International Postal System

Free Reporting Guide

version 2023

Last updated: 23 November 2023

Contact

Postal Technology Centre - Universal Postal Union Weltpoststrasse 4, 3015 Bern - Switzerland

Phone: +41 31 350 31 11

Service Desk Portal: https://support.upu.int

PTC
PostalTechnologyCentre
UNIVERSAL POSTAL UNION

Table of contents

About this document 4
Intended audience 4
How to use this manual 4

Introduction 5
The Functions screen 5

Creating IPS free reports 6

Defining stored procedures 12
Executing the report 13

Appendix A: Free report examples 15
Version 1 - All users returned 15
Version 2 - Valid or invalid users returned 18
Version 3 - Valid or invalid users returned, with user group selection 21
Version 3.1 - Current user not displayed shortcut 27
Version 3.2 - Current user not displayed using a hidden parameter 28

Appendix B: Free reporting tips 31

About this document

Intended audience

This manual is intended to provide IPS System administrators and technical users who have a good understanding of SQL and stored procedures with the necessary information to generate simple on-demand reports for ordinary IPS users to run.

How to use this manual

Other than guidelines and procedures, this manual also contains free report examples and useful tips. Refer to the relevant topic:

- for the steps on creating free reports, see "Creating IPS free reports " on page 6
- for writing stored procedures, see"Defining stored procedures " on page 12
- for free report examples, see "Appendix A: Free report examples " on page 15
- for tips on creating free reports, "Appendix B: Free reporting tips " on page 31

Introduction

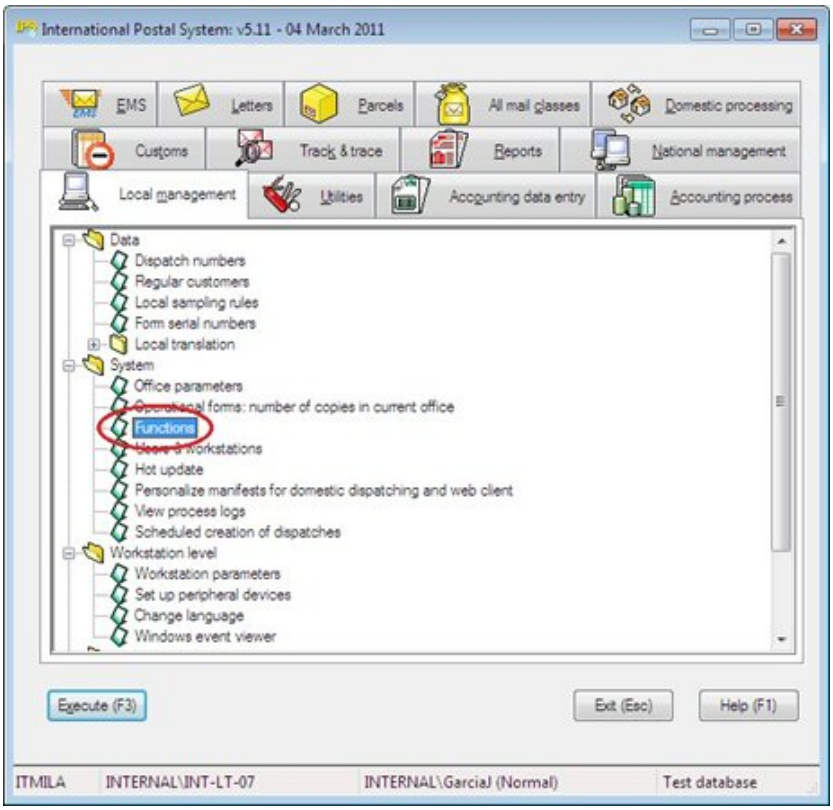
IPS provides the possibility to generate your own simple on-demand reports. To use this feature, you must have a good understanding of SQL and be able to write stored procedures. You simply create a stored procedure and configure the window. Users can run your report using the familiar IPS interface without having to use stored procedures directly or understand SQL.

Reports are handled like other functions in IPS, although they are not based on a separate .DLL or executable file for each new report. The advantage of handling reports like normal functions is that reports are displayed like all other functions in the IPS main menu. You can also set up user security as you want.

When you configure the window, you add parameters as necessary for users to be able to generate the report. IPS provides several standard criteria you can use to construct the report, such as check-boxes, date and time fields, or edit boxes.

The Functions screen

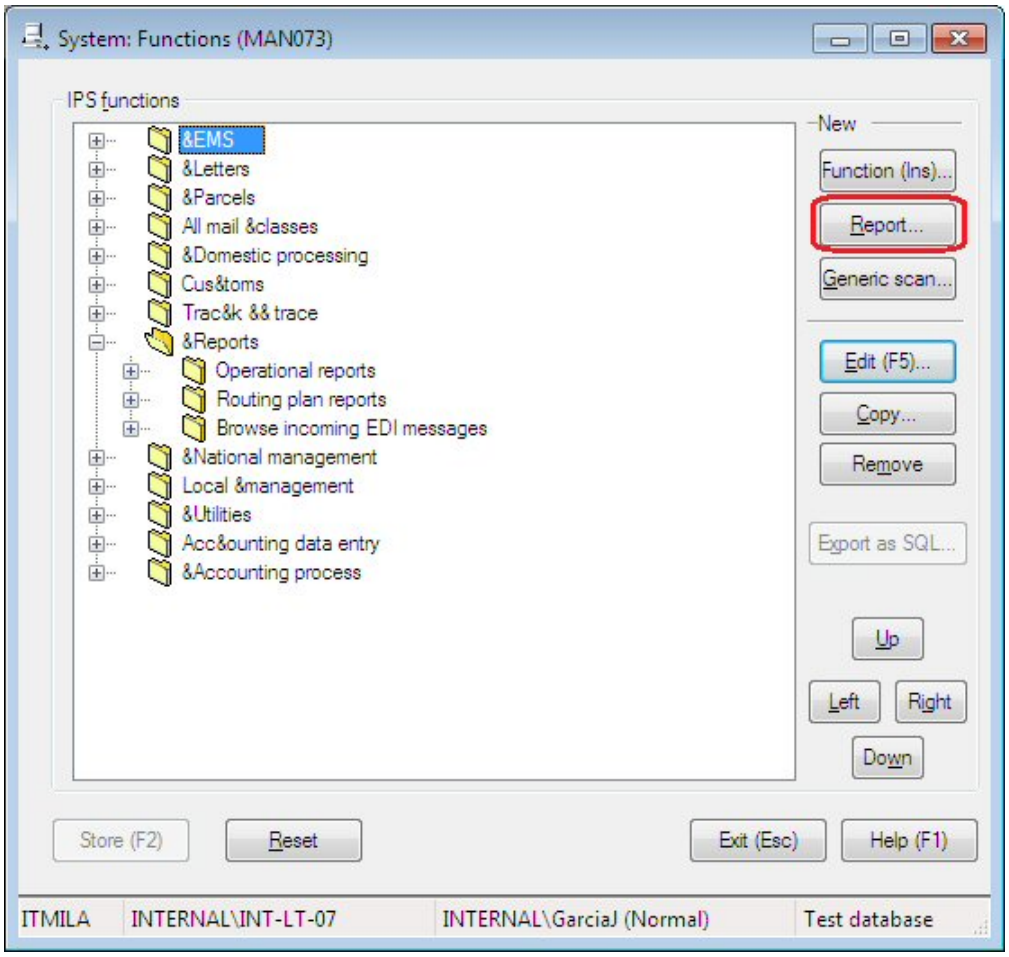
For all the procedures described in this guide, you must first open the Functions window. This is where you create new free report screens. Select the **Local management** tab, then **System > Functions** in the main IPS window:



Creating IPS free reports

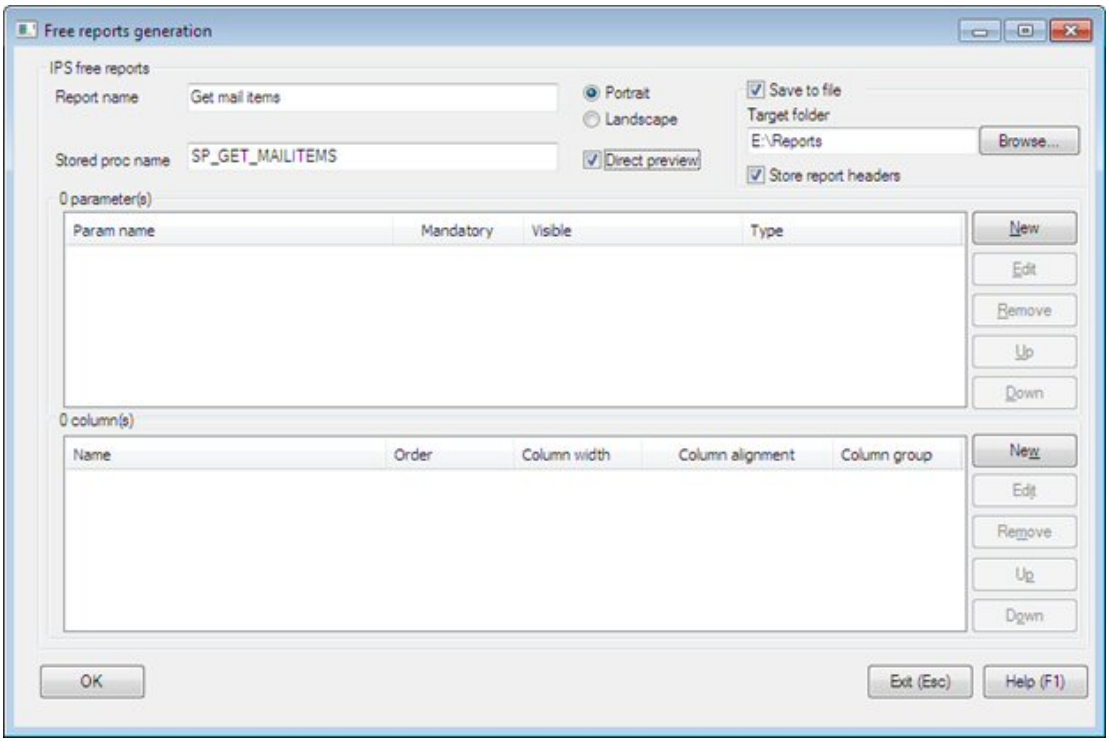
To create a free report in IPS, follow the steps below:

1. Open the **Functions** window.
2. Select the folder where you want the new report screen to be located.
3. In the "New" area, click the **Report** button, shown in the following screen.

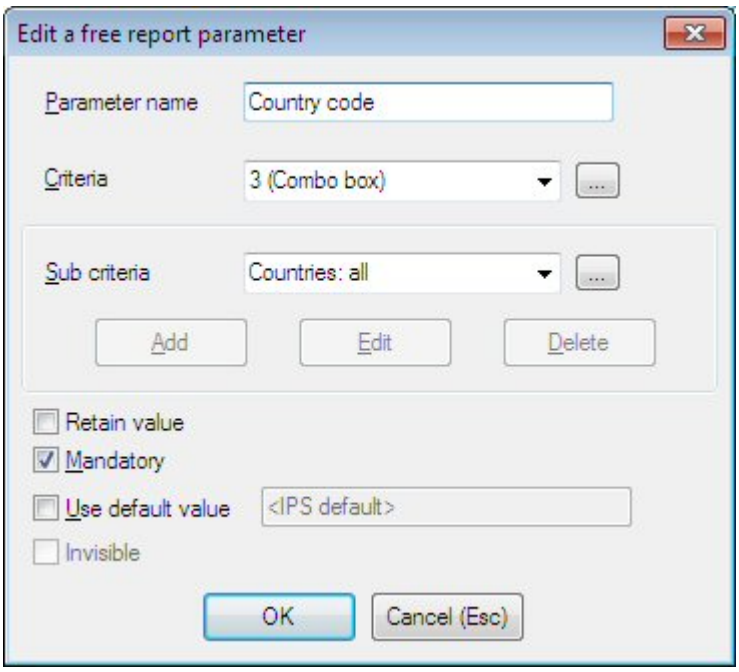


4. Type the new free report name in the Report name field. This is the name that appears in the IPS main menu.
5. In the Stored proc name field, type the name of the stored procedure that extracts the data from the database to create the report. The procedure name must be entered without the .SQL extension. The procedure must return a simple list of data, and all its input parameters must be strings. Use conversions in the stored procedure for numeric values and dates and times.
6. Select the **Portrait** or **Landscape** radio button depending on the report's intended layout; vertical or horizontal.
7. Select the **Direct preview** checkbox if you want IPS to automatically display a preview of the report on the screen for the user.

8. Select the **Save to file** checkbox if you want to automatically save the report to a file. Click the **Browse...** button in the Target folder field and locate the directory where you want to save the report. It will be saved as a comma separated file.
9. Select the **Store report headers** checkbox if you want to include column headers in the report.



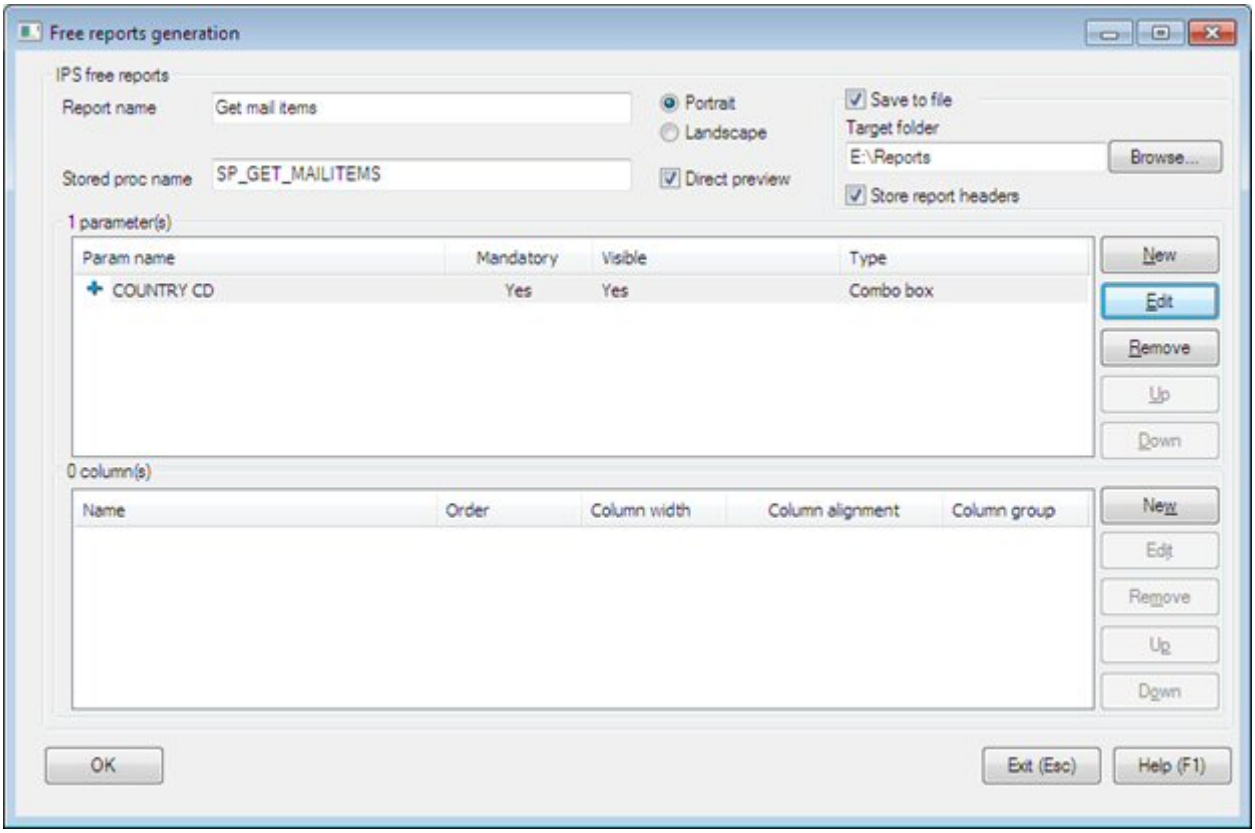
10. In the parameters area, click the **New** button. A new window opens.



11. Define the parameters for the report.

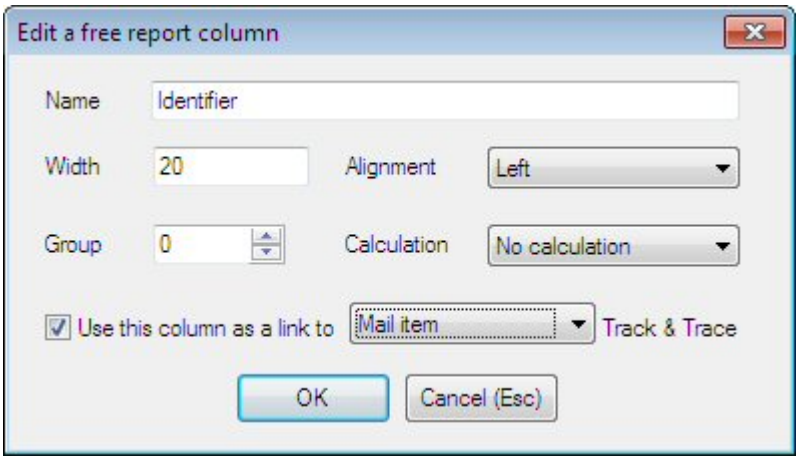
Parameter name	The label that will be displayed to help users enter the criteria.
Criteria	The type of data element to display on the user interface screen, such as a checkbox or a combo box. The data element you specify will appear on the window used by the user to generate the report. Some types of criteria require you to specify sub-criteria (see below).
Sub-criteria	This option is enabled only for certain criteria types. Select a value to define the type of criteria. For example, if you specified combo box in the Criteria field, you must select the list of possible options for the combo box as the sub-criteria; i.e. you can specify a sub-criterion that lets users select from a list of airports, countries or mail classes, etc.

12. Click the **OK** button to close this window. The parameter you defined appears in the parameters list box with a plus sign (+) next to it.

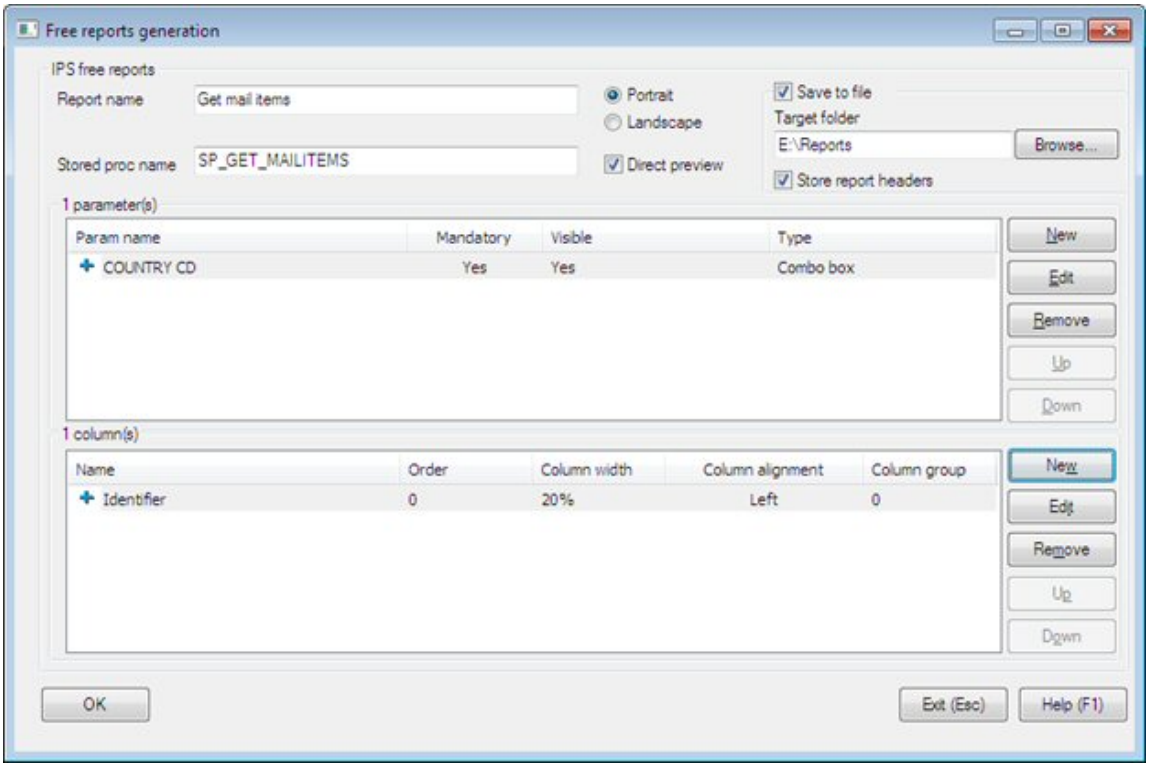


13. Repeat steps 10 to 12 for each parameter you want to define in your report. Add as many parameters as necessary. The order of parameters is very important. Your stored procedure receives parameters in the same order as you define here. If you want to change the order of any parameters, use the **Up** or **Down** buttons to reposition them.

14. The Columns list box shows the columns that will appear on the final report as the user sees it and the settings for each column. When creating a report, you need to define the columns returned by the stored procedure. To define a column click the **New** button. Alternatively, if you want to change the appearance of the report, select an existing column and click the **Edit** button. Enter the column Name in the text area. You can also specify the column's width and alignment. The default column width is 10% of the total report width and the default alignment is left. If you do not want a column to be displayed, set the column width to 0%. You can also assign a link from the column to a track and trace attribute, by selecting the checkbox on the screen. In the example below, it is linked to the Mail item attribute. It is also possible to define column groups and value calculations. This is described later on in this document.

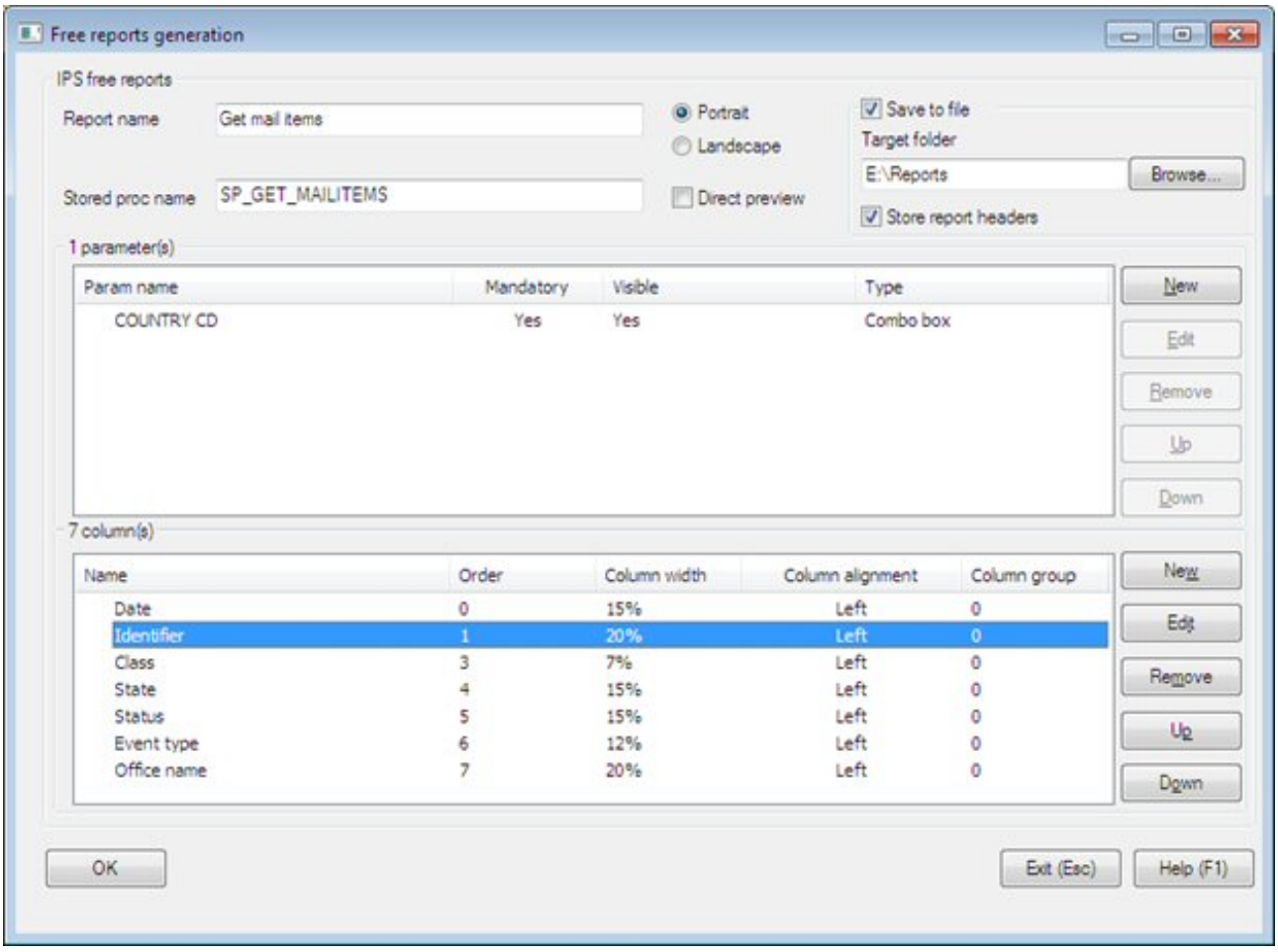


15. Click the **OK** button to close this window. The column you defined appears in the Column list box with a plus sign (+) next to it.

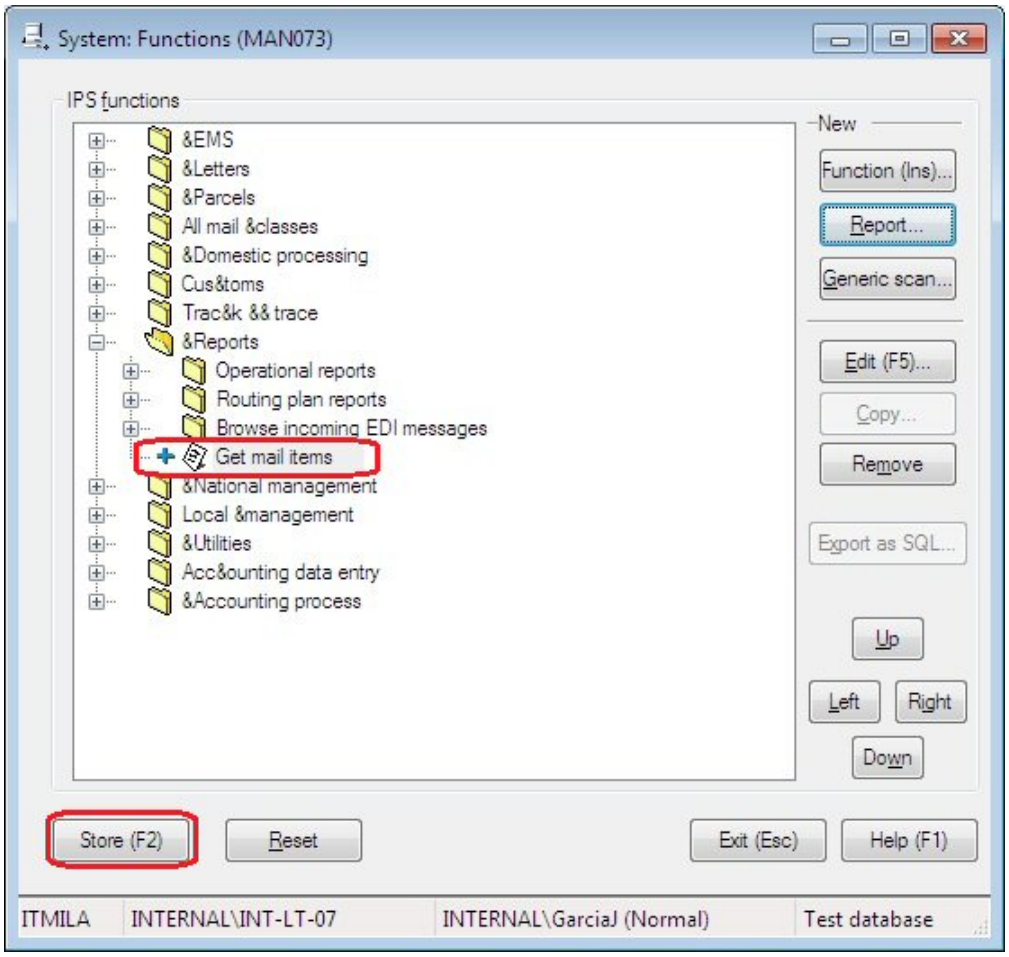


16. The order of columns is very important as it reflects the order of the dataset returned from the stored procedure. As before, if you want to change the order of any of the columns, use the **Up** or **Down** buttons to reposition them.

17. In the example below, the following columns have been added to the result set; Date, Identifier, Class, State, Status, Event type, and Office name. When you have finished adding columns click the **OK** button.



18. On the System: Functions screen, in the IPS functions list you see a plus (+) sign next to the new report. Click the **Store** button and restart the IPS application to be able to start using your free report.



Defining stored procedures

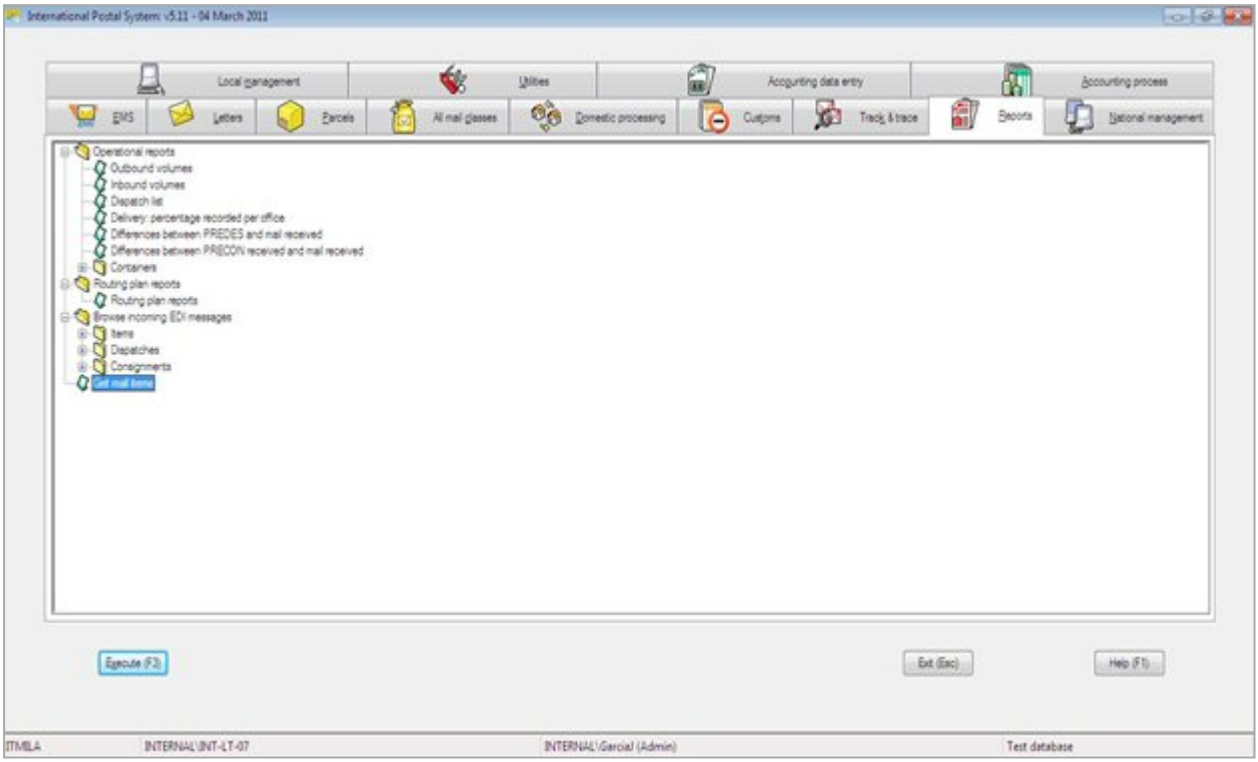
The following stored procedure extracts from the database the data required to produce the Get mail items report that was created using the procedure given in section "Creating IPS free reports " on page 6. Note that the seven values returned in the resulting data set of the stored procedure match the seven return values defined in the free reports generation configuration screen. The actual names of the attributes defined are not exact matches, but the number (seven) and sequence (date, identifier, etc.) of the attributes must be respected by the report.

```
GO

SET ANSI_NULLS ON

GO
SET QUOTED_IDENTIFIER ON

GO

--exec SP_GET_MAILITEMS @COUNTRY_CD=N'CD'

ALTER PROCEDURE [dbo].[SP_GET_MAILITEMS]

          @COUNTRY_CD char(2)

AS
BEGIN

     SELECT EVT_GMT_DT AS DATE, MAILITM_FID AS IDENTIFIER,

               MAIL_CLASS_CD AS MAIL_CLASS,
               CSI.STATE_IND_NM AS STATE,
               CPS.POSTAL_STATUS_NM AS STATUS,
               CET.EVENT_TYPE_NM AS EVENT_TYPE,
               NOO.OFFICE_NM AS OFFICE_NAME

     FROM L_MAILITMS LMS
     INNER JOIN C_STATE_INDS CSI ON LMS.STATE_IND_CD = CSI.STATE_IND_CD
     INNER JOIN C_POSTAL_STATUSES CPS ON

                       CPS.POSTAL_STATUS_CD = LMS.POSTAL_STATUS_CD

     INNER JOIN C_EVENT_TYPES CET ON CET.EVENT_TYPE_CD = LMS.EVT_TYPE_CD
     INNER JOIN N_OWN_OFFICES NOO ON NOO.OWN_OFFICE_CD = LMS.EVT_OFFICE_CD
     WHERE LMS.ORIG_COUNTRY_CD = @COUNTRY_CD

END
GO

GRANT EXECUTEON dbo.SP_GET_MAILITEMS TO IPSUsers

GO
```

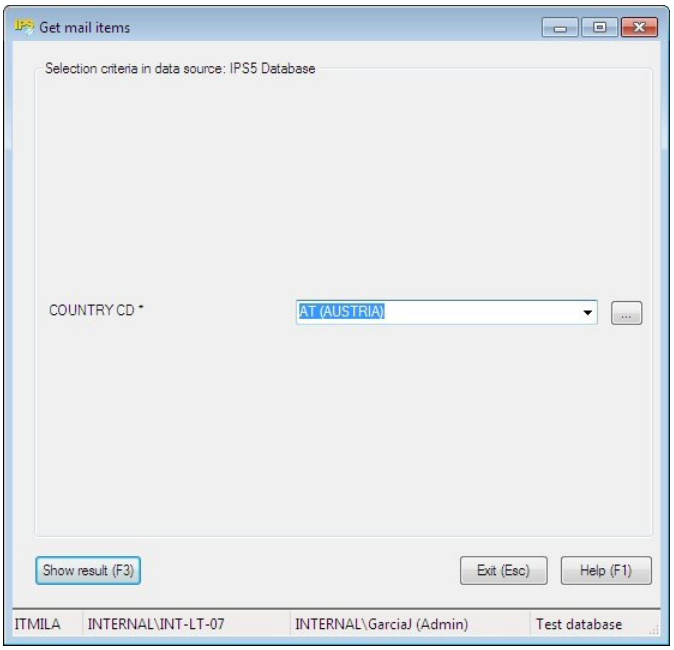
To produce the data set required, the operational table L_MAILITEMS has been joined with three reference data tables; C_, and with a configuration data table; N_.

Executing the report

In order to execute the report, IPS users will need to access its associated function within the application area (tab and folder/subfolder) where it was defined. In the example report described above, the "Get mail items" report was added to the Reports tab within the root folder.

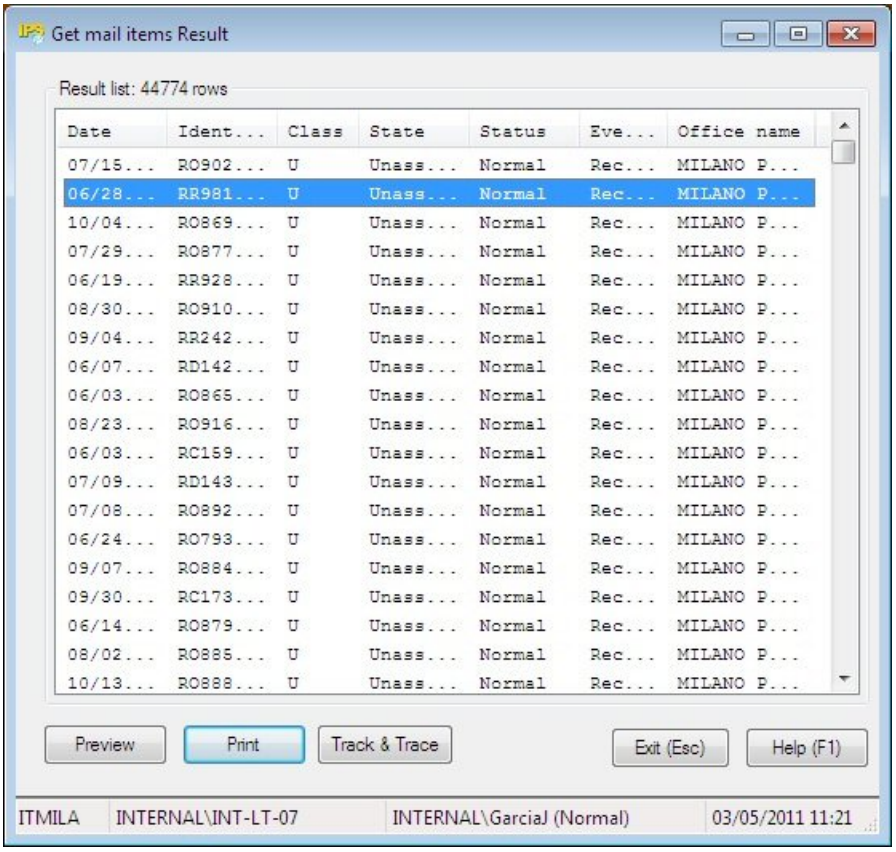


On the report's screen, select the appropriate values for the report's parameters and click the **Show result (F3)** button.



Depending on whether in the report configuration screen you selected to “Save the report to a file”, the report’s result set will be saved to a file in the directory configured for this purpose, or displayed on a new screen.

On the new screen, you can preview and print the report, and show the track and trace data for any of the items.



Appendix A: Free report examples

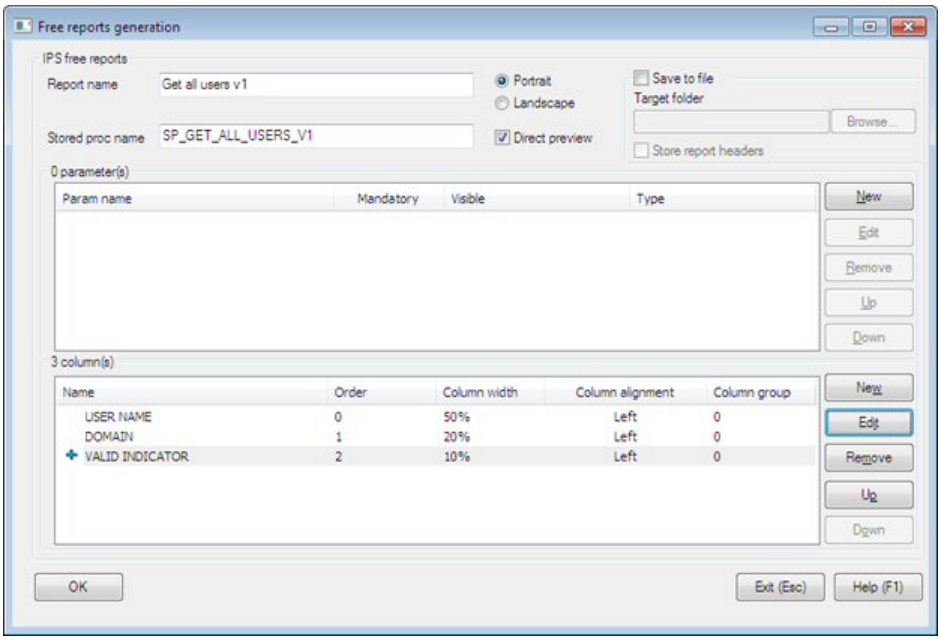
The following examples of free reports generate lists of users. To illustrate the different free reporting concepts, we will demonstrate five different versions of the report:

- Version 1: no parameters, report returns all users, with login, name and domain
- Version 2: adding a checkbox parameter 'including marked as deleted'. When the box is not checked, only valid users are returned; when it is checked, all users are returned, including the ones marked as invalid
- Version 3: adding a custom combo box parameter 'user groups'. When it is left blank, the report behaves like version 2; when selected, the report only shows users belonging to that group. The groups are provided by a stored procedure that will be created for this purpose
- Version 3.1: the report must never return the current user (this current user must not be hard-coded and must not be shown as a parameter)
- Version 3.2: as for the previous version, the report must not return the current user, but this time this is achieved using a hidden parameter

Version 1 - All users returned

Report configuration screen

On the free report definition screen we will add the three columns of data to be returned by the report.



Stored procedure

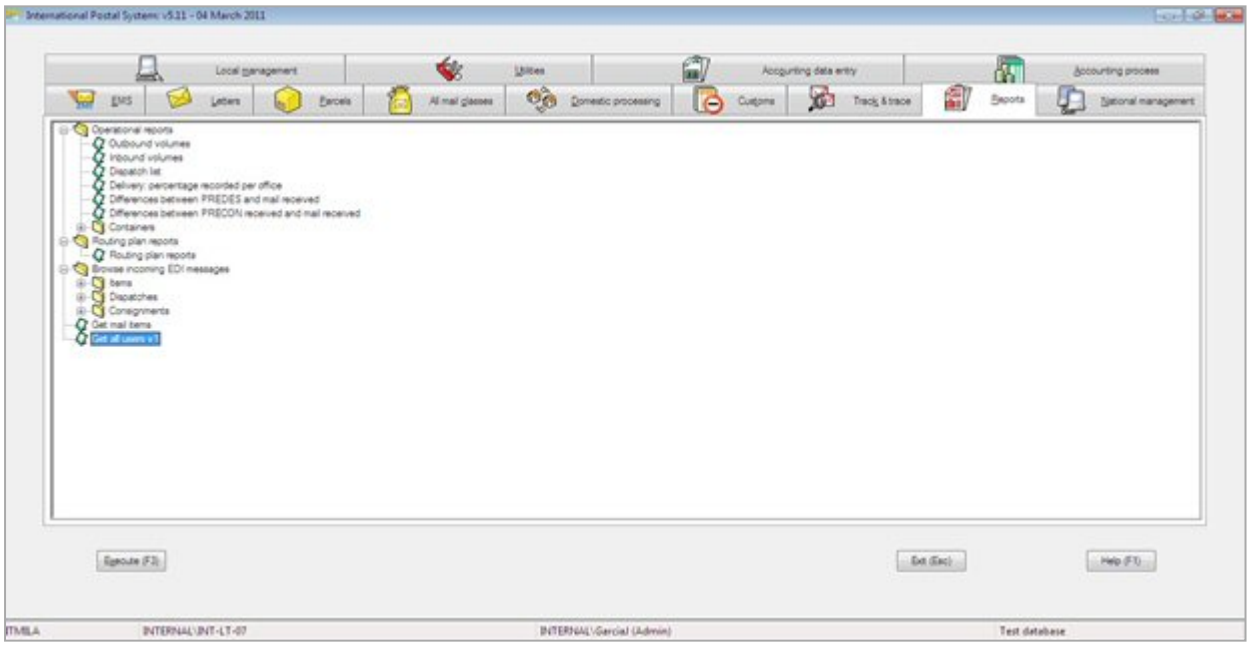
We define the report's stored procedure in the following way. Note that the third column referring to the user's valid indicator is encoded as "VALID" or "INVALID" for visibility purposes on the report.

```
GO
SET ANSI_NULLS ON
```

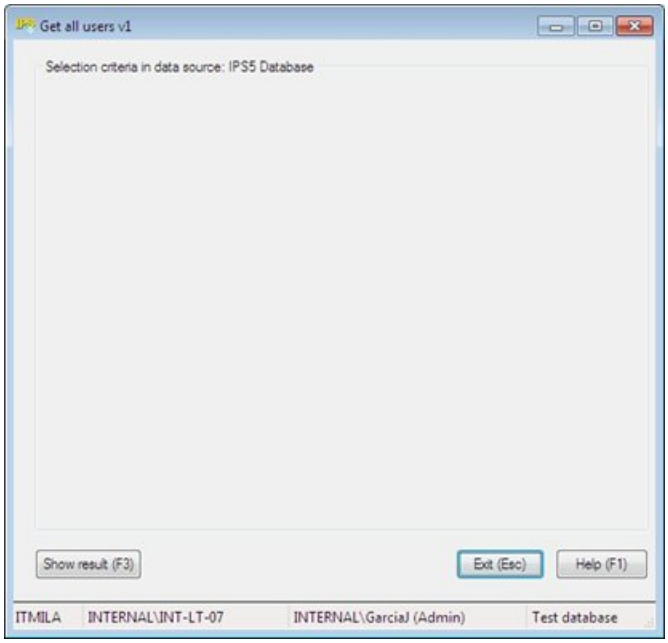
```
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[SP_GET_ALL_USERS_V1]
AS
BEGIN
      SELECT USER_NM, USER_DOMAIN,
      CASE WHEN VALID_IND = 3 THEN 'VALID' ELSE 'INVALID' END
      FROM L_USERS
END
GO


GRANT EXECUTE ON dbo.SP_GET_ALL_USERS_V1 TO IPSUsers
GO
```

Running the report

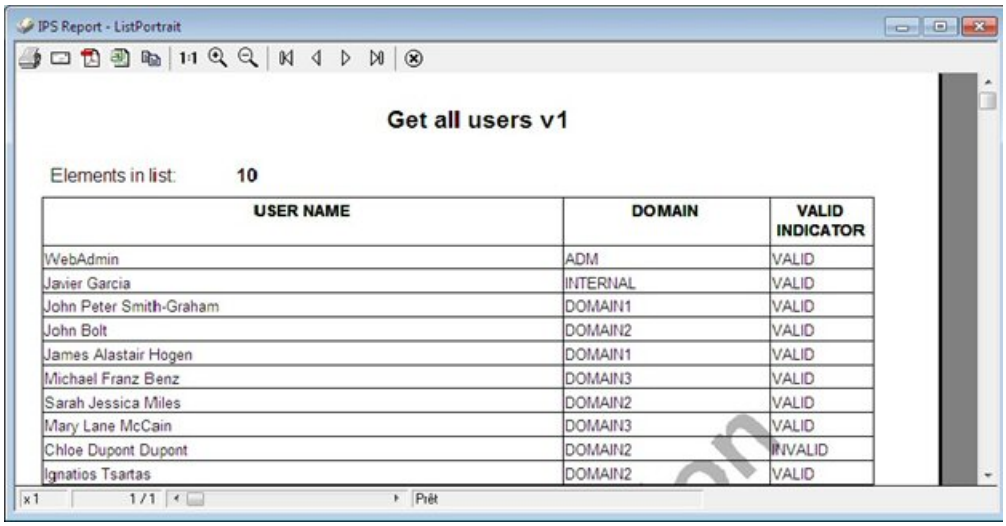
The "Get all users v1" report will be accessible from the root folder of the "Reports" tab of IPS.



No input parameters will be displayed on the report screen since none were defined.



Click the **Show result (F3)** button to display this first version of the list users report.



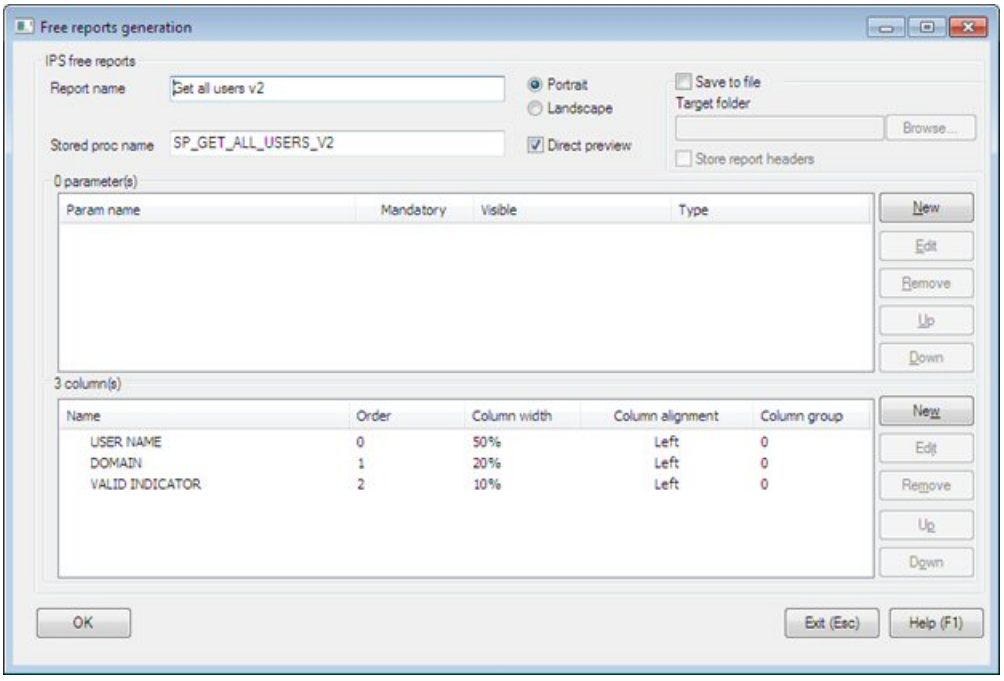
Version 2 - Valid or invalid users returned

Report configuration screen

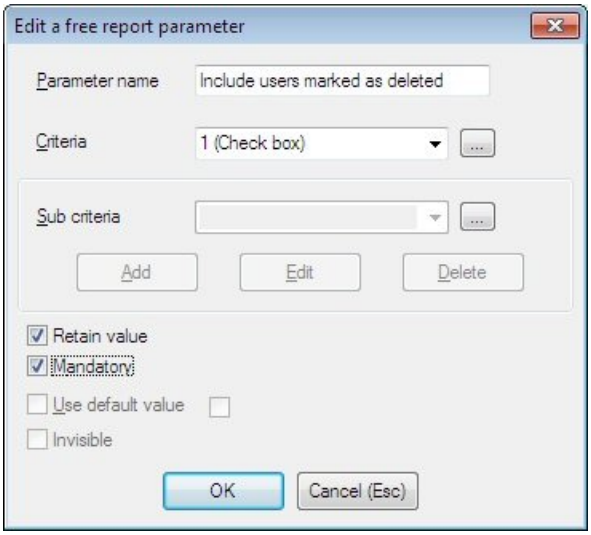
For this example, we will extend the first report and add a checkbox parameter, 'Include users marked as deleted'. When the box is not checked, only valid users are returned; when it is checked, all users are returned, including the ones marked as invalid.

The first step in extending the first report is to access the free reports generation configuration screen at "Local Management > Functions", and then double-click the "Get all users v1" report.

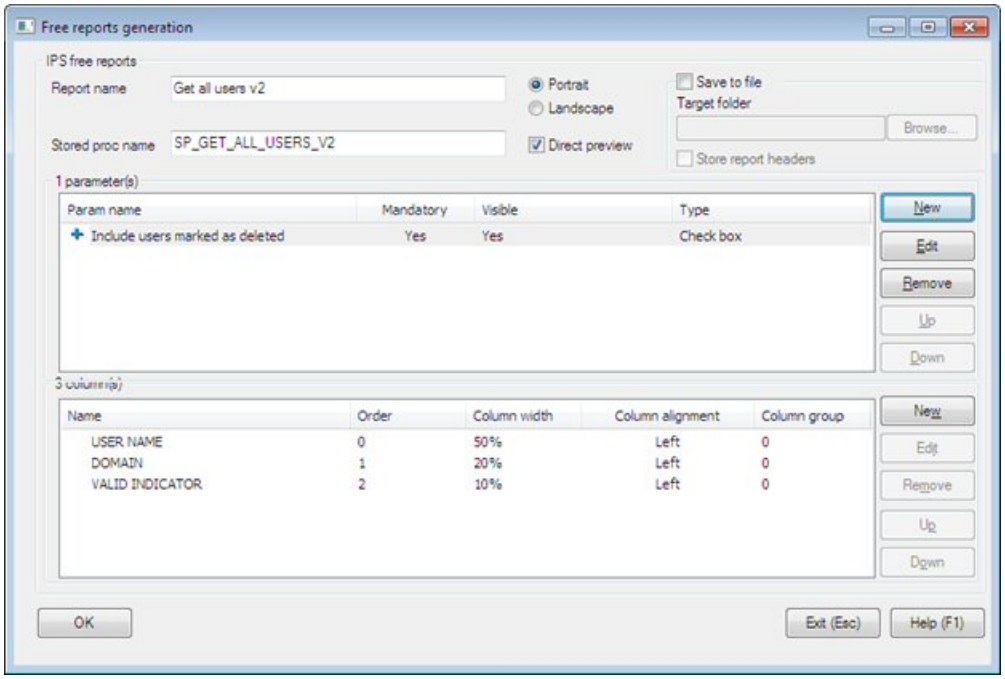
When this screen is open, the report name is then changed to "Get all users v2" and the stored procedure it calls is changed to SP_GET_ALL_USERS_V2.



On the reports parameter area, we click the **New** button to add the intended parameter.



We will make this parameter mandatory, and we will set it to retain its value, so that next time the report function is accessed, it will automatically produce the last run version of the report



Finally, we click the **OK** button on the "Free reports generation" configuration screen and the **Store** button on the "Functions" screen to save the changes.

We then need to close and restart IPS to be able to use the new function.

Stored Procedure

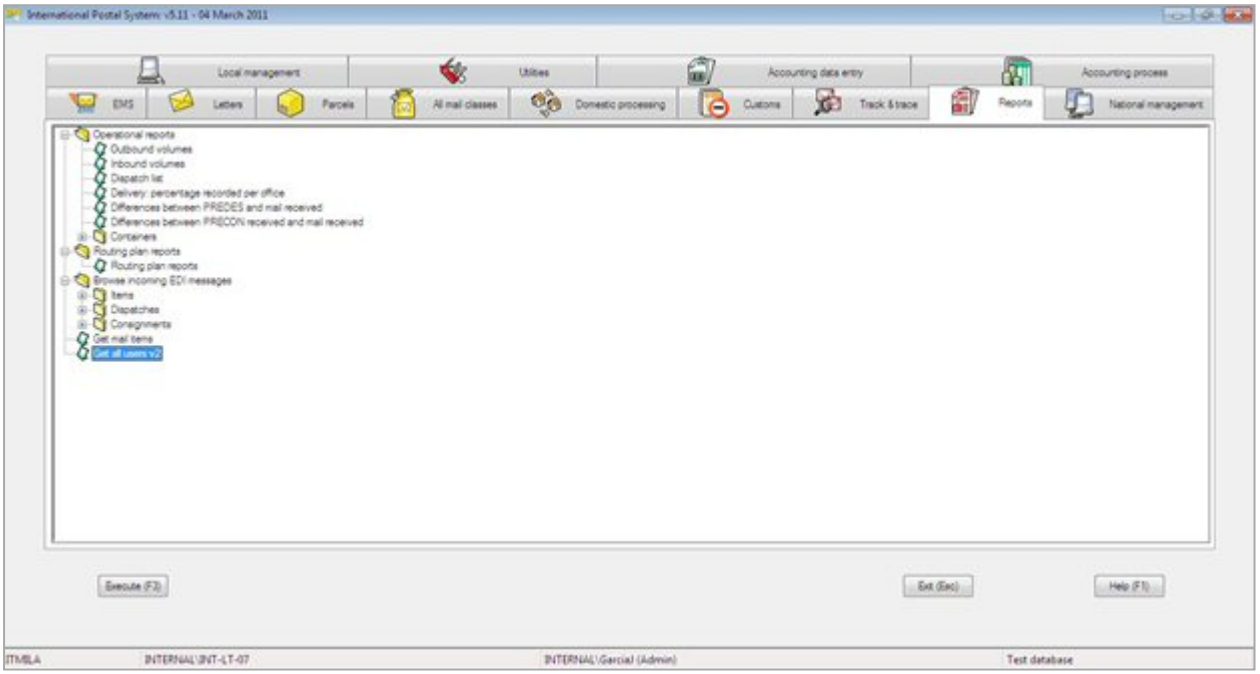
We define the report's stored procedure in the following way. Note that the value from the include deleted users checkbox is passed as a bit value (0 or 1).

```
GO
SET ANSI_NULLS ON

GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[SP_GET_ALL_USERS_V2]
@INVALID_USERS BIT
AS
BEGIN
DECLARE @VALID_IND_PAR INT
SET @VALID_IND_PAR = 3


IF @INVALID_USERS = 1
BEGIN
SET @VALID_IND_PAR = 4
END


SELECT USER_NM, USER_DOMAIN,
```

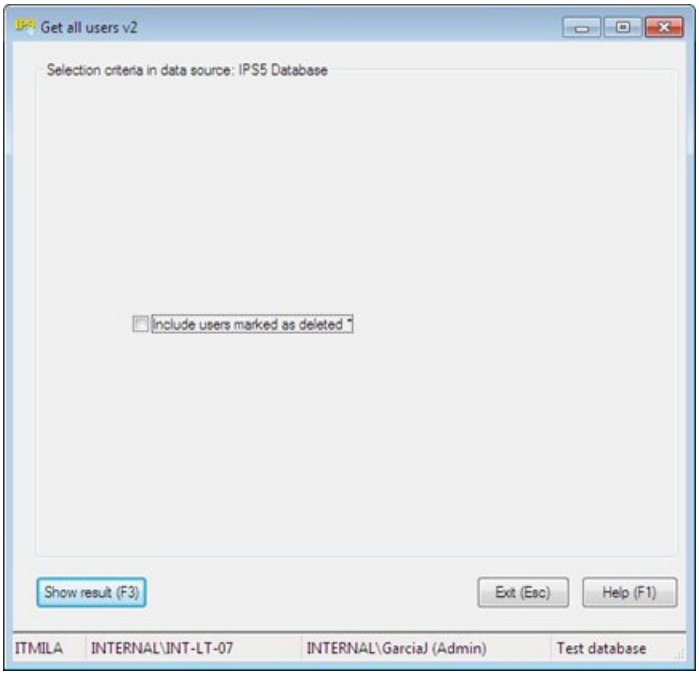
```
CASE WHEN VALID_IND = 3 THEN 'VALID' ELSE 'INVALID' END
FROM L_USERS
WHERE VALID_IND = 3 OR VALID_IND = @VALID_IND_PAR
END
GO
GRANT EXECUTE ON dbo.SP_GET_ALL_USERS_V2 TO IPSUsers
GO
```

Running the report

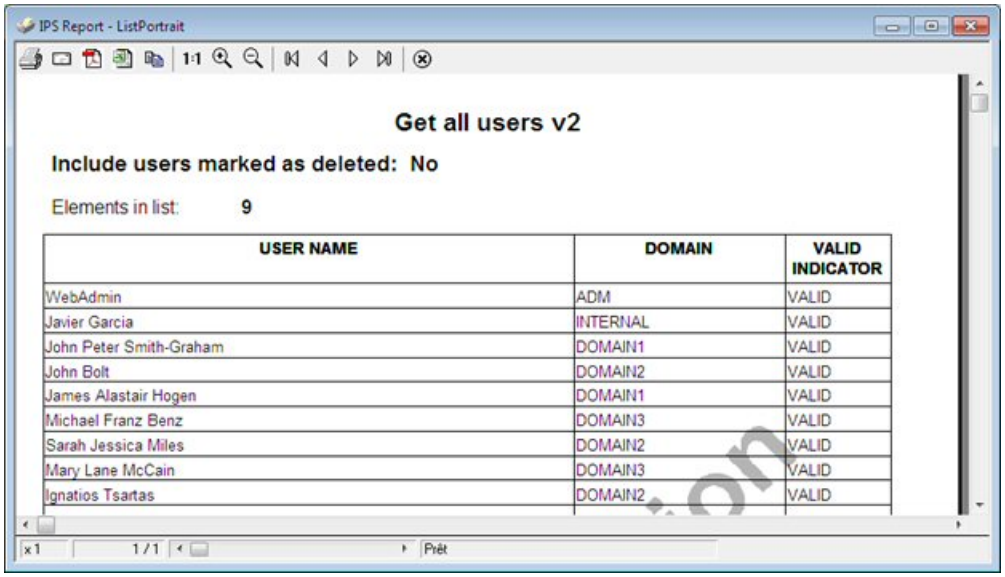
The "Get all users v2" report will be accessible from the root folder of the "Reports" tab of IPS.



On the report screen there is now a checkbox labelled "Include users marked as deleted*". The asterisk at the end shows the parameter is mandatory. If for cosmetic purposes you want to remove the asterisk from the checkbox name, you can do so by making the parameter non-mandatory on the free reports generation configuration screen. The functionality remains the same.



Click the **Show result (F3)** button, without checking the "Include users marked as deleted" checkbox, to display this version of the list users report.



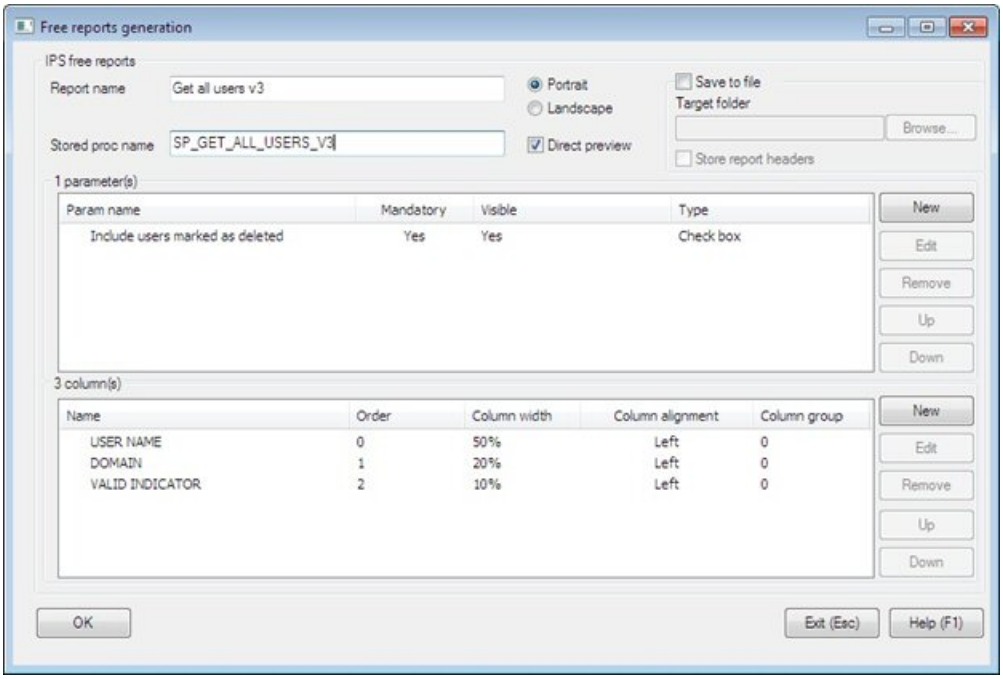
Version 3 - Valid or invalid users returned, with user group selection

For this example, we will extend the second report by adding a custom combo box parameter 'User groups'. When it is left blank, the report behaves like version 2, but if an item from the combo box is selected, the report only shows users belonging to that group. The groups are provided by a stored procedure that will be created for this purpose.

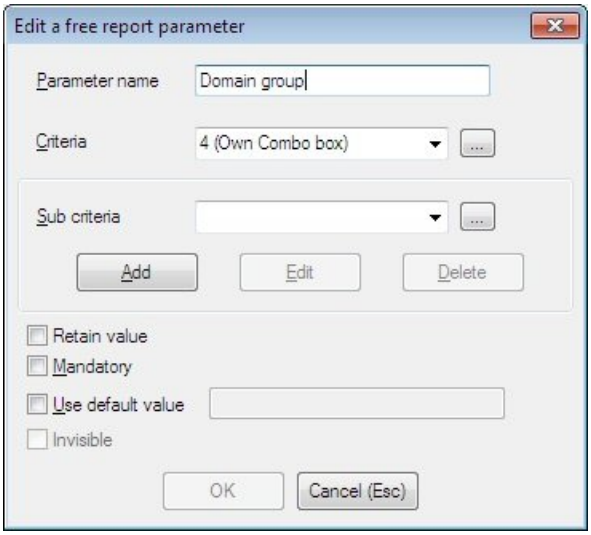
The first step in extending the second report is to access the free reports generation configuration screen under "Local Management > Functions", and then double-click the "Get all users v2" report.

Once on this screen, the report name is changed to "Get all users v3" and the stored procedure it calls changes to SP_GET_ALL_USERS_V3.

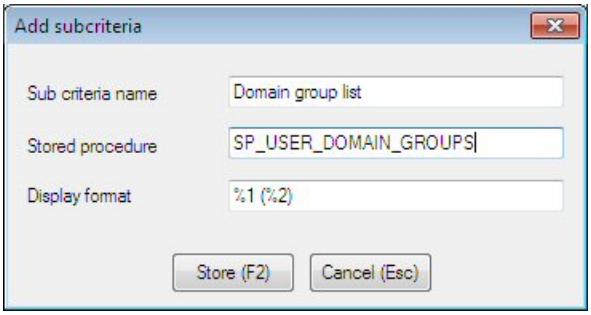
Report configuration screen



In the reports parameter area, click the **New** button to add the intended parameter.



As the parameter name, enter "Domain group", select "4 (Own Combo box)" in Criteria and to fill in the Sub criteria, click the **Add** button.

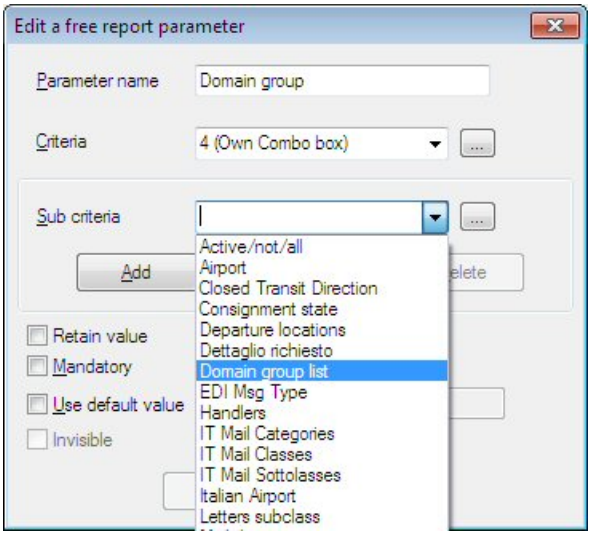


In the Add subcriteria screen, select a name (in this case "Domain group list"), a stored procedure that will provide us with the content of our own combo box, as well as a display format within the combo box. This display format is a sort of mapping between the result set of the stored procedure and the display within the combo box. In this case, the first value returned by the stored procedure %1 – DOMAIN_CODE will be displayed first, and the second value %2 – DOMAIN_NAME, will be displayed afterwards in brackets. The opposite would be achieved with the following display format string; %2 (%1).

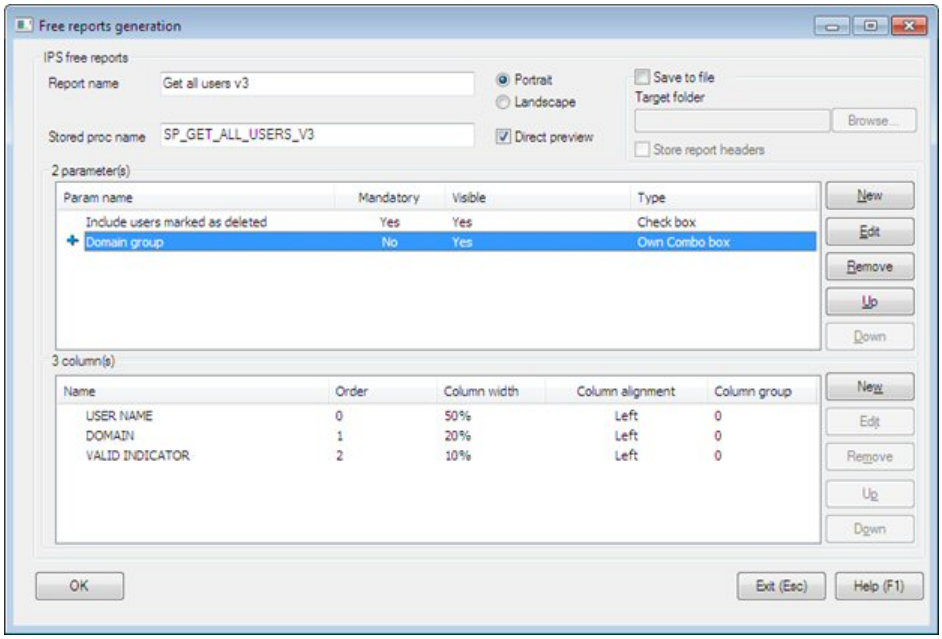
In the IPS database, you also need to define the stored procedure that populates your own combo box. In this simple example, we will take the last letter of each domain name as domain code, and we will list them together with the domain names.

```
SET ANSI_NULLS ON
GO

SET QUOTED_IDENTIFIER ON

GO

CREATE PROCEDURE dbo.SP_USER_DOMAIN_GROUPS

AS

SELECT DISTINCT SUBSTRING(USER_DOMAIN, LEN(USER_DOMAIN), 1 ) AS DOMAIN_CODE,

     USER_DOMAIN AS DOMAIN_NAME
     FROM L_USERS
     ORDER BY DOMAIN_CODE ASC

GO

GRANT EXECUTE ON dbo.SP_USER_DOMAIN_GROUPS TO IPSUsers
```

Once the new subcriteria data has been added, you need to select it in the Sub criteria drop-down list.



This time, we will retain the parameter's value, but we will not make it mandatory. After clicking the **OK** button we will be able to see that the new parameter has been added to the report. We can then click the **OK** button in the "Free reports generation" configuration screen and then click the **Store** button in the functions editing screen. We will need to close the IPS application and restart it in order to be able to use the newly defined report.



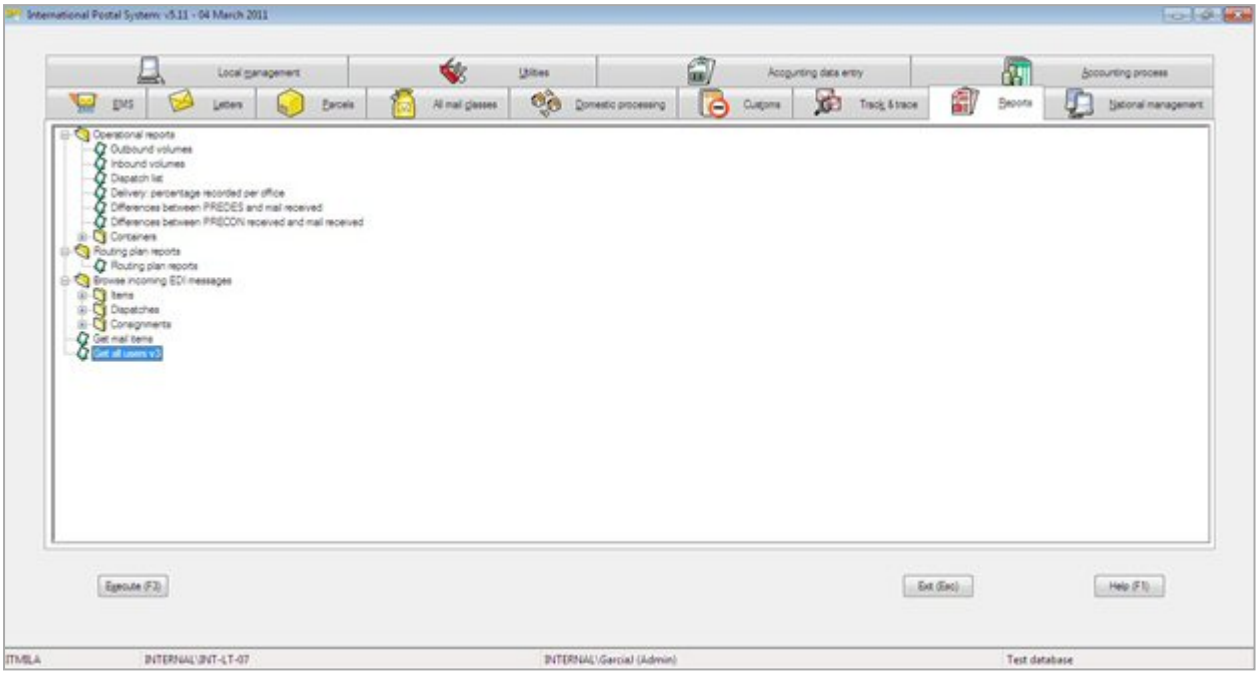
Stored Procedure

We define the report's stored procedure in the following way. Note that we are now passing a new CHAR parameter; @DOMAIN_CODE, and then select the domain to be displayed from the users table using the same expression contained in the "Domain group list" sub criteria stored procedure.

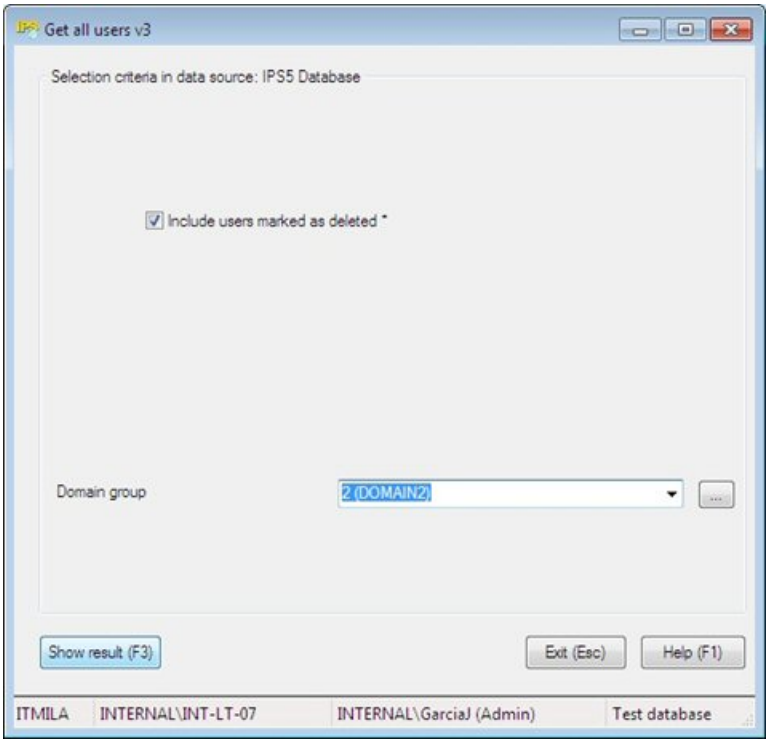
```
GO
SET ANSI_NULLS ON
GO
SET QUOTED_IDENTIFIER ON
GO
CREATE PROCEDURE [dbo].[SP_GET_ALL_USERS_V3]
@INVALID_USERS BIT,
@DOMAIN_CODE CHAR
AS
BEGIN
DECLARE @VALID_IND_PAR INT
SET @VALID_IND_PAR = 3


IF @INVALID_USERS = 1
BEGIN
SET @VALID_IND_PAR = 4
END


SELECT USER_NM, USER_DOMAIN,
CASE WHEN VALID_IND = 3 THEN 'VALID' ELSE 'INVALID' END
FROM L_USERS
WHERE (VALID_IND = 3 OR VALID_IND = @VALID_IND_PAR)
AND ( SUBSTRING(USER_DOMAIN, LEN(USER_DOMAIN), 1 ) = @DOMAIN_CODE

     OR @DOMAIN_CODE IS NULL )

END

GO

GRANT EXECUTE ON dbo.SP_GET_ALL_USERS_V3 TO IPSUsers
```

Running the report

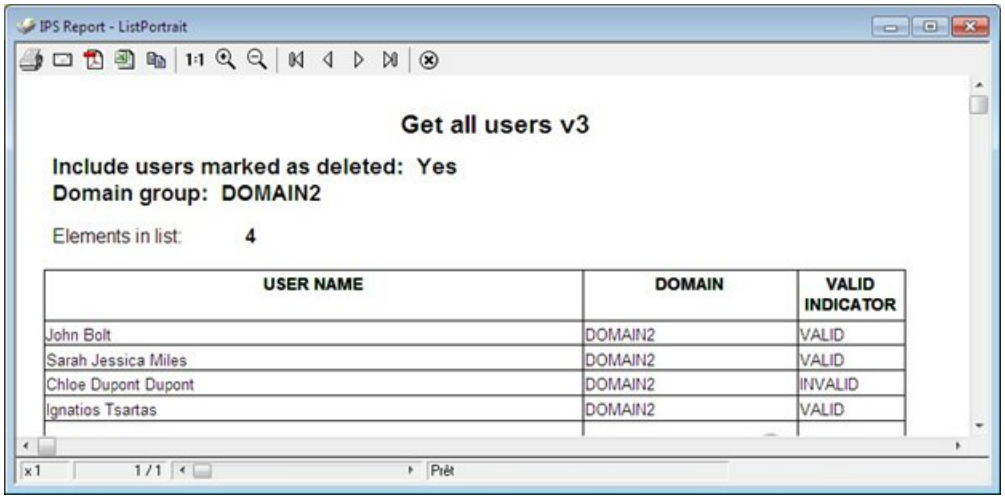
The "Get all users v3" report will be accessible from the root folder of the Reports tab of IPS.



On the report screen there is now a combo box showing the different domain groups as returned by the SP_USER_DOMAIN_GROUPS stored procedure. This parameter is not mandatory, so the combo box selection can be left empty and the report will return all users from all domains.



Click the **Show result (F3)** button to produce a report according to the predefined parameters.



Version 3.1 - Current user not displayed shortcut

For this version of the report if we are using Windows Authentication on IPS, and we are connecting directly to the DB server, we can just use the SQL Server user login which would coincide with the current IPS user. We would therefore only need to modify the stored procedure by adding the following conditional to exclude the current user from the user result set / list contained in the report.

```
(USER_FID <> right(system_user, len(user_fid)))
```

Stored Procedure

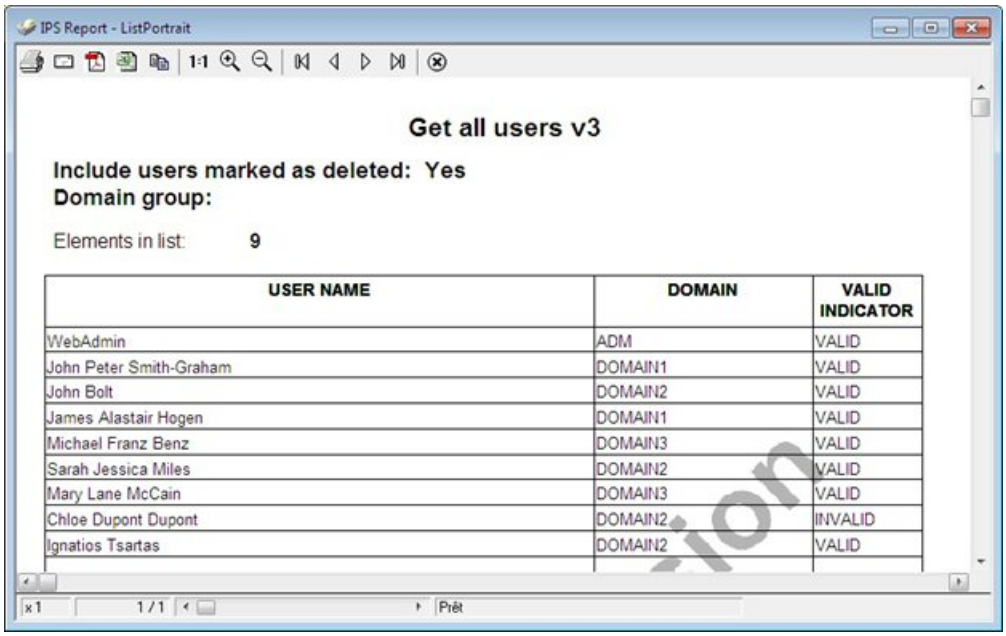
To alter the stored procedure, we will add the additional condition by running the command below.

```
GO
SET ANSI_NULLS ON

GO
SET QUOTED_IDENTIFIER ON
GO

ALTER PROCEDURE [dbo].[SP_GET_ALL_USERS_V3]
@INVALID_USERS BIT,
@DOMAIN_CODE CHAR
AS
BEGIN
DECLARE @VALID_IND_PAR INT
SET @VALID_IND_PAR = 3


IF @INVALID_USERS = 1
BEGIN
SET @VALID_IND_PAR = 4
END


SELECT USER_NM, USER_DOMAIN,
CASE WHEN VALID_IND = 3 THEN 'VALID' ELSE 'INVALID' END
FROM L_USERS
WHERE (VALID_IND = 3 OR VALID_IND = @VALID_IND_PAR)
AND ( SUBSTRING(USER_DOMAIN, LEN(USER_DOMAIN), 1 ) = @DOMAIN_CODE
```

```
      OR @DOMAIN_CODE IS NULL)
AND (USER_FID <> right(system_user, len(user_fid)))
END
GO

GRANT EXECUTE ON dbo.SP_GET_ALL_USERS_V3 TO IPSUsers
```

Running the report

After updating the stored procedure in the database we can follow the same process defined to execute version 3 of the report and we will obtain the result set detailed below. Note that the current IPS user INTERNAL\Garciaj has been excluded from the user list contained in our sample report.



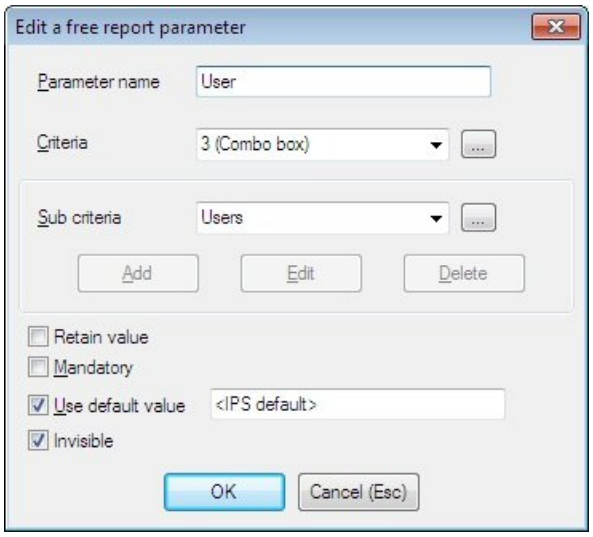
Version 3.2 - Current user not displayed using a hidden parameter

For version 3.2 of our example users report, we will use the <IPS default> property, which will return the current user, and we will make this parameter hidden, so it cannot be modified, and it is always passed to the stored procedure as expected.

Report configuration screen

The first step in extending the third report will be to access the free reports generation configuration screen in "Local Management > Functions", and then double-click the "Get all users v3" report.

On the reports parameter area, we then click the **New** button to add the intended hidden parameter. We will then select "combo box" as parameter criteria, and "Users" as sub-criteria. By clicking on the default value checkbox and entering command string "<IPS default>", the current IPS user will always be selected as the default value on this parameter. Checking the "Invisible" checkbox will hide the parameter in the report execution screen, so that the current user will always be passed on to the stored procedure.



We will make this parameter mandatory, and we will select it to retain its value, so that next time the report function is accessed it will automatically produce the last run version of the report. After clicking the **OK** button, we will be able to see that the new parameter has been added to the report. Click **OK** in the "Free reports generation" configuration screen and then click the **Store** button in the functions editing screen. We will need to close the IPS application and restart it in order to be able to use the newly defined report.

Stored Procedure

We will need to update version 3 of the list users stored procedure in the following way. Note that we are now passing a new SMALLINT parameter @USER_PID, which matches the ID of the current IPS user. We will then exclude it from the return list by adding the following condition to the stored procedure.

```
AND @USER_PID <> USER_PID
```

The alter script on the stored procedure would be the following.

```
GO
SET ANSI_NULLS ON

GO
SET QUOTED_IDENTIFIER ON
GO
ALTER PROCEDURE [dbo].[SP_GET_ALL_USERS_V3]
@INVALID_USERS BIT,
@DOMAIN_CODE CHAR,
@USER_PID SMALLINT

AS
BEGIN
DECLARE @VALID_IND_PAR INT
SET @VALID_IND_PAR = 3


IF @INVALID_USERS = 1
BEGIN
```

```
SET @VALID_IND_PAR = 4
END


SELECT USER_NM, USER_DOMAIN,
CASE WHEN VALID_IND = 3 THEN 'VALID' ELSE 'INVALID' END
FROM L_USERS
WHERE (VALID_IND = 3 OR VALID_IND = @VALID_IND_PAR)
AND ( SUBSTRING(USER_DOMAIN, LEN(USER_DOMAIN), 1 ) = @DOMAIN_CODE

     OR @DOMAIN_CODE IS NULL )

AND @USER_PID <> USER_PID
END
GO
GRANT EXECUTE ON dbo.SP_GET_ALL_USERS_V3 TO IPSUsers
```

Running the report

After updating the stored procedure in the database we can follow the same process defined to execute the previous option for version 3.1 of the report and as expected, we will obtain exactly the same result set as we did in report version 3.1, with the shortcut.

Appendix B: Free reporting tips

This appendix contains a number of tips that may be useful when creating free reports.

1. Store the SP (stored procedure) in a file, with a template (provided separately)

a. The template guarantees that all IPS users can run the report

b. Script ready to deploy the SP on various databases

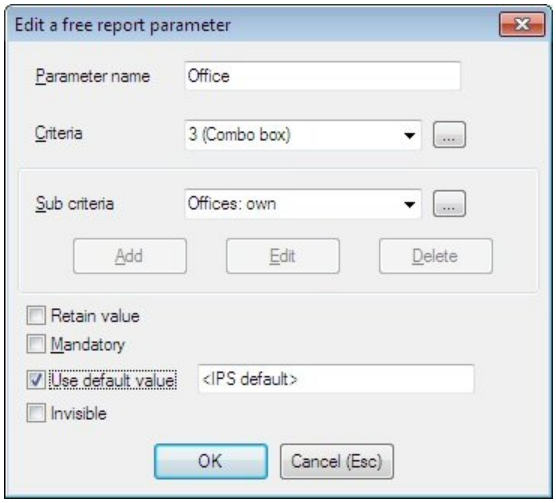
2. GMT/local date-time conversion. Many date-times are stored as GMT in the database, but report criteria are in local date-time, so you need to convert it in the SP sample script:

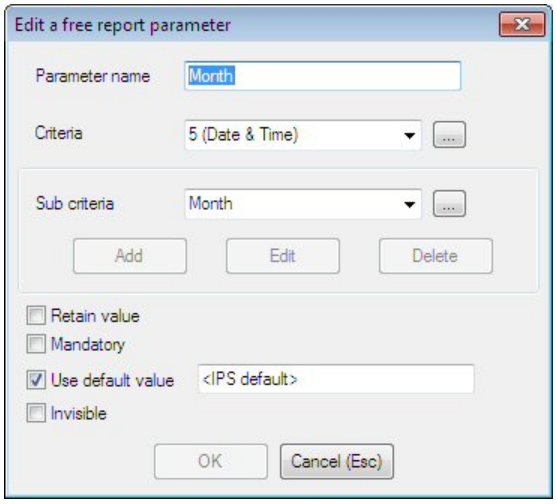
```
declare @diff int

    set @diff = datediff(mi, GETDATE(), GETUTCDATE())

    set @date_start = dateadd(mi, @diff, @date_start)

    set @date_end = dateadd(mi, @diff, @date_end)

    set @date_end = dateadd(dd, 1, @date_end)
```

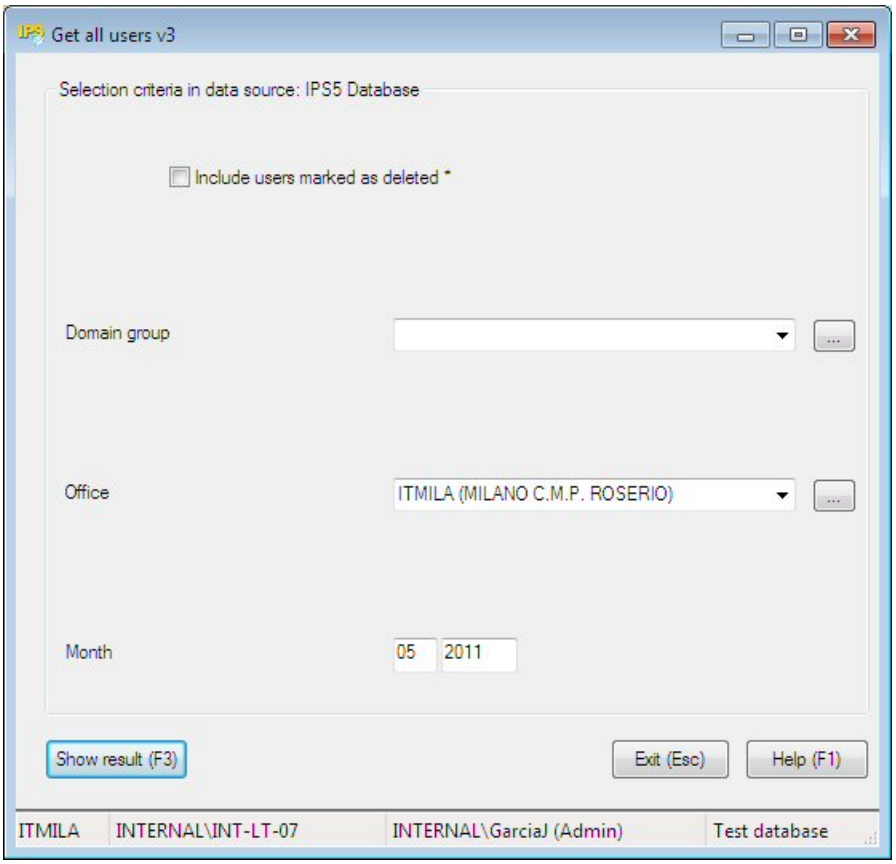
3. The report requirement must be clear and detailed before implementation starts.

4. A prototype of the report screen with the different report criteria/parameters is very easy to create and useful for reaching an exact solution before implementing the stored procedure, which normally involves the heaviest workload. Note that in this guide all examples have first defined the prototype screen, and secondly shown the stored procedure.

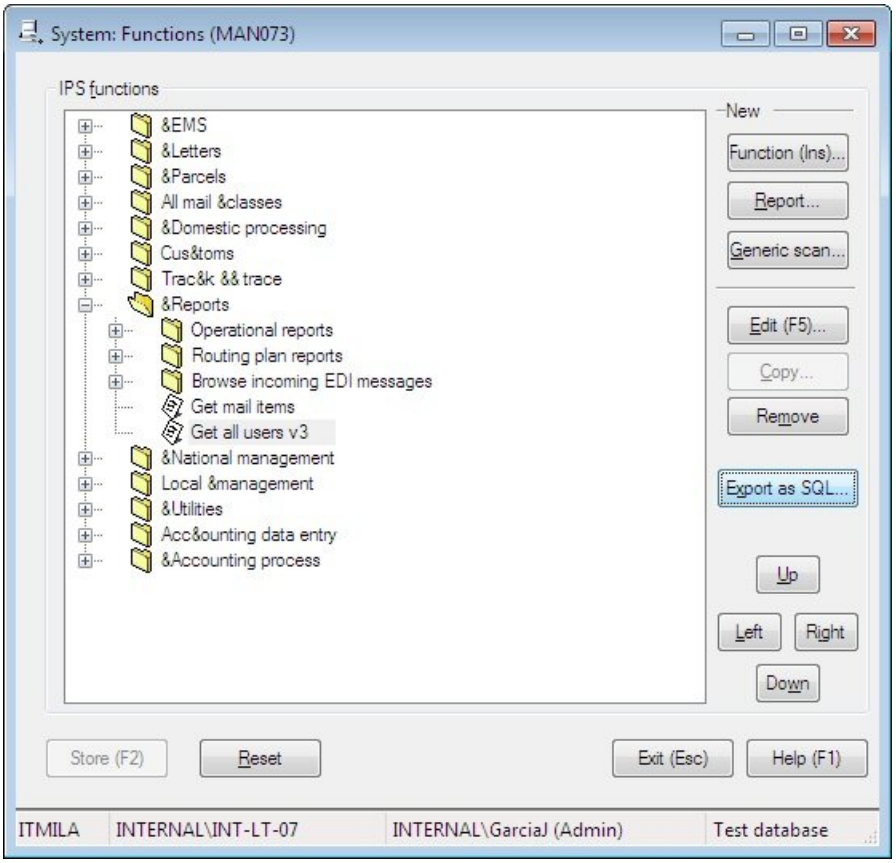
5. The command string; "<IPS default>" in the "Use default value" field of the parameter's configuration screen, can be used to retrieve the current IPS user's values for the following combo boxes: offices, countries, languages, currencies, users, workstations. If used with a "Date & time" control, it will use the current date and time as default value.







6. When the report is finalized, export the report as an SQL script. Keep this script safe as a reference in case the live report is altered. In order to export the report as an SQL script, you will need to access the Local management *tab->System subfolder->*Functions (MAN073) screen. Once there, select the report, and click the **Export as SQL...** button.



The resulting script can then be used to recreate the defined report on another database (i.e. moving the report from the test DB to the production DB, across local DBs, etc.).

Example SQL script exported from a free report

```
-----------------------------------------------------------------------

-- FILE GENERATED BY IPSv511 FOR THE FREE REPORT 'Get all users v3'

-- The new free report will be created under REPORTS tab

-- Run this script on the IPS database using SQL Query Analyzer

-----------------------------------------------------------------------

-----------------------------------------------------------------------

-- Check if the following stored procedures exist on your IPS database

-- If they do not, you need to create them manually!

-- SP_GET_ALL_USERS_V3

--SP_USER_DOMAIN_GROUPS


-----------------------------------------------------------------------

-- Criteria subtypes

DECLARE @crit_subtype_param25 T_CRITERIA_TYPE_ID

SELECT TOP 1 @crit_subtype_param25 = RPT_CRIT_SUBTYPE_ID FROM C_RPT_CRITERIA_
SUBTYPES WHERE RPT_CRIT_TYPE_ID = 4 AND RPT_CRIT_SUBTYPE_NM = 'Domain group list'

IF @crit_subtype_param25 IS NULL
```

```
BEGIN

     SELECT @crit_subtype_param25 = MAX(RPT_CRIT_SUBTYPE_ID) + 1 FROM C_RPT_
     CRITERIA_SUBTYPES WHERE RPT_CRIT_TYPE_ID = 4

     INSERT INTO C_RPT_CRITERIA_SUBTYPES VALUES(4, @crit_subtype_param25,
     'Domain group list', 'SP_USER_DOMAIN_GROUPS', '%1 (%2)', 1)

END


-----------------------------------------------------------------------

-- New function

DECLARE @new_function_id T_FUNCTION_IDENTIFICATION
DECLARE @new_function_sq T_FUNCTION_SEQUENCE_NUMBER

SELECT @new_function_id = MAX(FUNCTION_ID) + 1 FROM C_FUNCTIONS WHERE FUNCTION_ID
>= 3000 and FUNCTION_ID < 5000
if @new_function_id is null set @new_function_id = 3000
SELECT @new_function_sq = MAX(FUNCTION_SEQUENCE_NO) + 1 FROM C_FUNCTIONS WHERE
PARENT_FUNCTION_ID = 8
INSERT INTO C_FUNCTIONS VALUES(@new_function_id, 'FRT' + CONVERT(VARCHAR(10), @new_
function_id),'Get all users v3', 8, @new_function_sq, NULL,
'SP_GET_ALL_USERS_V3', '0#1#0', 'N', NULL)
INSERT INTO L_FUNCTIONS_U_GROUPS VALUES(@new_function_id, 'DEFAULT', 7)
INSERT INTO L_FUNCTIONS_W_GROUPS VALUES(@new_function_id, 'DEFAULT')


-----------------------------------------------------------------------

-- Function translations

INSERT INTO CT_FUNCTIONS(FUNCTION_ID, LANGUAGE_CD, LOCAL_FUNCTION_NM) VALUES(@new_
function_id, 'EN', N'Get all users v3')


-----------------------------------------------------------------------

-- Report parameters and columns

INSERT INTO N_FREE_REPORT_PARAMETERS VALUES(@new_function_id, 266, 1, NULL,
'Include users marked as deleted', 0, 1, NULL, 1, 1)
INSERT INTO N_FREE_REPORT_PARAMETERS VALUES(@new_function_id, 267, 4, @crit_sub-
type_param25, 'Domain group', 1, 0, NULL, 1, 1)
INSERT INTO N_FREE_REPORT_PARAMETERS VALUES(@new_function_id, 268, 3, 29, 'User',
3, 0, '<IPS default>', 0, 0)
INSERT INTO N_FREE_REPORT_COLUMNS VALUES(@new_function_id, 407, 'USER NAME', 50, 0,
0, 0, 0, NULL)
INSERT INTO N_FREE_REPORT_COLUMNS VALUES(@new_function_id, 408, 'DOMAIN', 20, 1, 0,
0, 0, NULL)
INSERT INTO N_FREE_REPORT_COLUMNS VALUES(@new_function_id, 409, 'VALID INDICATOR',
10, 2, 0, 0, 0, NULL)

-----------------------------------------------------------------------

-- Report parameters and columns translations
```