



Contents:

- Overview on POWER List concept
- Designing new POWER Lists
- Implementing new POWER Lists

- The topic of POWER Lists is divided into two chapters. In the first chapter, we will see an introduction into the POWER Lists. The second chapter will go into more detail, in specific areas like reporting with POWER Lists.

POWER Lists (Part I): Unit Objectives

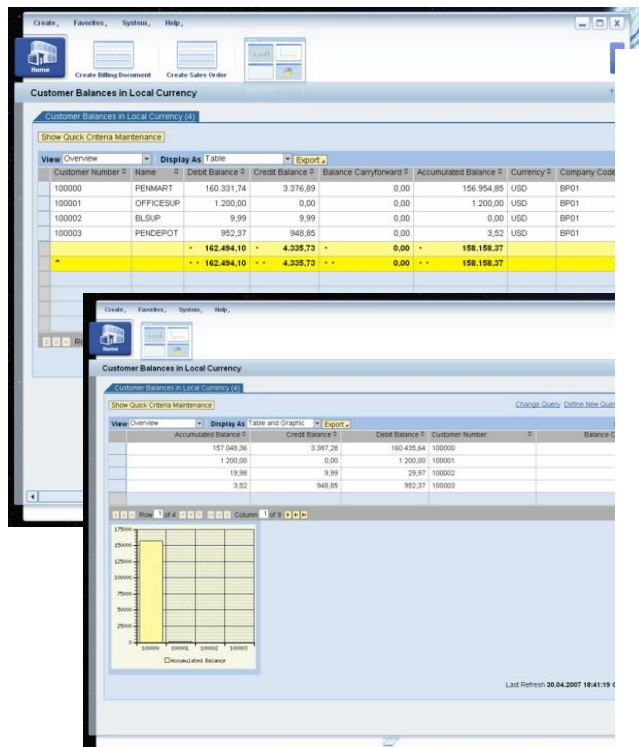


At the conclusion of this unit, you will be able to:

- Describe the concept of POWER Lists
- Define your own POWER Lists including an efficient concept for data queries, results output and target definition
- Implement your own POWER Lists in a SAP All-In-One system

- This chapter will guide you through the main areas of the POWER Lists, describing the concept of POWER Lists, explaining how POWER Lists can be developed and finally how they can be implemented.
- The focus of the present chapter is on how to build new POWER Lists rather than changing or copying existing POWER Lists delivered by SAP. At a later stage, you might leverage from existing POWER Lists, changing them to your needs. In this way you will reduce the number of steps you are required to do carry out. Nevertheless, if you discover problems such as error messages you will have to be prepared and you will need to understand entirely the concept behind the POWER Lists.
- As a result, we're going to go through the whole concept, developing a POWER List from scratch. Finally, you will have touched all involved elements in the system which will help you to follow up with errors in the future.

Power Lists at a Glance



List of specific business objects

Homogenous user interface

Elementary enabler for OBN

Personalizable by every end-user

Pre-definitions by SAP and Partners

Simplifies business processes

Web Dynpro for ABAP based framework

Developers need pure ABAP Knowledge

© SAP AG 2007, SAP All-in-One: UE Workshop 3

THE BEST-RUN BUSINESSES RUN SAP™



- The POWER List is the key element to enable the business object oriented navigation in SAP All-in-One. The POWER List – standing for Personal Object Work Entity Repository List – is basically a framework which can list business objects and allows specific activities (actions) based on these business objects.
- Summarizing the most important advantages we can say that POWER Lists offer:
 - A central point of access related to business objects (e.g. all types of sales orders), with one standard user interface for business objects AND reports. No need to educate end users in different UIs.
 - Quick access for users and comprehensive overview of the work environment and all related business objects. Enables users to start their daily work depending on the operation queue. = Enhanced user productivity, ensures sound decision making
 - Optionally, less steps of user interaction through automated process steps (e.g. pressing a button on POWER List to accept invoices instead of entering any transaction). Direct guidance to a specific transaction with parameter transfer (Object Based Navigation – OBN)
 - Personalization by each user or alternatively by an administrator
 - Pre-delivered POWER Lists with pre-settings from SAP and Partners
 - High performance through intelligent caching mechanism. POWER Lists remain available until results are refreshed
 - Finally, it should be known, that POWER Lists are developed with Web Dynpro for ABAP. With this, it becomes clear that existing ABAP knowledge can be leveraged. Furthermore, there is no need for a Java Engine running in the backend system.

The POWER List

Annotations:

- User can easily access all queries (pre-delivered and user specific) via tab navigation**: Points to the tab strip at the top showing 'Billing Due List (0)' and 'Billing Documents (92)'.
- Application related buttons provide object based functionality**: Points to buttons like 'Show Invoice as PDF', 'Print Invoices', and 'Cancel Invoice'.
- Queries can be created, changed, categorized or temporarily hidden as needed**: Points to the 'Change Query', 'Define New Query', and 'Personalize' links.
- Selection criteria of a query can be changed temporarily (as long as POWER list is open) and quick without changing the whole query.**: Points to the 'Search Criteria' field.
- With the given bottom bar, the user can navigate through the data table**: Points to the row and column navigation controls at the bottom.
- A filter can be switched on to allow filtering within the data table**: Points to the 'Filter Settings' button.
- Further settings can be made to personalize the data table (e.g. shown columns, number of rows, sorting)**: Points to the 'Filter Settings' button.

Query Area: Indicated by a bracket on the right, it encompasses the top section including the tab strip, search criteria, and query management links.

List Area: Indicated by a bracket on the right, it encompasses the data table and the bottom navigation bar.

Billing Document	Sold-To Party	Name of Sold-To Party	Billing Date	Net Value	Document Currency
90000091	100002	R2R Supply	10.01.2007	37,62	USD
90000090	100000	Walmart	16.01.2007	37,62	USD
90000089	100000	Walmart	01.03.2007	12,54	USD
90000088	100002	R2R Supply	21.01.2007	50,16	USD
90000087	100000	Walmart	18.01.2007	25,08	USD

Row 1 of 92 | Column 1 of 6 | Last Refresh 30.03.2007 15:42:02 CET Refresh

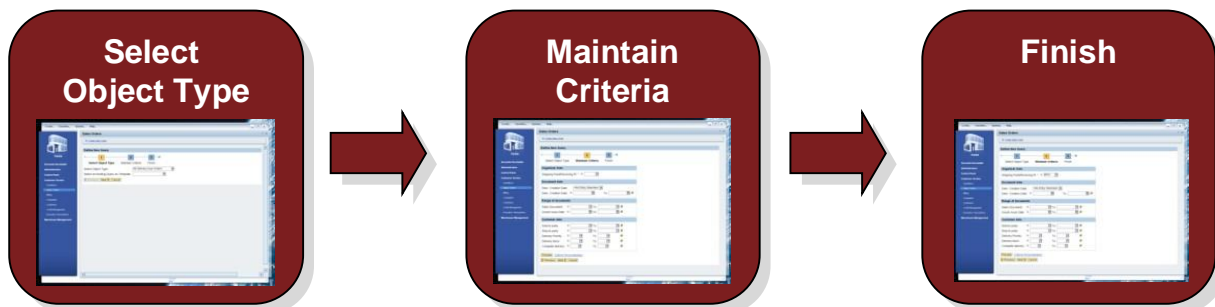
© SAP AG 2007, SAP All-in-One: UE Workshop 4

THE BEST-RUN BUSINESSES RUN SAP™



- As said before, the POWER List offers a selection of business objects such as invoices, stock items and so on. Technically, these objects are selected on the database in the backend system. The data retrieval in the backend can be either a simple DB or a complex data selection, using function modules and more. In all cases the output of these selections result in a list.
- Looking closer at the POWER List, we can divide the list into two pieces. The “query area” and the “list area”.
- The query area is the place where the user can toggle between existing queries (tab strips), change temporarily some settings for search criteria of a current query or create and change new queries. Additionally, a personalization capability allows the user to personalize the query area to his needs.
- The list area offers the results of the specific query (tab strip). In other words, it lists business objects and based on these objects allows several actions to be initiated.
- In the list area, the user finds several buttons on top of the list. Some of them are standard functionalities like export to MS Excel. Standard however, does not mean that these buttons are visible; whether such a button is visible or not depends on the developer of the POWER List.
- Another functionality of the list area is the availability of buttons: on the top of the list, more buttons could be available. Again this depends on how the POWER List was developed and naturally also depends on what makes sense in relation to the listed business objects. These buttons can reach from starting a transaction with the selected business object as parameter, to a scenario where clicking the button initiated several process steps in the background (e.g. accept invoices)
- The list area also allows personalization. Furthermore, it offers filtering capabilities and the option to save the user specific settings.

Definition of POWER List Queries



© SAP AG 2007, SAP All-in-One: UE Workshop 5

THE BEST-RUN BUSINESSES RUN SAP™



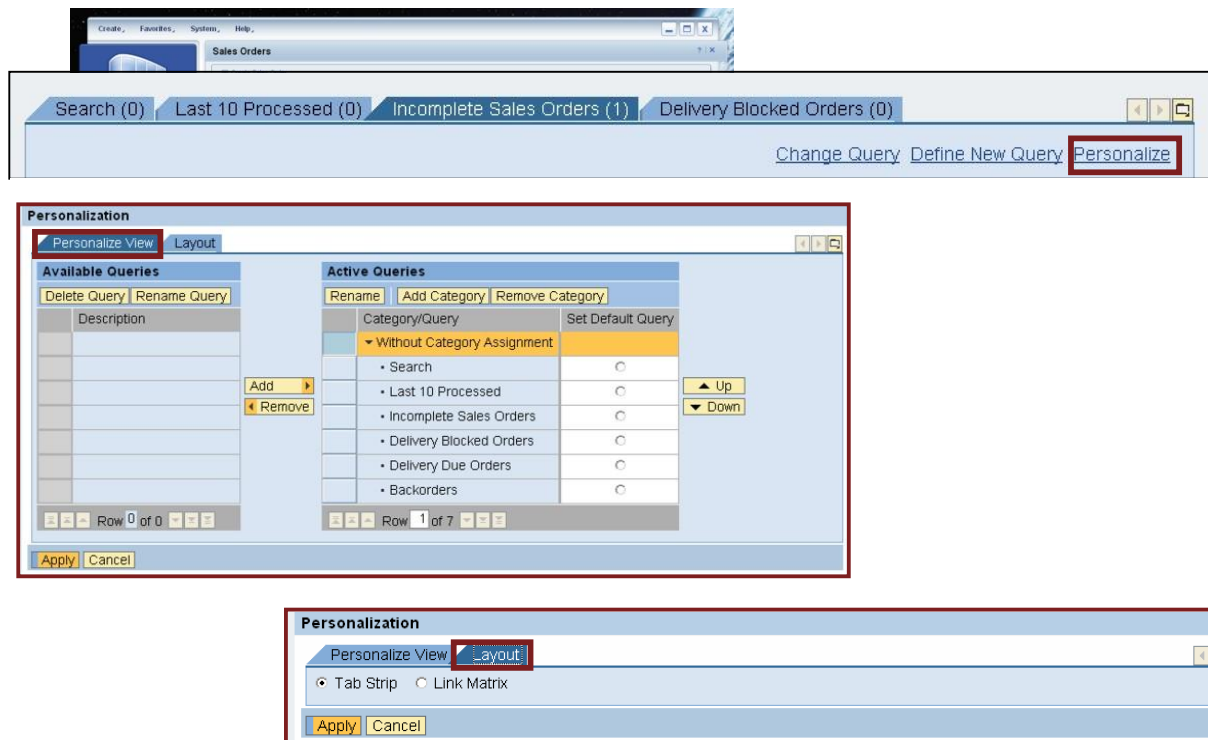
- As explained in the previous slide, one area in the POWER List is specifically designed for queries (data selection).
- Creating a new query always follows the same flow
 - Select Object Type

For sure, a user cannot select the data directly from the database as he would like to do (e.g. full select on a table MARA). Instead, the developer of the POWER List has pre-defined several possible selections. This pre-definition is named “type”. With the selection of an object type, the user does nothing else but choosing one of the pre-defined types of the POWER List.
 - Maintain Criteria

As with the types, each POWER List can provide a set of criteria, to allow the user reducing the amount of result data and to show exactly those business objects later in the list which are relevant to the user. Again, this is based on what the developer of the POWER List has defined.
 - Finish

Finally, the user can name a new query to find it easily on the tab strip.

Personalization of POWER List Queries



- Besides creating own queries, the user also has the option of personalizing the query area. Basically, this includes two things:
- Firstly, a user can set which queries should be shown (active) and which should be still kept but not shown for the moment (inactive). Once a query is inactive, it can also be deleted. If located at the same place, queries can be categorized if needed.
- Secondly, the personalization allows to switch from the tab strip view to a link matrix view. The link matrix is the view which shows the possible categorizations made by the user. The tab strip on the other hand, represents more a flat sequence of queries.

Settings of POWER List Queries

The screenshot shows the SAP POWER List interface with a table of sales documents. A 'Settings' button is highlighted in the top right. Below it, the 'Settings' dialog box is open, showing five tabs: Column Selection, Sort, Calculation, Filter, and Display. Red lines connect each tab to a descriptive text box:

- Column Selection** Hides/un-hides specific columns
- Sort** Defines columns to be sorted
- Calculation** Sets and activates calculations (e.g. total, min, max)
- Filter Settings** Defines static filter
- DISPLAY Settings** Defines POWER List table/graphic

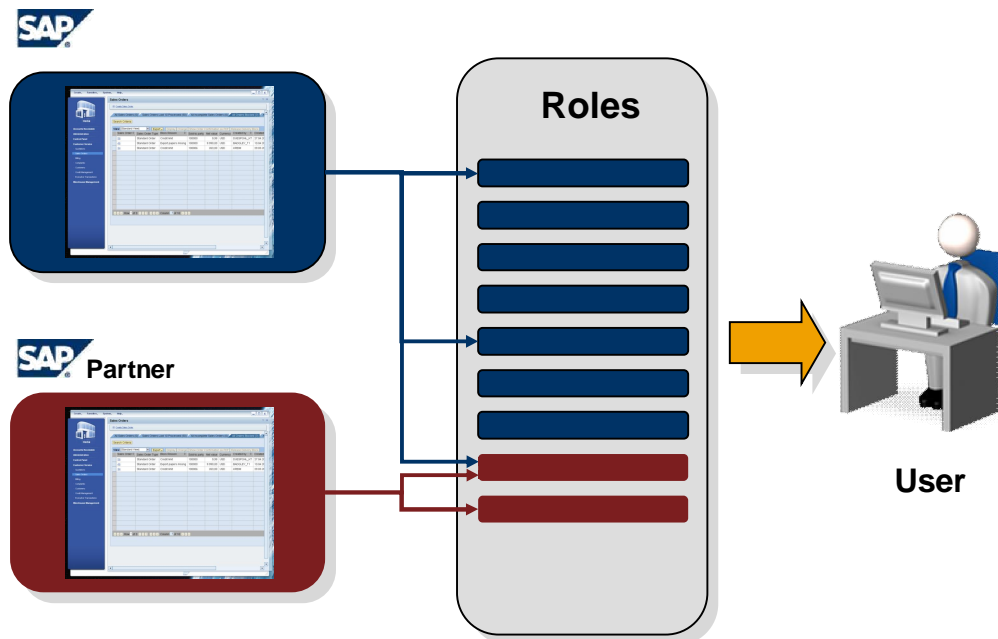
© SAP AG 2007, SAP All-in-One: UE Workshop 7

THE BEST-RUN BUSINESSES RUN SAP™



- The second area of the POWER Lists is – as mentioned before – the list area.
- Here, the user also has the chance to personalize. This includes:
 - **Column Selection**
Here, columns can be removed and sorted the way the user wants them to be.
 - **Sort**
In this tab, a sorting mechanism can be activated based on selected columns.
 - **Calculation**
This personalization allows to put calculations into the list like total amount, maximal or minimal amount.
 - **Filter Settings**
As a POWER List user, you can either do temporary filtering using the link on the top of the list or you can define a filter which remains set. A filter can be “show of all items starting with A”.
 - **Display**
In this tab, the look of the list table can be personalized. If the POWER Lists represents a report, you have the option to turn on the graphic mode. Otherwise, the personalization just offers table settings like numbers of rows.

Pre-defined POWER Lists



Pre-Defined POWER Lists include:

- Default Queries
- Default Settings
- Default Personalization

© SAP AG 2007, SAP All-in-One: UE Workshop 8

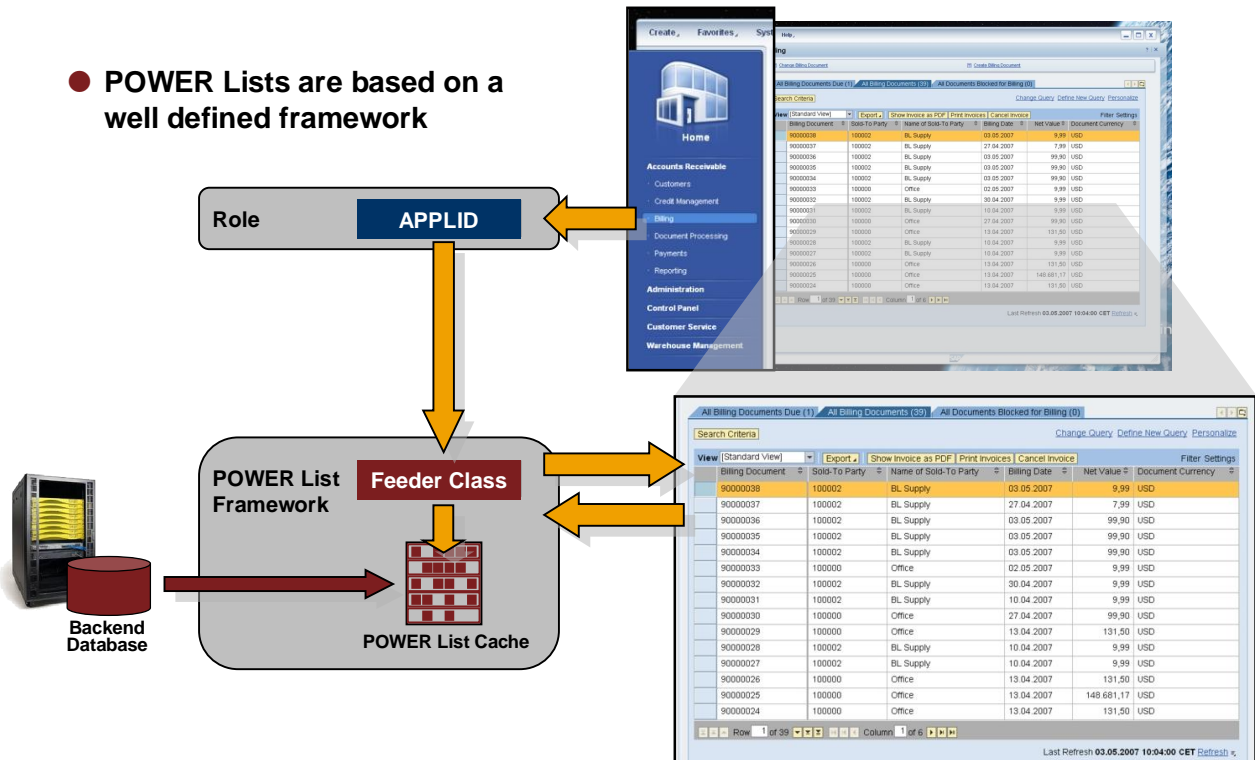
THE BEST-RUN BUSINESSES RUN SAP™



- SAP ships pre-defined POWER Lists with SAP All-in-One. These lists can be used as templates to define other POWER Lists, or to slightly modify them. A customer could also use them directly without modification. However, this needs to be checked in every case, as the business objects in the system for sure depend on the customizing settings of the customer. In some cases, a pre-defined POWER List cannot be used as it is, because of these customer specific settings and needs to be adjusted.
- In all cases, SAP recommends to copy the pre-defined POWER Lists into customer namespace (Z* or Y*) to avoid conflicts in later system upgrades.
- For sure, also SAP Partners can pre-define POWER Lists for their customers. They can leverage from the SAP Power Lists or they can develop completely new POWER Lists from scratch.
- A pre-defined POWER List is at least developed with 1 to n query types and a selection criteria view.
- Additional, POWER Lists can be delivered with pre-settings for default queries, default selection criteria and even personalization.
- As the connection between the user and the POWER Lists is done via the roles, it is possible to arrange several different POWER Lists from SAP and/or Partners into one or several roles. In other words, the concept is pretty open for any kind of arrangement a partner or customer would like to have in front of the user.

The POWER List: Technical Background

- **POWER Lists are based on a well defined framework**

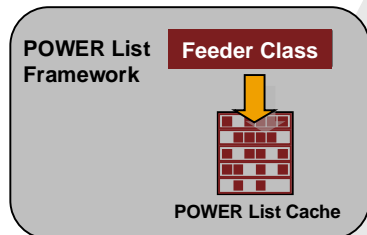


© SAP AG 2007, SAP All-in-One: UE Workshop 9

THE BEST-RUN BUSINESSES RUN SAP™



- From a technical perspective, the POWER Lists are well defined through a complete development framework.
- The roles are the access point to all the POWER Lists in the system. While exceptions may apply, in most cases, POWER Lists are launched as “homepages” within the canvas area of the SAP NetWeaver Business Client while having the navigation panel on the left side. As already explained in the role chapter, homepages can be set on each folder in a navigation structure. Clicking such a folder starts the POWER List on the right.
- What happens technically behind the scenes while clicking a folder? Well, homepages in their roles can be defined in several ways, however if it should launch a POWER List, then the folder item in the role holds the information that a POWER List (POWL Application) with a specific Application ID (APPLID) needs to be called.
- The APPLID which needs to be called is, as said, defined in the role. Nevertheless, it seems quite logical that somewhere in the backend system an APPLID needs to be defined first. In the next slides you will learn in more detail how the APPLID is built and what steps you need to take to make such a APPLID available for a call from a role.
- At this point, the interesting part is the so called “feeder class”. The feeder class is the piece which feeds the POWER List with the appropriate data. We will see this in detail in some seconds.
- The feeder class communicates with the database selecting specific data, forwards the data to a POWER List internal cache and refreshes the POWER List on the users client on demand.
- Additionally, the feeder class also includes the handling of actions initiated by the user while pressing a button.

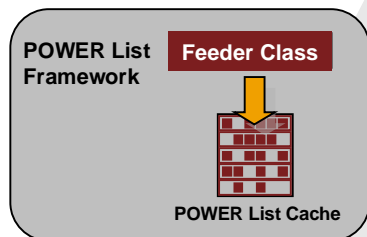


Mandatory Steps

- 1 GET_OBJECT_DEFINITION**
Define a container to store data provided by data retrievals
- 2 GET_OBJECTS**
Define data retrieval (e.g. database select, function module)

- The feeder class can be seen as the central and most important place while developing or modifying POWER Lists.
- NOTE: For an easier explanation, we have talked about developing your own POWER List. But now it should become clear that we're more talking about developing our own feeder class.
- Developing a feeder class doesn't mean that you need to build all coding from scratch. As mentioned, this all relies on a well defined framework. Creating a new feeder class means that you develop based on a pre-defined interface which offers you all the methods you need, to operate the POWER List.
- Not all methods provided by the POWER List interface need to be used from the start. You can start developing a simple feeder, which only shows business objects. From there, you can then improve your feeder step by step. All programming at this point requires ABAP and OO ABAP experience.
- The mandatory steps to take, to develop a simple feeder (to feed your first own POWER List) are:
 - Define data container (Method "GET_OBJECT_DEFINITION")
This method is used to define the container (e.g. specify field types) where the selected data gets stored. Caching and other mechanisms of the POWER Lists technology will be handled automatically in the background based on these settings. So there is no need to explicitly take care on things like caching data and so on.
 - Retrieve data from the backend system (Method "GET_OBJECTS")
Here you need to define the data retrieval itself. This can be either a very simple database select (e.g. select * from xyz) or a complex selection where you use existing SAP function modules or your own coding.
- With maintaining only these two methods with coding, you will already have your first developed feeder at hand.

How to Create a POWER List: Concept

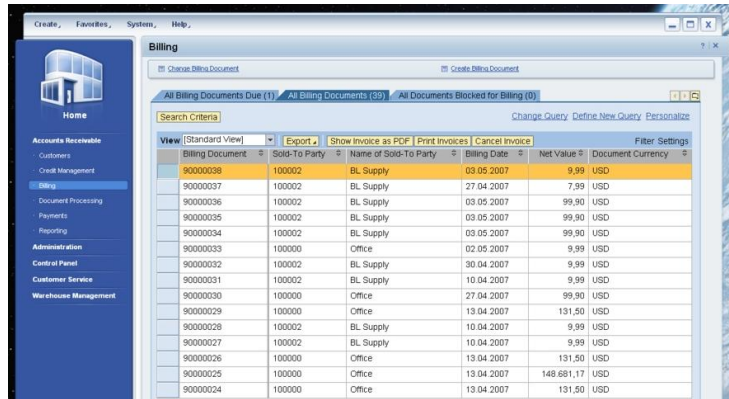


Optional Steps

- 3 GET_SELECTION_CRITERIA**
 Define selection criteria that can be used to define new queries against your Feeder
- 4 GET_FIELD_CATALOG**
 Define the field catalog to be used for query results table display
- 5 GET_ACTIONS & HANDLE_ACTIONS**
 Define buttons and their explicit actions
- 6 GET_ACTION_CONF**
 Define confirmation messages for explicit actions
- 7 GET_DETAIL_COMP**
 Define detail component area which can be used to improve visualization of selected data

- Nevertheless, having a POWER Lists which shows some business objects only isn't really that cool. So as next, you might think of enhancing your feeder. This can be done with the methods above
- In detail, these methods enable you to:
 - Define selection criteria (Method "GET_SELECTION_CRITERIA")
 With this method you can define, which selection criteria is visible and selectable by the user. Example: You have a POWER List showing billing documents. You could offer the selection criteria "billing date" so that the user can later retrieve the data directly the way he searches for it.
 - Define the field catalog (Method "GET_FIELD_CATALOG")
 This method defines the field catalog to be used for query results table display.
 - Define buttons and their actions (Methods "GET_ACTIONS" & "HANDLE_ACTIONS")
 By maintaining the two methods GET_ACTIONS and HANDLE_ACTIONS, you have a huge variety of options to improve the POWER Lists significantly. First you need to define the buttons with name, index and more (GET_ACTIONS). Second you need to define the actions which should be initiated if the user presses such a button. The action can simply be launching a transaction and forwarding the business object parameters to it. Or it could be used to simplify a whole process by using the buttons to call several function modules in a sequence automating the process in the background based on the selected item(s) in the POWER List.
 - Define a confirmation pop-up (Method "GET_ACTION_CONF")
 Using this method allows you to throw a pop-up with some information like a confirmation. Think of a business object you can delete via a button in the POWER list. A confirmation pop-up could ask whether the user is sure to delete this object.
 - Enable the detail component feature (Method "GET_DETAIL_COMP")
 Finally, this method can be used in case you want to show a detailed view of a specific business object below the POWER List. This could be helpful if you have large data sets where a horizontal scrolling is too time consuming or not quite usable. In this case, the detailed component offers a good alternative as it provides a detailed view area below the list, where you can show all the different fields without the need of horizontal scrolling.

How to Create a POWER List: Concept



The screenshot shows the SAP Billing interface. On the left is a navigation menu with options like Home, Accounts Receivable, Customers, Credit Management, Billing, Document Processing, Payments, Reporting, Administration, Control Panel, Customer Service, and Warehouse Management. The main area displays a table of billing documents with columns: Billing Document, Sold-To Party, Name of Sold-To Party, Billing Date, Net Value, and Document Currency. The table contains 15 rows of data.

Billing Document	Sold-To Party	Name of Sold-To Party	Billing Date	Net Value	Document Currency
90000038	100002	BL Supply	03.05.2007	9.99	USD
90000037	100002	BL Supply	27.04.2007	7.99	USD
90000036	100002	BL Supply	03.05.2007	99.90	USD
90000035	100002	BL Supply	03.05.2007	99.90	USD
90000034	100002	BL Supply	03.05.2007	99.90	USD
90000033	100000	Office	02.05.2007	9.99	USD
90000032	100002	BL Supply	30.04.2007	9.99	USD
90000031	100002	BL Supply	10.04.2007	9.99	USD
90000030	100000	Office	27.04.2007	99.90	USD
90000029	100000	Office	13.04.2007	131.50	USD
90000028	100002	BL Supply	10.04.2007	9.99	USD
90000027	100002	BL Supply	10.04.2007	9.99	USD
90000026	100000	Office	13.04.2007	131.50	USD
90000025	100000	Office	13.04.2007	148.681,17	USD
90000024	100000	Office	13.04.2007	131.50	USD

DESIGN

DEVELOP

TEST

GO
PRODUCTIVE

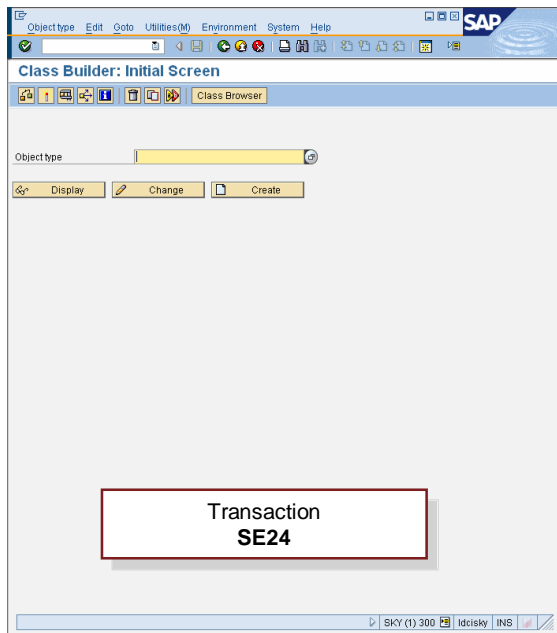
© SAP AG 2007, SAP All-in-One: UE Workshop 12

THE BEST-RUN BUSINESSES RUN SAP™



- Before starting developing a feeder (and a new POWER Lists) we should shortly underline another aspect of the whole concept.
- Developing a POWER Lists doesn't mean to develop some feeder coding, only. Developing the coding is one task of a sequence you need to do.
- The most important step to highlight is the design phase. Be aware, that a good design upfront can not only speed up the process of developing the coding but also to increase the efficiency of the final POWER List and the way the user can work with it.
- For a good design you should at least have a draft available for:
 - The data you want to select. Most common questions: Where can find the data? What are the data types? Which function modules can be used? Do I need to develop own function modules.
 - The selection criteria you want to offer. Most common questions: Which selection make sense for the user? Are there performance impacts to be considered?
 - The buttons you want to include. Most common questions: What are the actions? Button names? Can I use function modules? Will the buttons launch transactions?
 - Detailed component. Most common questions: Do I need the detailed component? Which data do I need to show up in the detailed view?
- After developing the coding, you also need to test your POWER List. Make sure, especially for the pre-defined setting you might add to your feeder, that you are testing in a real environment, not only with real data, but also with a user which doesn't hold SAP_ALL or similar authorization.

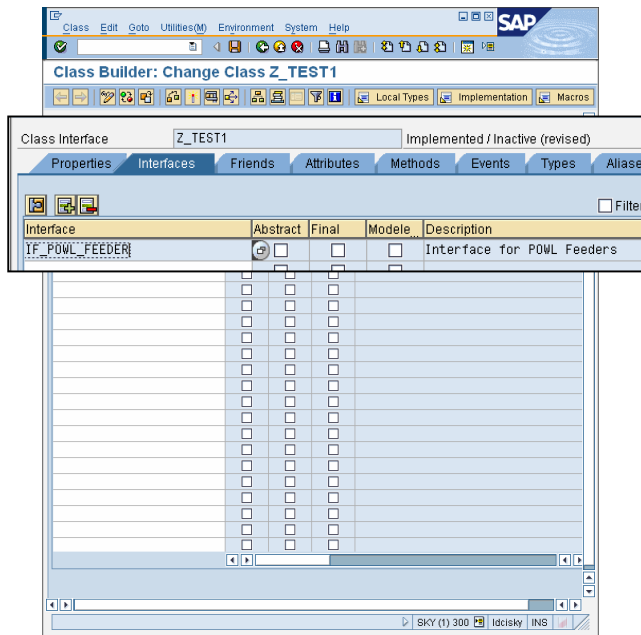
How to Create a POWER List: Technical Flow



- With transaction SE24, a new POWER List can be developed based on the POWER List framework
- Transaction SE24
 - <POWER List Name>
 - “Create”
 - “Ok”

- After learning what a POWER List is and what the technical concept of the POWER Lists framework is, we are now ready to start developing our first feeder.
- Again, make sure you have a good design at hand. For the hands-on in the workshop, you will get a design by your instructor.
- The transaction where we going to build this feeder is SE24.
- Specify the name of your new feeder using the prefix “Z” or “Y” to ensure your feeder is developed in the customer namespace.
- Press the “create” button and simply confirm the upcoming pop-up with “ok”.

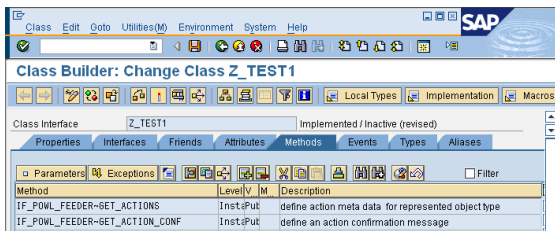
How to Create a POWER List: Technical Flow



- The interface “IF_POWL_FEEDER” provides all necessary methods to start building a new POWER List

- When you have entered the transaction, you will see the class builder in front of you. Normally, you would now start developing a class. As mentioned earlier, SAP provides an interface which delivers all the methods you need to develop the feeder.
- Therefore, switch to the tab “Interfaces” and enter the interface name “IF_POWL_FEEDER”. Don’t forget to press enter.
- The interface should be accepted and the methods are uploaded into the class builder.

How to Create a POWER List: Technical Flow



- The tab “methods” offers all “IF_POWL_FEEDER”-related methods

Class Interface

Z_TEST1

Implemented / Inactive (revised)

Properties

Interfaces

Friends

Attributes

Methods

Events

Types

Aliases

Parameters

Exceptions

</

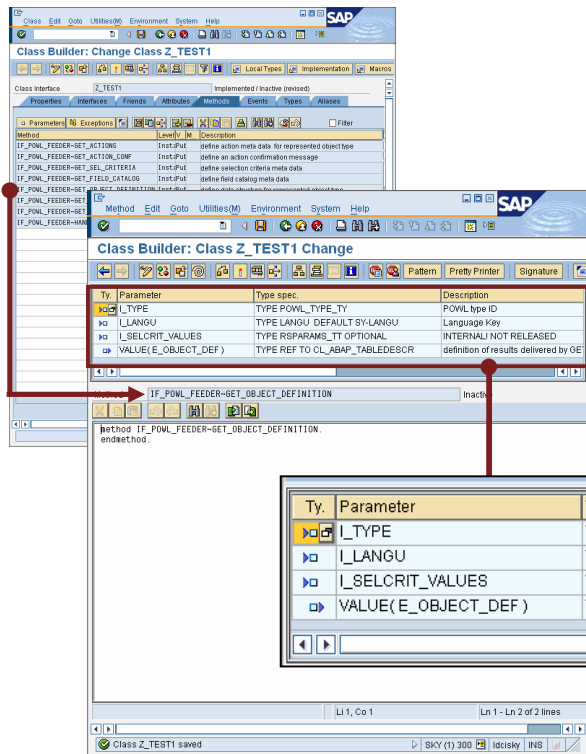
© SAP AG 2007, SAP All-in-One: UE Workshop 15

THE BEST-RUN BUSINESSES RUN SAP™



- In tab “Methods” you will now see all the methods which were uploaded with the IF_POWL_FEEDER interface.
- You find all the methods explained earlier in this chapter. Remember: Two methods are mandatory: “GET_OBJECT_DEFINITION” (1) & “GET_OBJECTS” (2)
- The methods are optional and can be developed as needed.

How to Create a POWER List: Technical Flow

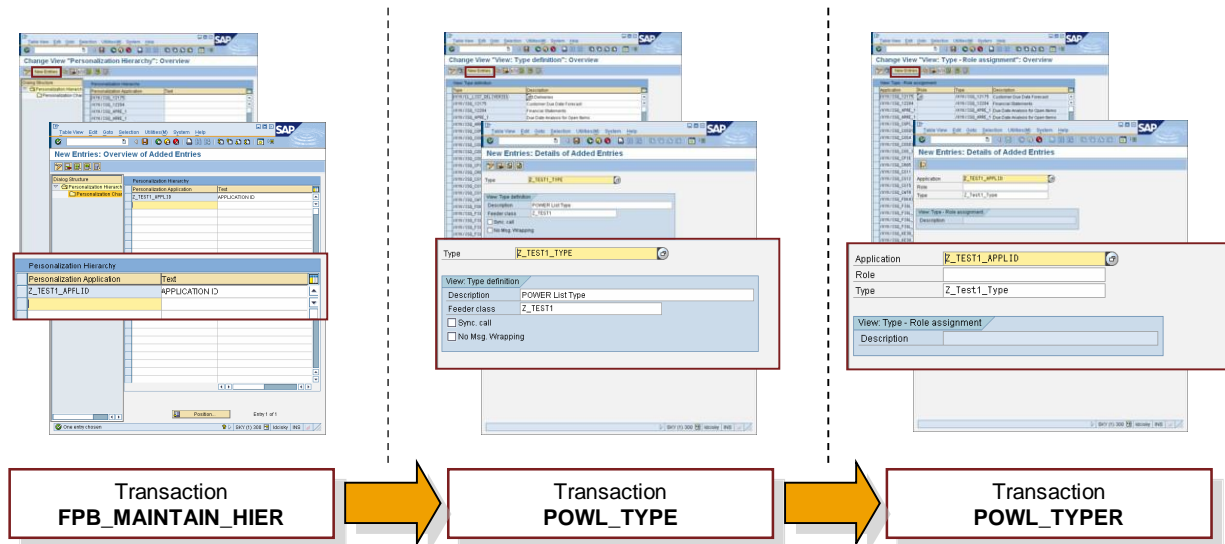


- Double click on specific method to open up the editor
- Top frame shows up all possible parameters provided by the method

Ty.	Parameter	Type spec.	Description
PO	I_TYPE	TYPE POWL_TYPE_TY	POWL type ID
PO	I_LANGU	TYPE LANGU DEFAULT SY-LANGU	Language Key
PO	I_SELSCRIT_VALUES	TYPE RSPARAMS_TT OPTIONAL	INTERNAL! NOT RELEASED
PO	VALUE(E_OBJECT_DEF)	TYPE REF TO CL_ABAP_TABLEDESCR	definition of results delivered by GE

- If you enter a method by double-clicking on it, a new window will open up. In this window, the method specific coding takes place.
- At the top of the screen you will notice a frame which shows all possible parameters which are defined for the method you are currently in.
- From this point, you should be able to put in some coding to the methods to create your first feeder.
- Your instructor will give you more information on what you can develop during the workshop.

Registering a POWER List & Make it Visible



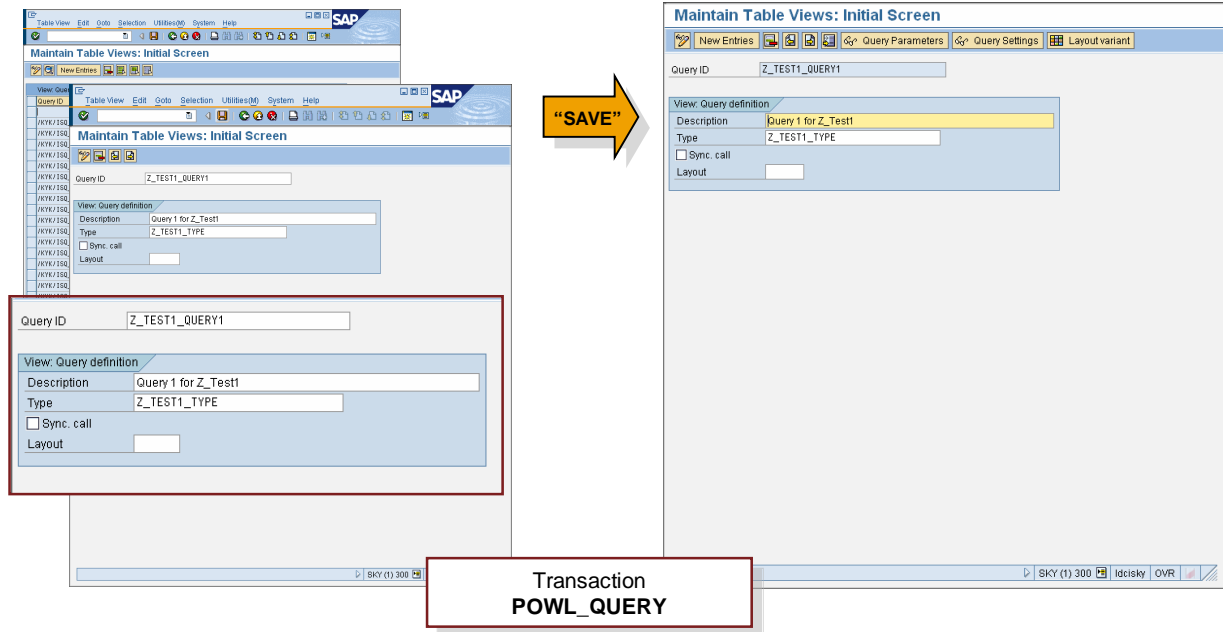
© SAP AG 2007, SAP All-in-One: UE Workshop 17

THE BEST-RUN BUSINESSES RUN SAP™



- After a feeder is developed, it needs to be made visible to the roles. Basically, this means we need to register the feeder under a specific APPLID, define a POWER List type and introduce it to the roles.
 - Transaction FBP_HIER_MAINTAIN
First of all, you need to specify the so called APPLID (Application ID). This ID will later be used in the role to specify the target (your feeder) which will then be shown as POWER List homepage in the SAP Net Weaver Business Client. The APPLID is more or less just a name to specify.
 - Transaction POWL_TYPE
As next step, you need to specify the POWER List type. In the first slides of this chapter we have seen, that a POWER List provides 1 to n types of object types a user can select from. At this point it becomes clear, that the types are exactly the feeders we can develop. With other words, we need to define our feeder as possible POWER List type.
 - Transaction POWL_TYPER
Finally, we connect the APPLID with the type and make it visible to the role. In detail this means, that I can select my APPLID in a role item, now.
- At this point, your POWER List should work. Nevertheless, you will discover your POWER List as empty, since there is no default query available.
- You could now – as a user – create your own query based on the object type (feeder) we have offered. But honestly, this isn't really the way it was meant to be.

Defining a Default Query



- So, what we are going to do now, is to define a default query for the POWER List.
- Default queries are defined using the transaction "POWL_QUERY". Here you can define a query (QUERYID) and connect it to a POWER Lists type. Finally, you can set the query.
 - Transaction "POWL_QUERY"

Enter the transaction and maintain a name for your query (QUERYID). Add the type which you have defined before. Press save. You should see two new buttons coming up "Query Parameters" and "Query Settings". If not, leave the screen and enter it again by double clicking on the QueryID you have defined.

Default Query Parameters & Settings

The screenshot displays the 'Maintain Table Views: Initial Screen' in SAP. The main window shows the 'Query Parameters' tab for a query named 'Z_TEST1_QUERY1'. Below this, a 'View: Query definition' section shows the description 'Query 1 for Z_Test1'. To the left, a list of parameters for the query 'POWL_PO_OPEN' is shown, including fields like 'Purchasing Document', 'Vendor', 'Purch. Organization', etc., with corresponding 'to' and 'from' values. To the right, a 'Selection criteria settings' window is open, showing a table with columns for 'Mandatory', 'Read only', 'Hidden', and 'Quick search' for various selection criteria.

	Mandatory	Read only	Hidden	Quick search
Purchasing...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vendor	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Purch. Org.	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Purchasing...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Document...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
TIMFR	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Purchasing...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Created by	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Supplying...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Company...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Incomplete	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Item	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Material	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Plant	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Material Gr...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vendor Mat...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Item Categ...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Acct Assign...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
EAN/UPC	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Vendor Subr...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Promotion	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Season	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Season Year	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Storage Loc...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Deletion In...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delivery Co...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Confirmat...	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Short Text	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Delivery Date	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Name	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
Purch. Doc. Categor...	<input type="checkbox"/>	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>

■ Now you can maintain the query parameters and the query settings

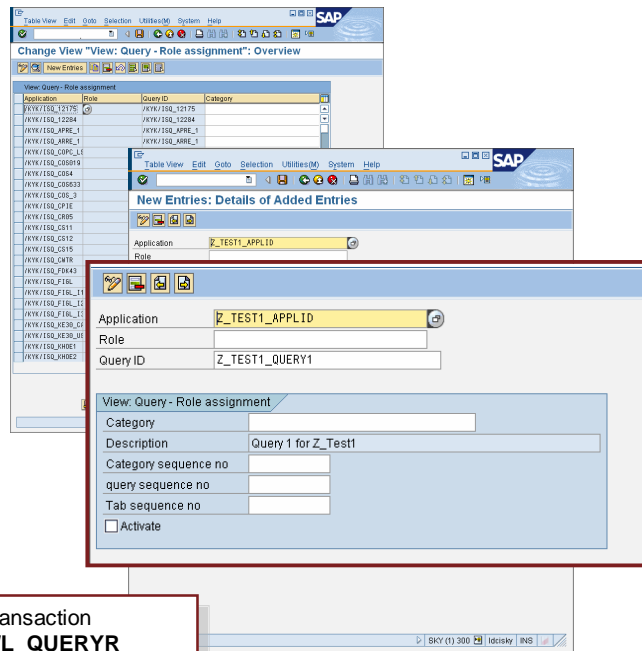
– Button “Query Parameters”

Here you can set the parameters the same way as you would do it as a user creating a new query. However, the difference here is, that this setting is available system wide and is therefore available to every user. Define your settings, press the “check” button followed by the “accept” button.

– Button “Query Settings”

In the upcoming window, you can specify several attributes mapped to each single selection criteria. TEXT NEEDS TO BE MAINTAINED

Make new Query Visible



Transaction
POWL_QUERYR

- As with the type, the query needs to be introduced to the roles. Therefore the APPLID and the QUERYID get mapped to each other.
 - Transaction POWL_QUERYR
 - Start the transaction and enter the APPLID of your feeder and the ID of the query you have defined earlier.
- Your POWER List should now come up with a predefined query.

POWER Lists (Part I): Exercise



As practice, please perform the following exercises:

- Design a new POWER List
 - The new POWER Lists should collect at least all sales orders in the backend system
 - At least one button should be available to display a specific sales order using transaction VA01
- Implement the new POWER List
 - Develop your POWER List with transaction se24
 - Please maintain at least:
 - Get_Object_Definition
 - Get_Objects
 - Get_Selection_Criteria
 - Get_Actions
 - Handle_Actions
 - Optional, you can maintain: Get_Field_Catalog
- Register your POWER Lists and make it visible

- Before we end this chapter. You should perform the following exercise.
- If you have questions, please get in contact with your instructor

POWER Lists (Part I): Unit Summary



You are now able to:

- Explain the concept of POWER Lists
- Identify cases where POWER Lists can bring value and make a design for these cases
- Start implementing your own POWER Lists

- You should now be able to explain what a POWER List is and name the advantages of it.
- You are able to identify the values of the POWER List which can help you in your business.
- You know, which transactions are required to start developing a POWER List.
- You can explain what a feeder is and what this has to do with the POWER List.
- You know how to introduce a POWER List to roles and how to use it

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

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

Microsoft, Windows, Excel, Outlook, and PowerPoint are registered trademarks of Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower and PowerPC are trademarks or registered trademarks of IBM Corporation.

Adobe, the Adobe logo, Acrobat, PostScript, and Reader are either trademarks or registered trademarks of Adobe Systems Incorporated in the United States and/or other countries.

Oracle is a registered trademark of Oracle Corporation.

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

Citrix, ICA, Program Neighborhood, MetaFrame, WinFrame, VideoFrame, and MultiWin are trademarks or registered trademarks of Citrix Systems, Inc.

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

Java is a registered trademark of Sun Microsystems, Inc.

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

MaxDB is a trademark of MySQL AB, Sweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, and other SAP products and services mentioned herein as well as their respective logos are trademarks or registered trademarks of SAP AG in Germany and in several other countries all over the world. All other product and service names mentioned are the trademarks of their respective companies. Data contained in this document serves informational purposes only. National product specifications may vary.

The information in this document is proprietary to SAP. No part of this document may be reproduced, copied, or transmitted in any form or for any purpose without the express prior written permission of SAP AG.

This document is a preliminary version and not subject to your license agreement or any other agreement with SAP. This document contains only intended strategies, developments, and functionalities of the SAP® product and is not intended to be binding upon SAP to any particular course of business, product strategy, and/or development. Please note that this document is subject to change and may be changed by SAP at any time without notice.

SAP assumes no responsibility for errors or omissions in this document. SAP does not warrant the accuracy or completeness of the information, text, graphics, links, or other items contained within this material. This document is provided without a warranty of any kind, either express or implied, including but not limited to the implied warranties of merchantability, fitness for a particular purpose, or non-infringement.

SAP shall have no liability for damages of any kind including without limitation direct, special, indirect, or consequential damages that may result from the use of these materials. This limitation shall not apply in cases of intent or gross negligence.

The statutory liability for personal injury and defective products is not affected. SAP has no control over the information that you may access through the use of hot links contained in these materials and does not endorse your use of third-party Web pages nor provide any warranty whatsoever relating to third-party Web pages.

Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, ohne die ausdrückliche schriftliche Genehmigung durch SAP AG nicht gestattet. In dieser Publikation enthaltene Informationen können ohne vorherige Ankündigung geändert werden.

Die von SAP AG oder deren Vertriebsfirmen angebotenen Softwareprodukte können Softwarekomponenten auch anderer Softwarehersteller enthalten.

Microsoft®, WINDOWS®, NT®, EXCEL®, Word®, PowerPoint® und SQL Server® sind eingetragene Marken der Microsoft Corporation.

IBM, DB2, DB2 Universal Database, OS/2, Parallel Sysplex, MVS/ESA, AIX, S/390, AS/400, OS/390, OS/400, iSeries, pSeries, xSeries, zSeries, System i, System i5, System p, System p5, System x, System z, System z9, z/OS, AFP, Intelligent Miner, WebSphere, Netfinity, Tivoli, Informix, i5/OS, POWER, POWER5, POWER5+, OpenPower und PowerPC sind Marken oder eingetragene Marken der IBM Corporation.

Adobe, das Adobe Logo, Acrobat, PostScript und Reader sind Marken oder eingetragene Marken von Adobe Systems Inc. in den USA und/oder anderen Ländern.

ORACLE® ist eine eingetragene Marke der ORACLE Corporation.

UNIX®, X/Open®, OSF/1® und Motif® sind eingetragene Marken der Open Group.

Citrix®, das Citrix-Logo, ICA®, Program Neighborhood®, MetaFrame®, WinFrame®, VideoFrame®, MultiWin® und andere hier erwähnte Namen von Citrix-Produkten sind Marken von Citrix Systems, Inc.

HTML, DHTML, XML, XHTML sind Marken oder eingetragene Marken des W3C®, World Wide Web Consortium, Massachusetts Institute of Technology.

JAVA® ist eine eingetragene Marke der Sun Microsystems, Inc.

JAVASCRIPT® ist eine eingetragene Marke der Sun Microsystems, Inc., verwendet unter der Lizenz der von Netscape entwickelten und implementierten Technologie.

MaxDB ist eine Marke von MySQL AB, Schweden.

SAP, R/3, mySAP, mySAP.com, xApps, xApp, SAP NetWeaver, und weitere im Text erwähnte SAP-Produkte und -Dienstleistungen sowie die entsprechenden Logos sind Marken oder eingetragene Marken der SAP AG in Deutschland und anderen Ländern weltweit. Alle anderen Namen von Produkten und Dienstleistungen sind Marken der jeweiligen Firmen. Die Angaben im Text sind unverbindlich und dienen lediglich zu Informationszwecken. Produkte können länderspezifische Unterschiede aufweisen.

Die in dieser Publikation enthaltene Information ist Eigentum der SAP. Weitergabe und Vervielfältigung dieser Publikation oder von Teilen daraus sind, zu welchem Zweck und in welcher Form auch immer, nur mit ausdrücklicher schriftlicher Genehmigung durch SAP AG gestattet.

Bei dieser Publikation handelt es sich um eine vorläufige Version, die nicht Ihrem gültigen Lizenzvertrag oder anderen Vereinbarungen mit SAP unterliegt. Diese Publikation enthält nur vorgesehene Strategien, Entwicklungen und Funktionen des SAP®-Produkts. SAP entsteht aus dieser Publikation keine Verpflichtung zu einer bestimmten Geschäfts- oder Produktstrategie und/oder bestimmten Entwicklungen. Diese Publikation kann von SAP jederzeit ohne vorherige Ankündigung geändert werden.

SAP übernimmt keine Haftung für Fehler oder Auslassungen in dieser Publikation. Des Weiteren übernimmt SAP keine Garantie für die Exaktheit oder Vollständigkeit der Informationen, Texte, Grafiken, Links und sonstigen in dieser Publikation enthaltenen Elementen. Diese Publikation wird ohne jegliche Gewähr, weder ausdrücklich noch stillschweigend, bereitgestellt. Dies gilt u. a., aber nicht ausschließlich, hinsichtlich der Gewährleistung der Marktgängigkeit und der Eignung für einen bestimmten Zweck sowie für die Gewährleistung der Nichtverletzung geltenden Rechts.

SAP haftet nicht für entstandene Schäden. Dies gilt u. a. und uneingeschränkt für konkrete, besondere und mittelbare Schäden oder Folgeschäden, die aus der Nutzung dieser Materialien entstehen können. Diese Einschränkung gilt nicht bei Vorsatz oder grober Fahrlässigkeit.

Die gesetzliche Haftung bei Personenschäden oder Produkthaftung bleibt unberührt. Die Informationen, auf die Sie möglicherweise über die in diesem Material enthaltenen Hotlinks zugreifen, unterliegen nicht dem Einfluss von SAP, und SAP unterstützt nicht die Nutzung von Internetseiten Dritter durch Sie und gibt keinerlei Gewährleistungen oder Zusagen über Internetseiten Dritter ab.