
2. If you get the error message that the RFC did not return data, you can use the [trace mode \[Seite 37\]](#) (set service parameter `ma_trace` to 'on') to determine at which point no data was passed and correct the error.

The MiniALV Trace

Use

If you receive the error message that the RFC did not return data, this means that the MiniALV service has not received any data and is therefore unable to display the table. Possible reasons may be that the report does not output any data at all or that the RFC contains errors.

The *MiniALV Trace* helps you to find out the cause of the error more precisely (see also: [MiniALV Architecture \[Extern\]](#)).

Prerequisites

To be able to use the MiniALV Trace, you must set the [service parameter \[Seite 35\]](#) `ma_trace` to 'on'.

The MiniALV Trace is available in the Workplace version as of Release 2.11 and in the standard version as of Release 5.0A.

Features

To track down the error, the system writes entries to a database table for the MiniALV Trace at several times. The system then returns a table with these entries for the MiniALV service as the result of the trace run. The table contains the following information:

Column	Meaning
Report name	Name of the report passed with the service parameter <code>report</code>
Variant	Name of the variant passed with the service parameter <code>variant</code>
Name	User who logged on to the system to execute the report
Date/time	Time at which the corresponding MiniALV component was called
Trace time	Trace times sorted in ascending order
Text	Specifies the component in which the trace entry was generated (see below)

The trace result is returned even if the MiniALV has been displayed correctly. In this case, two tables are shown in the Web browser.



The MiniALV Trace is also displayed if error messages other than RFC did not return data occur (for example, if the variant specified does not exist).

Trace Times

The trace times allow you to determine where the processing of the MiniALV output was interrupted:

Number	Text
1	<code>FLOW_LOGIC_MINIALV_CALL(START)</code>
2	<code>FLOW_LOGIC_MINIALV_CALL(BEFORE SUBMIT REPORT)</code>
3	<code>REUSE_MINIALV_LIST_DISPLAY(START)</code>
4	<code>MINI_TABLE_FOR_DISPLAY(END)</code>
5	<code>FLOW_LOGIC_MINIALV_CALL(AFTER_SUBMIT_REPORT)</code>
6	<code>FLOW_LOGIC_MINIALV_CALL(END)</code>

The text specifies the component (function module) of the MiniALV for which the system generated a trace entry. This tells you the following:

- If **no** trace is output and the error message `RFC did not return data has been issued`, either the proxy report or the function module `FLOW_LOGIC_MINIALV_CALL` has not been released for the Internet.
- If only the first trace time is output, in most cases either the proxy report or the variant specified contains errors or does not exist.
- If only the first three trace times are output in the trace, the error in most cases is due to the fact that the proxy report generates frontend objects, the function module `ITS_PING` has been used or the proxy report creates an endless loop.

Activities

Activating the Trace

For your own MiniApps that use the MiniALV and are still under development, you can activate the MiniALV Trace simply by setting the service parameter `ma_trace` to `on`.

If you want to temporarily activate the MiniALV Trace for an existing MiniApp, proceed as follows:

1. Click the right mouse button to call the context menu of the Internet Explorer in the MiniApp frame and choose `Create Shortcut`. The Internet Explorer then creates a shortcut to the MiniApp on your desktop.
2. Click the right mouse button on this shortcut and choose `Properties`. Under `Web Document` you can edit the hyperlink that points to the MiniApp.
3. For a MiniApp that uses the MiniALV service directly but does not include the service in another service, you can add `ma_trace=on` (separated by an ampersand from the other service parameters) to the URL.



For stateful MiniApps, the ITS adds characters for the user session to the URL. So that you can start the service with `ma_trace=on` added to the URL, you must delete these characters. For information on the URL structure, see [Calling the Internet Service \[Extern\]](#) in the standard documentation.

4. Restart the Internet service using the modified URL.

The Web browser should now display the MiniApp in the entire output area and contain the trace (if not, the following exceptions apply: the MiniALV service is included in a frame MiniApp that has disabled the trace explicitly with `ma_trace="off"`; the MiniApp does not use the MiniALV; the URL is not correct).

Correcting the Error

Try to correct the error. To do this, first check whether the [prerequisites \[Seite 31\]](#) for display have been met and then correct the problem.



See also [RFC Did not Return Data \[Extern\]](#) in the standard documentation.

You can use the report `bcalv_minialv_trace` to display or delete existing trace runs.

Implementing the Proxy Report

Use

The proxy report selects the data in the SAP System that you want to display in the Workplace using the MiniALV. In this context, the MiniALV copies particular properties and functions of the proxy report.

Integration

The proxy report must use function module `REUSE_MINIALV_LIST_DISPLAY` to display a list. This function module has been implemented in the Workplace especially for the MiniALV to enable customers operating a Release prior to 4.6C to use the MiniALV. Customers working with Release 4.6C or higher, can implement the proxy report using the class `CL_GUI_ALV_GRID`.



For a description of the MiniALV in the standard system, and of the class `CL_GUI_ALV_GRID` see the section [The MiniALV \[Extern\]](#) in the documentation [ALV Grid Control \(BC-SRV-ALV\) \[Extern\]](#).

Features

Function module `REUSE_MINIALV_LIST_DISPLAY` displays simple lists in the SAP System, which can preferably be used for display in the MiniALV. For this reason, the function module features are restricted to functions that are relevant to the display and interaction of lists in the MiniALV. In this context, existing structures from the SAP System were reused.



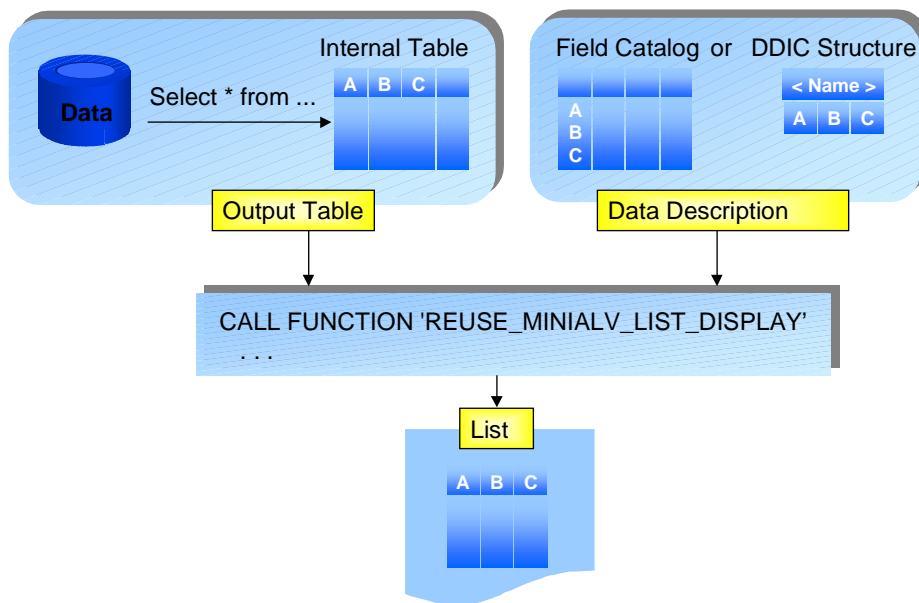
Since the function module focuses on features for the MiniALV, not all parameters of these structures are relevant. Therefore, you must use only those parameters and functions that are described in this documentation.

The function module and the relevant structures allow you to:

- Display data from an internal table (the *output table*) according to the currency or unit of measurement used
- Store layouts (current properties of a list, such as sorting or filter), which the user can select in the MiniALV (see [Layouts \[Extern\]](#) in the ALV Grid Control documentation; in our case, the layout is passed when `REUSE_MINIALV_LIST_DISPLAY` is called.)
- Use [exceptions \[Seite 47\]](#) (standardized icons). You can use the icons, for example, to indicate that threshold values are exceeded in a row.
- Integrate [hyperlinks \[Seite 43\]](#) [\[Seite 43\]](#) into the list, which the user can follow in the MiniALV.
- Integrate [checkboxes \[Seite 44\]](#) into the list, which allow operations on multiple lines in a round trip.

Displaying Lists Using `REUSE_MINIALV_LIST_DISPLAY`

Displaying a simple list using function module `REUSE_MINIALV_LIST_DISPLAY` executes as shown in the diagram below:



To display a list using function module `REUSE_MINIALV_LIST_DISPLAY`, the application must pass the following data:

- an internal table containing the data to be displayed – the *output table*
- a *data description* for the output table

The data description contains information on the individual columns or fields of the output table. This may be information on the data type or on the output format of the value to be displayed.

You can provide the data description for function module `REUSE_MINIALV_LIST_DISPLAY` in several ways:

- You can pass a structure to the module that is defined in the Data Dictionary (DDIC) for the output table. In this case the module automatically generates a *field catalog*, in which the data description is stored.
- You can create the field catalog manually and pass it at the function module interface.
- You can generate the field catalog from a structure of the Data Dictionary, modify it, and then pass it to the function module.

In any case, function module `REUSE_MINIALV_LIST_DISPLAY` accesses the data description via the field catalog.

First Steps

Simplest Call

The report below displays a flight table (transparent table `SFLIGHT`) using function module `REUSE_MINIALV_LIST_DISPLAY`:

```
REPORT DEMO.
```

```
DATA: GT_OUTTAB LIKE TABLE OF SFLIGHT.
```



```
* Data selection
SELECT * FROM SFLIGHT INTO TABLE GT_OUTTAB.

* Generate field catalog automatically
* and display list

CALL FUNCTION 'REUSE_MINIALV_LIST_DISPLAY'
  EXPORTING
    I_STRUCTURE_NAME   = 'SFLIGHT'
  TABLES
    T_OUTTAB           = GT_OUTTAB.
```

Result

At the call, the function module automatically generates the field catalog. Then, it displays the list on a carrier screen using the list processor. The function module sets its own GUI status that includes standard list functions such as sorting, filters, summing.



Before you display the list in the MiniALV, you should test whether your proxy report produces the correct output.

Constraints

When you generate the field catalog automatically, you cannot:

- Suppress columns before you display the list for the first time. The display always contains all fields of the output table.
- Use checkboxes in a column
- Add other fields to the output table that are not contained in the DDIC structure
- Use output formats that deviate from the standard

Interface Description

Call

```
CALL FUNCTION 'REUSE_MINIALV_LIST_DISPLAY'
  EXPORTING
    I_INTERFACE_CHECK   = <var. of type C>
    I_CALLBACK_PROGRAM  = <var. of type SY-REPID>
    I_STRUCTURE_NAME    = <var. of type DD02L-TABNAME>
    I_LIST_TITLE        = <var. of type WAOMA_LVC_TITLE>
    IS_LAYOUT           = <structure of type SLIS_LAYOUT_ALV>
    IS_LAYOUT2          = <structure of type LVC_S_LAY2_WAO>
    IT_FIELDCAT         = <internal table of type SLIS_T_FIELDCAT_ALV>
    IT_SORT             = <internal table of type SLIS_T_SORTINFO_ALV>
    I_DEFAULT           = <var. of type C>
    I_SAVE              = <var. of type C>
    IS_VARIANT          = <structure of type DISVARIANT>

  TABLES
    T_OUTTAB            = <internal table>
    IT_FIELDCAT_HLINK  = <internal table of type LVC_S_FC2_WAO>
    IT_HLINKS          = <internal table of type LVC_S_HYPE_WAO>.
```

Parameters

Export Parameters


Name	Description
I_INTERFACE_CHECK	In the list display, you can enter <code>&SOS</code> in the command field to check whether the passed parameters are consistent. If the system creates a short dump directly at the list display, you cannot use the command field. Instead, you use this parameter to execute the check.
I_CALLBACK_PROGRAM	Callbacks have no meaning in the MiniALV context. This parameter is used only to facilitate conversion of old reports. You must never pass the system field <code>SY-REPID</code> here directly.
I_STRUCTURE_NAME	Name of the DDIC structure (for example, 'SFLIGHT') for the data in the output table. If you specify this parameter, the field catalog is generated automatically [Seite 40] .
I_LIST_TITLE	The MiniALV displays the title specified here as the frame heading. If you start the report on the backend, the title is not visible.
IS_LAYOUT	Structure for passing the field names for the checkboxes or the exception.
IS_LAYOUT2	Structure with character field <code>EXCP_GROUP</code> . You use it to determine the type of exception (1-7).
IT_FIELDCAT	Describes the structure of the output table and the format of the data to be displayed.
IT_SORT	Table with sort attributes for columns that are to be sorted initially.
I_DEFAULT	This parameter determines whether the user is allowed to preset the layout: <ul style="list-style-type: none"> 'X': presetting the layout is allowed SPACE: presetting the layout is not allowed If it is allowed to preset the layout, if such a presetting exists, and if no other layout is specified in <code>IS_VARIANT</code> , then the system automatically loads this layout default when this method is called.
I_SAVE	Determines the possibilities a user has when storing a layout: <ul style="list-style-type: none"> 'X': only global storage 'U': only user-specific storage 'A': equals 'X' and 'U' SPACE: no storage
IS_VARIANT	Determines the layout to be used to display the output table. When using this parameter, you must fill at least the <code>REPORT</code> field of the structure of type <code>DISVARIANT</code> . <p style="text-align: center;"></p> In old reports <code>REPORT</code> was often set using the parameter <code>I_CALLBACK_PROGRAM</code> , so that there was no need to fill <code>IS_VARIANT</code> . <code>REUSE_MINIALV_LIST_DISPLAY</code> takes this into consideration.

Table Parameters

Name	Description
T_OUTTAB	Output table containing the data to be displayed.
IT_FIELDCAT_HLINK	Table for assigning hyperlink handles to columns of the output table.
IT_HLINKS	Table for defining hyperlinks [Seite 43] and their handles.

Application Scenarios

Defining Hyperlinks

Purpose

Use the `HREF_HNDL` field of the [field catalog \[Seite 51\]](#) to assign a handle for a hyperlink address to **all** cells of a column.

This section describes how you can link the cells of a column to different hyperlink addresses.

Process Flow

1. Use type group `SLIS`.

```
TYPE-POOLS: SLIS.
```

2. Define an internal table for the field catalog with line type `LVC_S_FC2_WAO` and a hyperlink table with line type `LVC_S_HYPE_WAO`:

```
data: gt_fieldcat2 type TABLE OF LVC_S_FC2_WAO,
      gt_hypetab type TABLE OF LVC_S_HYPE_WAO.
```

3. To display a hyperlink only in one cell of a column, you need an additional field in the output table. Enhance your output table by a field of type `INT4` as shown below:

```
DATA: BEGIN OF GT_OUTTAB OCCURS 0.
      INCLUDE STRUCTURE <DDIC structure>.
      DATA: HL_FNAME TYPE INT4. "hyperlink handle for field FNAME
DATA: END OF GT_OUTTAB.
```



For each column of the output table that you want to contain hyperlinks on a cell level, you must define one additional field for the handle. In this example, we want to display hyperlinks only in one column.

4. Generate the field catalog for your output table [semi-automatically \[Seite 50\]](#) or [manually \[Seite 51\]](#). In the [field catalog \[Seite 51\]](#) assign the name of the field for the hyperlink handle (in the example: `HL_FNAME`, see above) to field `WEB_FIELD`. The ALV Grid Control can now interpret the additional field correctly.
5. Create your hyperlink table with all desired targets, for example:

```
data: ls_hype type lvc_s_hype.

ls_hype-handle = '1'.
ls_hype-href = 'http://www.sap.com'.
append ls_hype to gt_hypetab.
```

```
ls_hype-handle = '2'.  
ls_hype-href = 'http://www.mysap.com'.  
append ls_hype to gt_hypetab.
```



Besides the prefix `http`, the MiniALV can also interpret other prefixes (for example, to start a transaction). See: [Hyperlinks \[Extern\]](#) in the standard documentation of the MiniALV.

6. Select your data and copy it into the output table.
7. Use a loop to read line after line of the output table. Within the loop, assign the desired handle to the field for the hyperlink handle (in the example: `HL_FNAME`). If the handle remains blank for a line, no hyperlink is displayed.
8. Use parameter `IT_FIELDCAT_HLINK` to pass the field catalog, parameter `IT_HLINKS` to pass the hyperlink table, and parameter `T_OUTTAB` to pass the output table in the function module interface.

Result

All cells of the column, to which you assigned a handle using the additional field, are displayed as hyperlinks. If there is no entry in the hyperlink table for the assigned handle, the contents of the cell are displayed normally.

Using Checkboxes

Purpose

You can use one column of the list to display checkboxes. In the SAP system, checkboxes cannot be changed. In the MiniALV, checkboxes are automatically input-enabled and you can use several methods to query and update them.



For a description of how to evaluate checkboxes in the MiniALV, refer to [Checkboxes \[Extern\]](#) in the standard documentation.

You can use the flow logic to pass the statuses of the checkboxes to the SAP system. This allows you to execute operations on a selection of lines in the list in a round trip.

Process Flow

1. Use type group `SLIS`:

```
TYPE-POOLS: slis.
```

2. Define the [layout structure \[Seite 57\]](#):

```
GS_LAYOUT TYPE SLIS_LAYOUT_ALV.
```

3. Enhance your output table by a column for the checkboxes:

```
DATA: BEGIN OF GT_OUTTAB OCCURS 0.  
  INCLUDE STRUCTURE SFLIGHT.  
DATA: CHECKBOX TYPE C.  
DATA: END OF GT_OUTTAB.
```

4. Select your data and set the checkboxes to the desired values:

```
DATA: GT_SFLIGHT LIKE TABLE OF SFLIGHT WITH HEADER LINE.
```

```
SELECT * FROM SFLIGHT INTO TABLE GT_SFLIGHT.  
LOOP AT GT_SFLIGHT.  
  MOVE-CORRESPONDING GT_SFLIGHT TO GT_OUTTAB.  
  GT_OUTTAB-CHECKBOX = 'X'.  
  APPEND GT_OUTTAB.  
ENDLOOP.
```

5. Use the layout structure to pass the field name for the checkbox:

```
gs_layout-box_fieldname = 'CHECKBOX'.
```

6. Pass the structure name of the field, the layout structure, and the output table at the function module interface:

```
CALL FUNCTION 'REUSE_MINIALV_LIST_DISPLAY'  
  EXPORTING  
    I_structure_name = 'SFLIGHT'  
    is_layout        = gs_layout  
  TABLES  
    T_OUTTAB        = GT_OUTTAB[ ].
```

Result

The checkboxes are displayed in the first column.

Coloring Columns, Rows or Cells

Use

The MiniALV uses the color settings of the proxy report. This allows you to highlight columns, rows or cells.

In the flight data model, for example, you could highlight cancelled flights in the table in red.



If you want to highlight rows based on threshold values, it is easier to use [exceptions \[Seite 47\]](#).

Prerequisites

The service parameter `ma_color` must be set to 'on'. This is the default setting.

Procedure

To determine the color value, you pass three values:

- The color number beginning with the character 'C'
- The value 1 or 0 to switch on / off intensified display
- The value 1 or 0 to switch on / off inverse display

You pass these values either through the components of a structure or through a four-digit character field (see below), for example 'C600' for red (no intensified and no inverse display).



For more information on color coding, see the F1 help on the ABAP statement `FORMAT`.

Coloring Columns

To color a column, you set the field [EMPHASIZE in the field catalog \[Seite 51\]](#) for this column to the desired four-digit color value.

Coloring Rows

1. Define the [layout structure \[Seite 57\]](#) of type `SLIS_S_LAYOUT_ALV`.
2. Add a four-digit character field to your output table as in the following example:

```
DATA: BEGIN OF GT_OUTTAB OCCURS 0.
INCLUDE STRUCTURE <DDIC structure> .

DATA: linecolor(4) type c. "Color for corresponding line
DATA: END OF GT_OUTTAB.
```
3. Select your data and copy it into the output table.
4. Read one output table row at a time in a loop. To change the color of a row, assign the above four-digit color code to the character field.
5. Assign the name of the internal table to the field `INFO_FIELDNAME` of the layout structure (in this case `'LINECOLOR'`, see step 2).
6. Pass the layout structure and the output table using the function module `REUSE_MINIALV_LIST_DISPLAY`

Coloring Cells

1. Define the [layout structure \[Extern\]](#) of type `SLIS_S_LAYOUT_ALV`.
2. Add a color table of type `LVC_T_SCOL` to your output table as in the following example:

```
DATA: BEGIN OF GT_OUTTAB OCCURS 0.
INCLUDE STRUCTURE <DDIC structure> .

DATA: CT TYPE slis_t_specialcol_alv. "Table for colors
DATA: END OF GT_OUTTAB.
```
3. Select your data and copy it into the output table.
4. Read one output table row at a time in a loop. A row of the color table (structure type `slis_specialcol_alv`) has three fields. If the field `NOKEYCOL` is set, you can change the color of key fields. Assign values to the remaining fields as follows:
 - If you want to color the entire cell, assign the desired values to the fields of the structure `COLOR` (the color value (without `'C'`) to `col`, the intensified indicator to `int`, and the inverse indicator to `inv`). The field `fieldname` remains empty in this case.
 - If you only want to color certain columns of the row, you must append one row to the color table for each column you want to color. You assign the name of the desired column to the field `fieldname` and the corresponding values to the fields of the structure `COLOR`.

As a final step, you assign the color table of the current row to the output table.
5. Assign the name of the internal table to the field `COLTAB_FIELDNAME` of the layout structure (in this case `'CT'`, see step 2).
6. Pass the layout structure and the output table using the function module `REUSE_MINIALV_LIST_DISPLAY`.




Result

The colors are displayed both in the SAP System and in the MiniALV.

Using Exceptions

Purpose

You use exceptions to indicate graphically that a threshold value was exceeded in a line of the list. In the flight model, for example, you would use a red traffic light to indicate that a flight is booked to capacity. By means of the colors you use the user can interpret the displayed data faster:
































Display	Internal value	Indicates, for example
	3	(Almost) full capacity available
	2	Half to almost no capacity available
	1	No more capacity available

The application developer must determine the threshold values and adapt the values of the exceptions to them.

Features

The function module `REUSE_MINIALV_LIST_DISPLAY` in the SAP system can display only the traffic lights shown above. In the MiniALV, however, you can use the parameter `EXCP_GROUP` of the [layout structure \[Seite 57\]](#) to display other icons:

Icons Available in the MiniALV

EXCP_GROUP	1	2	3	4	5	6	7
Icon and internal value	 0	 0	 0	 1	 1	 1	 1
	 1	 1	 1	 2	 2	 2	 2
	 2	 2	 2	 3	 3	 3	 3
	 3	 3	 3	 4	 4	 4	 3
			 4	 5	 5		

Process Flow

- Use type group SLIS:


```
TYPE-POOLS: SLIS.
```
- Define the layout structure of type `SLIS_LAYOUT_ALV`:


```
DATA gs_layout TYPE SLIS_LAYOUT_ALV.
```
- Enhance your output table by a variable of type C as shown below:


```
DATA: BEGIN OF GT_OUTTAB OCCURS 0.
      INCLUDE STRUCTURE <DDIC structure>.
      DATA: light TYPE C. "to display exceptions
      DATA: END OF GT_OUTTAB.
```
- Set the `LIGHTS_FIELDNAME` field of the [layout structure \[Seite 57\]](#) to the field name for the exception:


```
gs_layout-excp_fname = 'LIGHT'.
```
- If, instead of traffic lights, you want to use another icon as exception, set the `EXCP_GROUP` field of the layout structure to the appropriate value (see above).



In the SAP system, independent of this setting, only traffic lights are displayed. The MiniALV, however, displays the correct icons.

6. Use a loop to read each line of the output table and query the fields that are related to the exception. Set your variable for the exception display (in our example: **LIGHT**) to the desired internal value to display the corresponding icon.
7. Pass the layout structure and the output table at the [function module interface \[Seite 41\]](#).

Result

The exception is displayed in the first column of the list. The exception column is always the first.

Displaying Values with Currency/Quantity Units

Use

There are values of output fields that refer to quantities (for example, kilogram) or currencies (for example, Euro). To allow the system to display these values in the format that is appropriate for the respective unit (that is, using the correct number of decimal places), you must assign such fields to a currency or quantity. The field catalog offers three ways to do this:

- Referring the value field to a related currency or quantity unit field
- Explicitly specifying a currency or quantity unit for the entire column
- Manually formatting the value field

Reference to a Currency or Quantity Unit Field

You use the fields **CFIELDNAME** or **QFIELDNAME** (see below) to assign value fields to a currency or quantity unit field. When summing this column, this assignment causes two sums to be calculated and displayed, one for each unit.

For the unit field, there is also an entry in the field catalog. If you don't want the unit to appear as a column in the list or to be included interactively, you can set the field **TECH** to identify the field catalog entry for the unit field as a technical field. This may be of use, for example, if the unit is always unique and is displayed explicitly by the caller in the list header.

For initial value or unit fields, note the following:

Summing and Displaying Value Fields

		Value of the Unit Field	
		<i>Not initial</i>	<i>Initial</i>
Value of the Value Field	<i>Not initial</i>	Display of the decimal places according to the unit in the list and in the total.	For such cells, the unit SPACE is introduced. In the summing display, this unit appears as an individual unit.
	<i>Initial</i>	Display according to the unit as '0' (if the field NO_ZERO of the field catalog is initial). When summing, the system calculates with value '0' and the specified unit.	Display as SPACE . When summing, the value field is ignored.

One Currency or Quantity Unit for the Entire Column

For fields that use the same quantity or currency unit for all values in the entire column, the field catalog contains the fields **CURRENCY** and **QUANTITY** (see below). You use these fields to determine a unit for your value field (for example, **USD** or **KG**). This is reasonable if there is only one quantity or currency unit for the entire column (for example, because it was previously determined by the user in an input field). In this case, the output table does not require an additional field.

Prerequisites

For the value field, note the following:

- The field is of ABAP data type P (see also **INTTYPE** in [Fields of the Field Catalog \[Seite 51\]](#)).
- There is a field in the internal output table that contains the corresponding unit.



Refer to the descriptions of the fields **CURRENCY**, **QUANTITY**, **CFIELDNAME**, and **QFIELDNAME** in [Fields of the Field Catalog \[Seite 51\]](#).

Field Catalog

Definition

The field catalog describes the structure of the data in the output table. The function module interface contains two parameters that refer to the field catalog:

- **IT_FIELDCAT** with fields for formatting, texts, data type, and so on
- **IT_FIELDCAT_HLINKS** with fields for including hyperlinks



See also: [Displaying Lists Using REUSE_MINIALV_LIST_DISPLAY \[Seite 39\]](#).

Structure

The field catalog contains a number of [fields \[Seite 51\]](#) that have no meaning in the MiniALV environment. In addition, there are internally used fields that do not exist in older releases.



You must use only fields that are described in this documentation.

Integration

You pass the field catalog at the [function module interface \[Seite 41\]](#) of module **REUSE_MINIALV_LIST_DISPLAY**.

Structure of the Field Catalog

You need one field catalog for each list that you display using function module `REUSE_MINIALV_LIST_DISPLAY`. However, there are several ways to create this field catalog:

- automatically using a structure of the Data Dictionary (see: [First Steps \[Seite 40\]](#))
- [manually \[Seite 51\]](#) in the ABAP program
- [semi-automatically \[Seite 50\]](#) by combining these two procedures

Generating the Field Catalog Semi-Automatically

Purpose

When generating the field catalog semi-automatically, you combine structure information of the Data Dictionary with your own structure information. In this context, you can:

- modify or
- add structure descriptions of new fields to

the field catalog generated automatically using the DDIC structure. For example, this method is helpful for the following scenarios:



You want to display a Data Dictionary table without displaying all possible columns initially (using field `NO_OUT` of the field catalog).



You want to display additional columns containing icons or other information. See the table in [Generating the Field Catalog Manually \[Seite 51\]](#) for the required fields. This section also explains how to define texts for a new column.

Process Flow

To generate a field catalog semi-automatically, proceed as follows:

1. Use type group `SLIS`.
2. Declare an internal table of type `slis_t_fieldcat_alv`.
3. Call function module `REUSE_ALV_FIELDATALOG_MERGE` and pass the DDIC structure of the output table and the internal table for the field catalog. The function module generates the field catalog and fills the internal table accordingly.
4. Read the lines you want to change and adapt the fields accordingly. If your output table contains more fields than are stored in the Data Dictionary, you must append one line for each new field in the field catalog.

To display the output table with the settings of the field catalog, pass the field catalog in module `REUSE_MINIALV_LIST_DISPLAY`.

Generating a Field Catalog Manually

Purpose

It may be the case that the data you want to display is not at all or only partially represented by a structure in the Data Dictionary. Then you must use the fields of the field catalog to describe the structure of the output table.

Process Flow

The field catalog is defined through table type `slis_t_fieldcat_alv` of type group `SLIS`. Each row of the field catalog table explains a field in your output table. Depending on whether a reference structure exists in the DDIC or not, you must at least fill the following fields of the field catalog structure for each field:

Required Fields of the Field Catalog

Fields of the output table <i>with</i> reference to DDIC	Fields of the output table <i>without</i> reference to DDIC	Description
FIELDNAME	FIELDNAME	Name of the field of the internal output table
REF_TABNAME		Name of the DDIC reference structure
REF_FIELDNAME		Name of the DDIC reference field (required only if other than <code>FIELDNAME</code>)
	INTTYPE	ABAP data type of the field of the internal output table
	OUTPUTLEN	Column width
	SELTEXT_S, SELTEXT_M, SELTEXT_L, and REPTTEXT_DDIC	Column heading, see Texts in the Field Catalog [Seite 56]
	SELTEXT_L	Column description in column selection for layout





Fields of the Field Catalog



The tables below provide an alphabetical overview on the fields of the field catalog that are passed using the interface parameters:

- `IT_FIELDCAT` or
- `IT_FIELDCAT_HLINK`

For more information on the minimum selection of fields required in the field catalog, refer to [Generating the Field Catalog Manually \[Seite 51\]](#).

The meanings of the fields are classified by icons:

- : Reference of the field catalog entry to the output table and to the Data Dictionary
- : Display of values with currency/quantity units
- : Output options of a column
- : Formatting of column contents









-  : Texts
-  : Parameters for fields without DDIC reference









You use the **FIELDNAME** field of the field catalog to create the reference to a field in the output table. Any settings made using fields in the same row of the field catalog refer to the output column specified in **FIELDNAME**.

















You are not allowed to use any fields other than the fields described below.



Alphabetical Index ([Interface Parameter \[Seite 41\]](#) IT_FIELDCAT)

Cl.	Field name	Description	Value range
	CFIELDNAME <i>TYPE</i> <i>SLIS_FIELDNAME</i>	Creates the reference to <i>currency units</i> . The field specified in the FIELDNAME field is linked to the field for currency units specified in the CFIELDNAME field. The field specified in CFIELDNAME must have its own entry in the field catalog. See: Displaying Values With Currency/Quantity Units [Seite 48] .	SPACE, name of a field of the output table
	CHECKBOX(1) <i>TYPE C</i>	Displays checkboxes [Seite 44] in the column.	SPACE, 'X'
	COL_POS <i>LIKE SY-CUCOL</i>	Determines the relative column position of the field in the list display. If you change this value in the field catalog, you must do this consistently for all existing columns.	Natural number
	CURRENCY(5) <i>TYPE C</i>	Explicitly specifies a currency (for example, DEM , USD). The values are displayed for the column specified in FIELDNAME according to the conventions of this currency. See: Displaying Values With Currency/Quantity Units [Seite 48] .	SPACE, name of a currency
	DDICTXT(1) <i>TYPE C</i>	Relevant only for fields with a Data Dictionary reference. Use the values 'L', 'M', 'S' or 'R' to determine whether to use SELTEXT_L , SELTEXT_M , SELTEXT_S , or REPTEXT_DDIC as text for the column selection.	SPACE, 'L', 'M', 'S', and 'R'
	DECIMALS_OUT(6) <i>TYPE C</i>	Relevant only if no currency field is assigned to the field and CURRENCY is not set. The value in this field determines the number of decimal places in the display.	Initial, natural number
	DECIMALSFIELDNAME <i>TYPE</i> <i>SLIS_FIELDNAME</i>	Defines the decimal places line by line. You can use an additional field in the output table to determine the number of decimal places to be displayed in the specified line.	SPACE, field name of the output table
	DO_SUM(1) <i>TYPE C</i>	If this field is set, the system calculates a total through this field. (Corresponds to the generic sum function in the toolbar).	SPACE, 'X'




	EDIT_MASK <i>TYPE</i> <i>SLIS_EDIT_MASK</i>	By setting a conversion exit (for example, conv = '==ALPHA' for function module CONVERSION_EXIT_ALPHA_OUTPUT) you enforce an output conversion to the corresponding output field. (See also documentation for the WRITE formatting option USING EDIT MASK).	SPACE, conv (conversion exit)
	EMPHASIZE(4) <i>TYPE C</i>	Color the column. See: Coloring Columns, Rows or Cells [Seite 45] .	'Cxyz', where x is the color number, y the intensified indicator and z the inverse indicator.
	EXPONENT(3) <i>TYPE C</i>	Determines a fixed exponent for the field. The field must be of internal type F. (See also F1 documentation for the WRITE formatting option EXPONENT).	initial, integer
	FIELDNAME <i>TYPE</i> <i>SLIS_FIELDNAME</i>	Assigns a field name of your output table to a row of the field catalog. Any settings made in this row then refer to the corresponding column of the output table.	Field name of the output table (required)
	ICON(1) <i>TYPE C</i>	If this field is set, the column contents of the output table are displayed as icon. The column contents must consist of valid icon strings (@xx@ or @xx\Q <quickinfo> @). Bear in mind that there may arise problems when printing the icons.	SPACE, 'X'
	INTTYPE <i>LIKE</i> <i>DD03P-INTTYPE</i>	Required only for fields without a Data Dictionary reference.	ABAP data type, see value range of domain INTTYPE
	JUST(1) <i>TYPE C</i>	Relevant only for fields of data types CHAR or NUMC . Alignments: <ul style="list-style-type: none"> • 'R': right-justified • 'L': left-justified • 'C': justified The alignment of the column heading always follows the alignment of the column contents. You cannot align the column heading independently.	SPACE, 'R', 'L', 'C'
	LOWERCASE <i>LIKE</i> <i>DD03P-LOWERCASE</i>	If set, the system recognizes case distinctions in the output table. This influences, for example, the sorting of fields.	SPACE, 'X'

	LZERO(1) <i>TYPE C</i>	Relevant only for fields of data type NUMC , which the system by default displays right-justified without leading zeros. If you set LZERO the system displays leading zeros.	SPACE, 'X'
	NO_OUT(1) <i>TYPE C</i>	If you set this field, you suppress the corresponding column in the list display. The column is still available to the user who can select it interactively as display field. The system displays the contents of suppressed fields in the details window for a line of the list.	SPACE, 'X'
	NO_SIGN(1) <i>TYPE C</i>	Relevant only for value fields. Setting NO-SIGN causes the value to be displayed without a sign (see also F1 documentation for the WRITE formatting options).	SPACE, 'X'
	NO_ZERO(1) <i>TYPE C</i>	If NO_ZERO is set, no zeros are displayed for initial value fields. The cell then remains empty.	SPACE, 'X'
	OUTPUTLEN <i>LIKE</i> <i>DD03P-OUTPUTLEN</i>	Determines the column width of the field: <ul style="list-style-type: none"> • If the field has a Data Dictionary reference, you can leave the field initial. The system in this case copies the output length of the corresponding domain. • For fields without DDIC reference you must determine the desired field output length here. 	0 (default), n
	QFIELDNAME <i>TYPE</i> <i>SLIS_FIELDNAME</i>	Creates a reference to the <i>quantity units</i> . The field specified in the FIELDNAME field is linked to the field for quantity units specified in QFIELDNAME . The field specified in QFIELDNAME must have its own entry in the field catalog. See: Displaying Values With Currency/Quantity Units [Seite 48] .	SPACE, name of a field of the output table
	QUANTITY(3) <i>TYPE C</i>	Explicitly specifies a unit (for example, KG). The values are displayed for the column specified in FIELDNAME according to the conventions of this unit. See: Displaying Values With Currency/Quantity Units [Seite 48] .	SPACE, name of a quantity unit

	REF_FIELDNAME <i>LIKE</i> <i>DD03P-FIELDNAME</i>	<p>You must fill this field if</p> <ul style="list-style-type: none"> the field of the output table described by the current entry in the field catalog has a corresponding field in the Data Dictionary and the field name in the output table differs from the field name of the field in the Data Dictionary. <p>If the field names are identical, it is sufficient to specify the DDIC structure or table in the REF_TABNAME field of the field catalog.</p>	SPACE, name of a field in the Data Dictionary that corresponds to a field of the output table
	REF_TABNAME <i>LIKE</i> <i>DD03P-TABNAME</i>	<p>You must fill this field only if the field of the output table described in the current entry in the field catalog has a corresponding field in the Data Dictionary. Using this allocation, REUSE_MINIALV_LIST_DISPLAY can, for example, copy the text for the column heading from the Dictionary.</p>	SPACE, name of a structure or table from the Data Dictionary that corresponds to a field of the output table
	REPTTEXT_DDIC <i>LIKE</i> <i>DD03P-REPTTEXT</i>	<p>Relevant only for fields with a Data Dictionary reference. For such fields, the system copies the field label for the heading of the corresponding data element into this field. See: Texts in the Field Catalog [Seite 56].</p>	Text copied from the Data Dictionary
	ROUND <i>TYPE I</i>	<p>Output of a value of type P scaled by the power of ten. For positive values of ROUND the ALV Grid Control shifts the comma to the left, otherwise to the right.</p> <p>(See also F1 documentation for the WRITE formatting options)</p>	initial, natural number
	ROUNDFIELDNAME <i>TYPE</i> <i>SLIS_FIELDNAME</i>	<p>Defines the scaled output line by line. You can use an additional field of the output table to determine how the corresponding field is scaled in each line.</p>	Name of a field of the output table
	SELTEXT_L <i>LIKE</i> <i>DD03P-SCRTEXT_L</i>	<p>Relevant only for fields with a Data Dictionary reference. For such fields, the system copies the long field label of the corresponding data element into this field. See: Texts in the Field Catalog [Seite 56].</p>	Text copied from the Data Dictionary
	SELTEXT_M <i>LIKE</i> <i>DD03P-SCRTEXT_M</i>	<p>Relevant only for fields with a Data Dictionary reference. For such fields, the system copies the middle field label of the corresponding data element into this field. See: Texts in the Field Catalog [Seite 56].</p>	Text copied from the Data Dictionary

	SELTEXT_S <i>LIKE</i> <i>DD03P-SCRTEXT_S</i>	Relevant only for fields with a Data Dictionary reference. For such fields, the system copies the short field label of the corresponding data element into this field. See: Texts in the Field Catalog [Seite 56] .	Text copied from the Data Dictionary
	TECH(1) <i>TYPE C</i>	If set, the corresponding field is not displayed in the list and cannot be displayed by interactive selection. The field is known only in the field catalog. (For example, you cannot specify it as a sort criterion).	SPACE, 'X'

Alphabetical Index ([Interface Parameter \[Seite 41\]](#) IT_FIELDCAT_HLINK)

Cl.	Field name (Type)	Description	Value range
	FIELDNAME <i>TYPE</i> <i>LVC_FNAME_WAO</i>	Assigns a field name of your output table to a row of the field catalog. Any settings made in this row refer to the corresponding column of the output table.	Field name of the output table (required)
	HREF_HNDL <i>TYPE INT4</i>	Handle to which a URL is assigned. In the list, all cells of the column are displayed as hyperlinks. You must maintain the target addresses of the hyperlinks in a table with reference structure <i>LVC_S_HYPE_WAO</i> and pass them with the table parameter IT_HLINKS . (For the table structure, refer to Defining Hyperlinks [Seite 43]).	Natural number
	WEB_FIELD <i>TYPE</i> <i>LVC_FNAME_WAO</i>	Defines hyperlinks [Seite 43] at cell level.	Name of a field of the output table

Texts in the Field Catalog

A list displayed using function module **REUSE_MINIALV_LIST_DISPLAY** contains several texts for each column:

- the column heading,
- the popup column selection (for example, in the current layout)

For fields with Data Dictionary reference, the system automatically adopts the field labels of the corresponding data elements. The texts for the short, medium, and long field labels as well as for the heading are copied into the fields **SELTEXT_S**, **SELTEXT_M**, **SELTEXT_L**, and **REPTEXT_DDIC** of the field catalog.

There are several possibilities how the ALV Grid Control assigns text to a column:

- If no other text is specified, the text copied from the Data Dictionary is used. The ALV Grid Control chooses the long field label (**SELTEXT_L**) as text in the popup column selection. The column heading is dynamically adjusted to the column width even if the user changes the column width. The ALV Grid Control always uses the longest DDIC text that still fits into the column.

- If field `DDICTXT` of the field catalog is not empty, the field determines the field label that is to be displayed as the column heading. The value range of this field is as follows: `SPACE`, `'L'`, `'M'`, `'S'`, and `'R'`.

Layout Structure

Use

The layout structure of function module `REUSE_MINIALV_LIST_DISPLAY` is used to

- Display [checkboxes \[Seite 44\]](#) (`BOX_FIELDNAME`)
- Display [exceptions \[Seite 47\]](#) (`LIGHTS_FIELDNAME`, `EXCP_GROUP`)
- Color rows or cells (`INFO_FIELDNAME`, `COLTAB_FIELDNAME`)

All other fields have no meaning in the MiniALV environment. You are not allowed to use them.

Structures

There are two structures for passing the parameters of the layout structure to the [function module interface \[Seite 41\]](#):



The term *layout structure* in this documentation always refers to both structures.

Parameter `IS_LAYOUT` (structure type `SLIS_S_LAYOUT_ALV`)

Field name	Description	Value range
<code>BOX_FIELDNAME</code> <i>TYPE</i> <code>SLIS_FIELDNAME</code>	Field name of a field of the output table, in which you want to display checkboxes [Seite 44] .	Name of a field in the output table
<code>INFO_FIELDNAME</code> <i>TYPE</i> <code>SLIS_FIELDNAME</code>	You use this table to declare an additional field in your output table that is used to color rows [Seite 45] in the table.	Name of a field in the output table
<code>COLTAB_FIELDNAME</code> <i>TYPE</i> <code>SLIS_FIELDNAME</code>	You use this table to declare an additional field in your output table that is used to color rows/cells [Seite 45] in the table.	Name of a field in the output table
<code>LIGHTS_FIELDNAME</code> <i>TYPE</i> <code>SLIS_FIELDNAME</code>	Field name of a field of the output table, in which you want to display exceptions [Seite 47] .	Name of a field in the output table

Parameter `IS_LAYOUT2` (structure type `LVC_S_LAY2_WAO`)

Field name	Description	Value range
<code>EXCP_GROUP</code> <i>TYPE CHAR1</i>	Type of exception ('1': traffic lights, '2': LEDs, '3'-'7': others.; see: Using Exceptions [Seite 47]).	'1'-'7'