SAPtips
www.SAPtips.com
SAPtips
February 2003 Volume I Issue 1

Optimizing your HR Implementation with Custom Objects

by Greg Robinette, Robinette Industries and Consulting

Editor's Note: It's not a heck of a lot of fun to do costly custom programming in the HR module, especially when you know you're going to lose all that work the next time you upgrade. But now there's a more affordable and effective solution: the use of custom objects. In this valuable tip loaded with step-by-step screen shots, HR Editor Greg Robinette tells us how to use custom objects to get the most out of the HR module. The best part: as Greg assures us, you don't need to be a hardcore ABAP programmer to do it! All who know their way around HR configuration tables can utilize the custom objects functionality Greg describes within.

What are custom objects and why would you want to use them?

Imagine this scenario: The "go live" date has finally come and all of your HR functionality is perfect!

Well, almost perfect – there are a few key issues to keep in mind as you approach this critical phase:

- Your users thought the organizational management functions would provide divisional views and reporting - even though there are no official organization structures defining a division.
- They may have thought that everyone who is responsible for a work center, even unofficially, will appear as a work center supervisor.
- Your final problem is in the career-planning department. They love the career and succession planning, but want to designate key persons and positions in addition to the standard functionality.

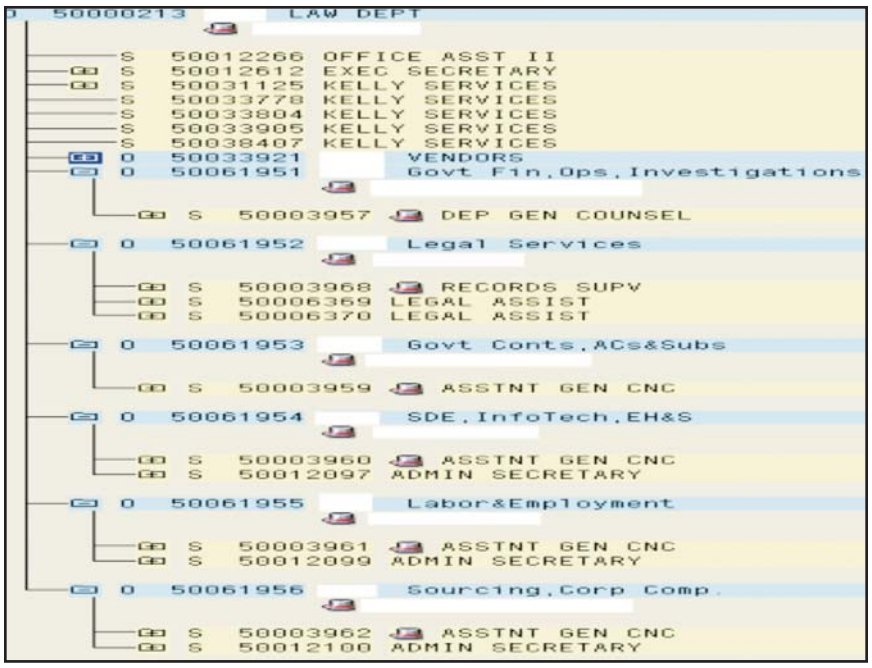

Figure 1: A basic organizational structure

In some installations, these issues would be -addressed through custom programming. However, a more efficient and effective way to address issues like these is to utilize the custom object functionality in the organizational management area.

This article will provide a brief overview of organizational management and then illustrate the creation and utilization of custom objects. This article assumes an understanding of the basics of SAP®'s organizational management functionality. Further information on organizational management is available in my SAPtips White Paper on Organizational Management, or from the SAP HelpPortal.

Basics of Organizational Management

Organizational management provides the tools to define and present a company's organizational structure. This structure can be used, through integration, to link people to the organizational attributes that help define their value in the organization. See Figure 1.

SAP uses organizational management objects and the relationships between those objects to drive organizational management functionality. SAP also provides the ability to create and use custom objects for those business conditions that are not in the delivered organizational management functionality. These custom objects are created in the same

Optimizing continued on page 2

SAPtips

Optimizing continued from page 1

tables and follow the same general rules as the standard objects, but will have the company's unique stamp on them.

The core component of all organizational management functionality is the object. Throughout the article, we will be referring to functional objects, not technical objects. (The greatest difference being functional objects are configured and technical objects are generally handled programmatically.) The main table containing the objects is the HRP1000 table. All the system objects are contained in this table. SAP delivers many SAP objects that are used primarily for HR functionality but also play a key part in security, workflow, and linking external components. See Figure 2.

It is important not to change these objects unless you are very sure of what effects your actions will have on the system. SAP has many functional modules, programs, BAPIs, and other programming components, which may utilize these objects. Also, keep in mind that even when changes to these standard objects have no effect today, they could very well limit or prevent an organization from utilizing new functionality in the future.

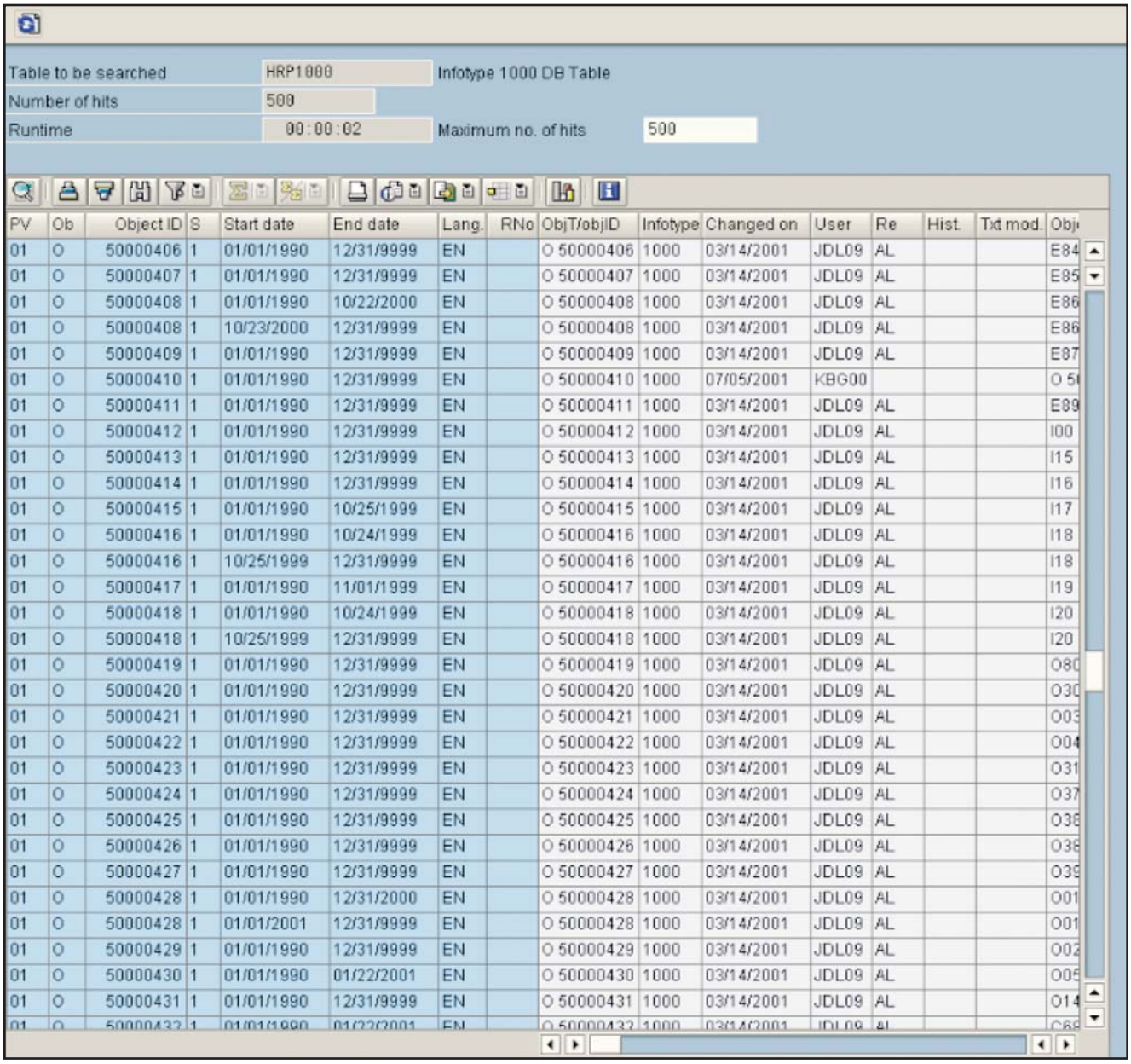

Table to be searched: HRP1000 — Infotype 1000 DB Table
Number of hits: 500
Runtime: 00:00:02 — Maximum no. of hits: 500

PV	Ob	Object ID	S	Start date	End date	Lang.	RNo	ObjT/objID	Infotype	Changed on	User	Re	Hist.	Txt mod.	Obj
01	O	50000406	1	01/01/1990	12/31/9999	EN		O 50000406	1000	03/14/2001	JDL09	AL			E84
01	O	50000407	1	01/01/1990	12/31/9999	EN		O 50000407	1000	03/14/2001	JDL09	AL			E85
01	O	50000408	1	01/01/1990	10/22/2000	EN		O 50000408	1000	03/14/2001	JDL09	AL			E86
01	O	50000408	1	10/23/2000	12/31/9999	EN		O 50000408	1000	03/14/2001	JDL09	AL			E86
01	O	50000409	1	01/01/1990	12/31/9999	EN		O 50000409	1000	03/14/2001	JDL09	AL			E87
01	O	50000410	1	01/01/1990	12/31/9999	EN		O 50000410	1000	07/05/2001	KBG00				O 5
01	O	50000411	1	01/01/1990	12/31/9999	EN		O 50000411	1000	03/14/2001	JDL09	AL			E89
01	O	50000412	1	01/01/1990	12/31/9999	EN		O 50000412	1000	03/14/2001	JDL09	AL			I00
01	O	50000413	1	01/01/1990	12/31/9999	EN		O 50000413	1000	03/14/2001	JDL09	AL			I15
01	O	50000414	1	01/01/1990	12/31/9999	EN		O 50000414	1000	03/14/2001	JDL09	AL			I16
01	O	50000415	1	01/01/1990	10/25/1999	EN		O 50000415	1000	03/14/2001	JDL09	AL			I17
01	O	50000416	1	01/01/1990	10/24/1999	EN		O 50000416	1000	03/14/2001	JDL09	AL			I18
01	O	50000416	1	10/25/1999	12/31/9999	EN		O 50000416	1000	03/14/2001	JDL09	AL			I18
01	O	50000417	1	01/01/1990	11/01/1999	EN		O 50000417	1000	03/14/2001	JDL09	AL			I19
01	O	50000418	1	01/01/1990	10/24/1999	EN		O 50000418	1000	03/14/2001	JDL09	AL			I20
01	O	50000418	1	10/25/1999	12/31/9999	EN		O 50000418	1000	03/14/2001	JDL09	AL			I20
01	O	50000419	1	01/01/1990	12/31/9999	EN		O 50000419	1000	03/14/2001	JDL09	AL			O80
01	O	50000420	1	01/01/1990	12/31/9999	EN		O 50000420	1000	03/14/2001	JDL09	AL			O30
01	O	50000421	1	01/01/1990	12/31/9999	EN		O 50000421	1000	03/14/2001	JDL09	AL			O03
01	O	50000422	1	01/01/1990	12/31/9999	EN		O 50000422	1000	03/14/2001	JDL09	AL			O04
01	O	50000423	1	01/01/1990	12/31/9999	EN		O 50000423	1000	03/14/2001	JDL09	AL			O31
01	O	50000424	1	01/01/1990	12/31/9999	EN		O 50000424	1000	03/14/2001	JDL09	AL			O37
01	O	50000425	1	01/01/1990	12/31/9999	EN		O 50000425	1000	03/14/2001	JDL09	AL			O38
01	O	50000426	1	01/01/1990	12/31/9999	EN		O 50000426	1000	03/14/2001	JDL09	AL			O38
01	O	50000427	1	01/01/1990	12/31/9999	EN		O 50000427	1000	03/14/2001	JDL09	AL			O39
01	O	50000428	1	01/01/1990	12/31/2000	EN		O 50000428	1000	03/14/2001	JDL09	AL			O01
01	O	50000428	1	01/01/2001	12/31/9999	EN		O 50000428	1000	03/14/2001	JDL09	AL			O01
01	O	50000429	1	01/01/1990	12/31/9999	EN		O 50000429	1000	03/14/2001	JDL09	AL			O02
01	O	50000430	1	01/01/1990	01/22/2001	EN		O 50000430	1000	03/14/2001	JDL09	AL			O05
01	O	50000431	1	01/01/1990	12/31/9999	EN		O 50000431	1000	03/14/2001	JDL09	AL			O14
01	O	50000432	1	01/01/1990	01/22/2001	EN		O 50000432	1000	03/14/2001	JDL09	AL			C68

Figure 2: Basic view of HRP1000 for the organizational unit object - O

Optimizing continued on page 3

Custom objects are a great way to extend SAP organizational management functionality

As important as the basic objects are, the relationships between these objects are equally important. These relationships allow the creation of the hierarchal view that truly depicts a corporate structure. These relationships are contained in the HRP1001 table. Very much like the SAP objects, there is a standard set of relationships delivered by SAP. The same caution should be used when contemplating changing these as was noted about the SAP delivered objects.

Custom Objects in Organizational Management

Custom objects are a great way to extend SAP organizational management functionality. They can be set up through configuration tables. They can be used without having to write custom ABAP routines. They can be reported on through delivered SAP tools such as Ad-hoc query in 4.6c and ABAP query in earlier versions.

Custom objects are defined and created within the SAP HRP1000 object realm. Since they are part of a standard structure, they are supported through upgrades and support package applications with little or no technical problems. They can use existing relationships, or have new ones created to support them.

The following "case study" demonstrates a practical application of the custom objects' functionality. While the scenario is simple, it illustrates the basic methods of using the custom object. The business requirement for this scenario has been simplified — to help us make our points.

Scenario: An SAP user requires the jobs in organizational management to be grouped into functional job families. This will aid in reporting and career development. Since there is no delivered SAP functionality to associate jobs to each other, a custom object will be defined and created to support this need.

The basic steps of creating a custom object are:

1. Define the custom object and make it available in HRP1000.

2. Define the relationships to the custom object, and any relevant conditions, such as a time constraint.

3. Provide a means to create the objects and deliver the functionality. These next steps are generic to creating any custom object.

For our "case study," we will use the implementation guide (IMG) menu paths and list the tables as we access them. Some users may be more at home using the table maintenance transactions SM30 or SM31. Either method is effective.

Start with transaction SPRO. We will be using the SAP reference IMG. Click the reference IMG button or press F5.

- Select the personnel management node.
- Select the organizational management node.
- Select the basic settings node.
- Select the data model enhancement node.

The data model node is the main area for configuring organizational management objects.

KEY POINT- Do not change any of the delivered SAP objects or relationships unless you are absolutely sure of the effects.

Execute maintain objects. This presents a maintenance view of tables T778O. See Figure 3. This is the main table of OM objects in the SAP R/3 system.

Custom objects can be used without having to write custom ABAP routines

Optimizing continued on page 4

Optimizing continued from page 3

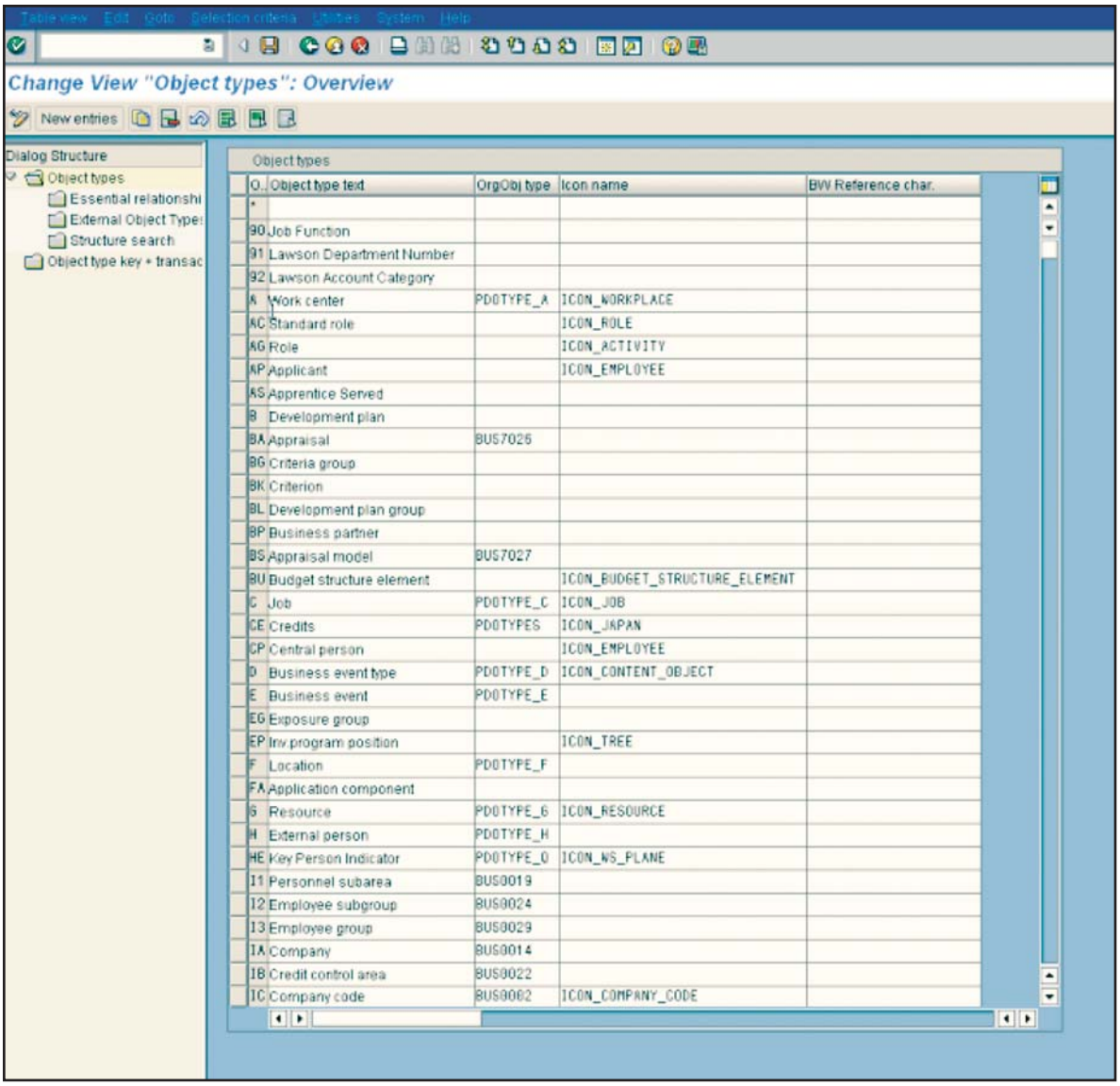

Figure 3: T7780 object maintenance table

Here is where we will define the custom objects. The New entries or copy icon can be clicked to define a new object. The new object will need a system-identifying, two-character object type name. This cannot be any two characters previously used. It's best if a naming convention is used. Z* or 9* are common methods of naming custom objects. The custom object's text is then added in the Object type text box. This name is what appears in the system.

The OrgObj type, Icon name, and the BW Reference char are not required for basic custom objects. See Figure 4. Once the basic custom object is created, you have the opportunity to create a required relationship. In our example, we will use the new object without an essential relationship requirement.

The next step is to setup the other info types for the new custom object. In the IMG under the "data model enhancement" node, open the "info type maintenance" node. Execute the "maintain info types" node.

Now we must set the time constraint for the new object. "Time constraint 1" means there can only be one object at a time, and there must be no gaps between them. "Time constraint 2" requires a record for the object, but allows gaps between records. "Time constraint 3" will allow the objects to exist as often as needed. Do not adjust time constraints on any object other than the custom object you are setting up. See Figure 5.

Optimizing continued on page 5

Optimizing continued from page 4

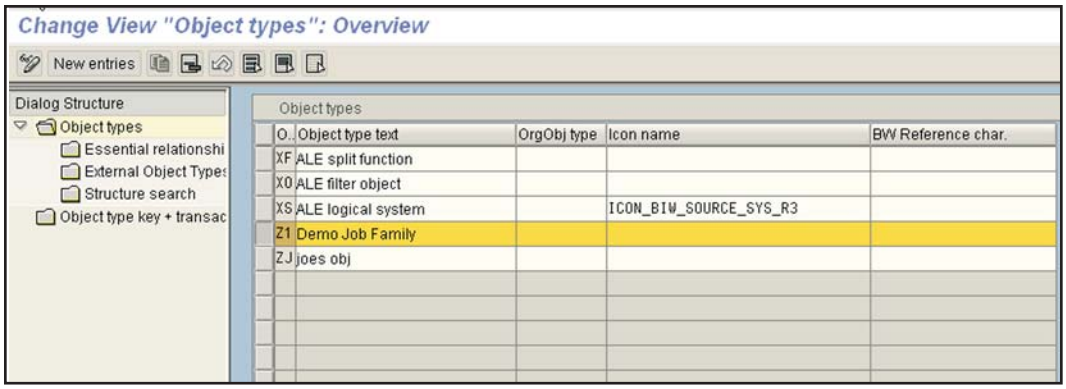

Figure 4: The new custom object

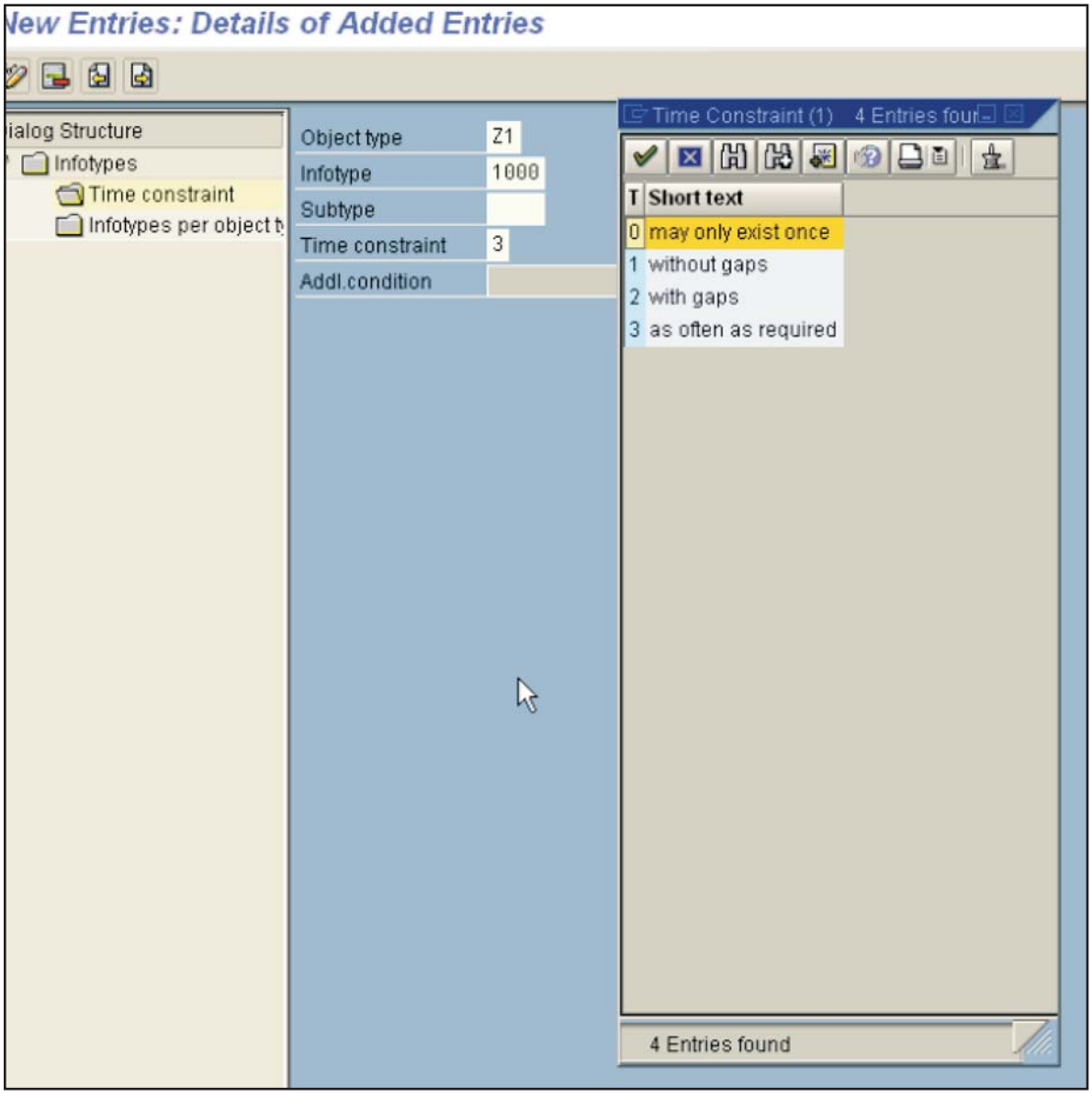

Figure 5: Setting the time constraint

Optimizing continued on page 6

Optimizing continued from page 5

For most custom objects, you will want to be able to create them without any restrictions, so a time constraint of "3" would be the normal setting. Next, the "info types per object" must be set up for the new object. Simply create a new entry for the custom object and select "it 1000" as the info type per object.

This customizing sets the basic structure for the new object. Now we need to set up what relationships will be allowed. This is the key to capturing the desired function with the new object. The requirements need to be mapped out before creating any relationships. This makes choosing the correct relationships easier. In our example, we will copy the SAP method of describing a position with a job. We will describe a job with our custom object DEMO JOB FAMILY. This will associate like jobs together, achieving our requirement.

Use the IMG again starting in the basic setting nodeà data model enhancementà Info type maintenanceà Execute maintain subtypes

Here we set the relationships subtype A007 and B007 to allow our new object Z1 the time constraint of 3 for sub type a007. You must set this up for each relationship associated with your custom object. Remember, you will need to set up the allowed reciprocal relationship for the object you want to relate to your new custom object. For our Z1 Demo job family, we only need to set the Z1 to job relationship A007, as the B007 is already allowed for the job object.

Next, move to the Relationship maintenance node. Here, execute "maintain relationships." Here we will select the 007 or "describes relationship." Next, we will select the "Allowed Relationships" folder to set up the Z1 object. See Figure 6.

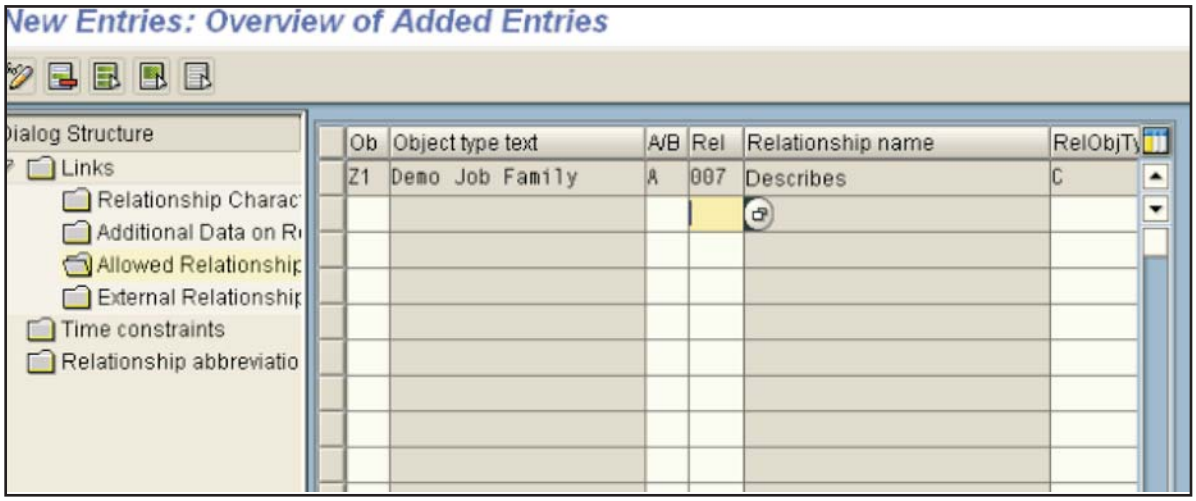

Figure 6: Relationship maintenance

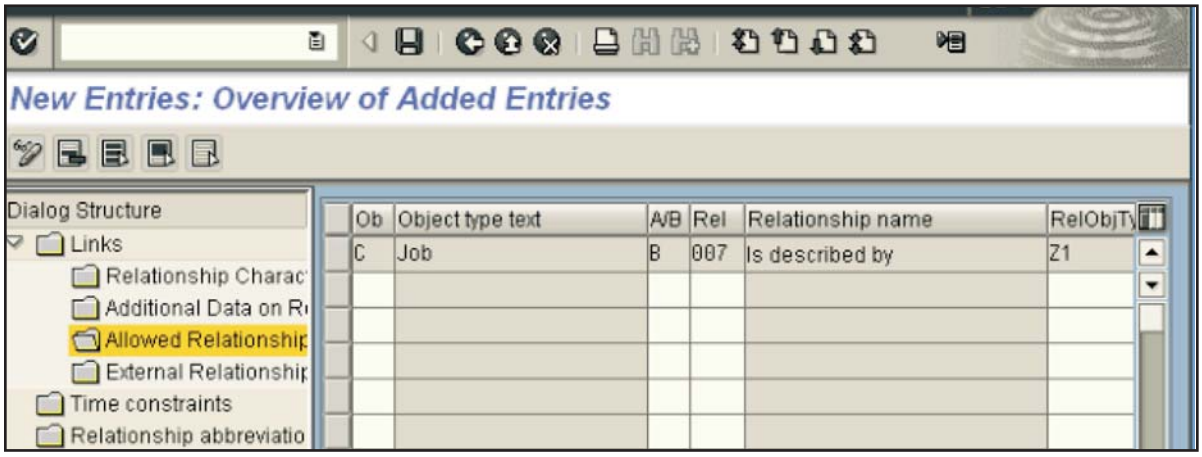

Figure 7: Defining the reciprocal relationship

Optimizing continued on page 7

Here we define the relationship 007 or the "Describes" relationship. We further define it with an "A"007 showing the new custom object- Z1 as describing the Job object- C. Now we must also define the reciprocal relationship B007. This shows the job object-C as being described by the new custom object- Z1. Every relationship must be defined and will always have a reciprocal. See Figure 7.

This provides the basic customizing required for the custom object. Once this is tested in the development environment, you will be able to create a transport to move it to the test/QA clients and ultimately to the productive clients.

Utilizing the custom object

Now the actual data objects can be created in SAP. The basic method for doing this will be transaction code PP01- Maintain object.

We will choose the new Z1 object – Demo Job Family from the Object type list, then select the Object under Infotype name. See Figure 8.

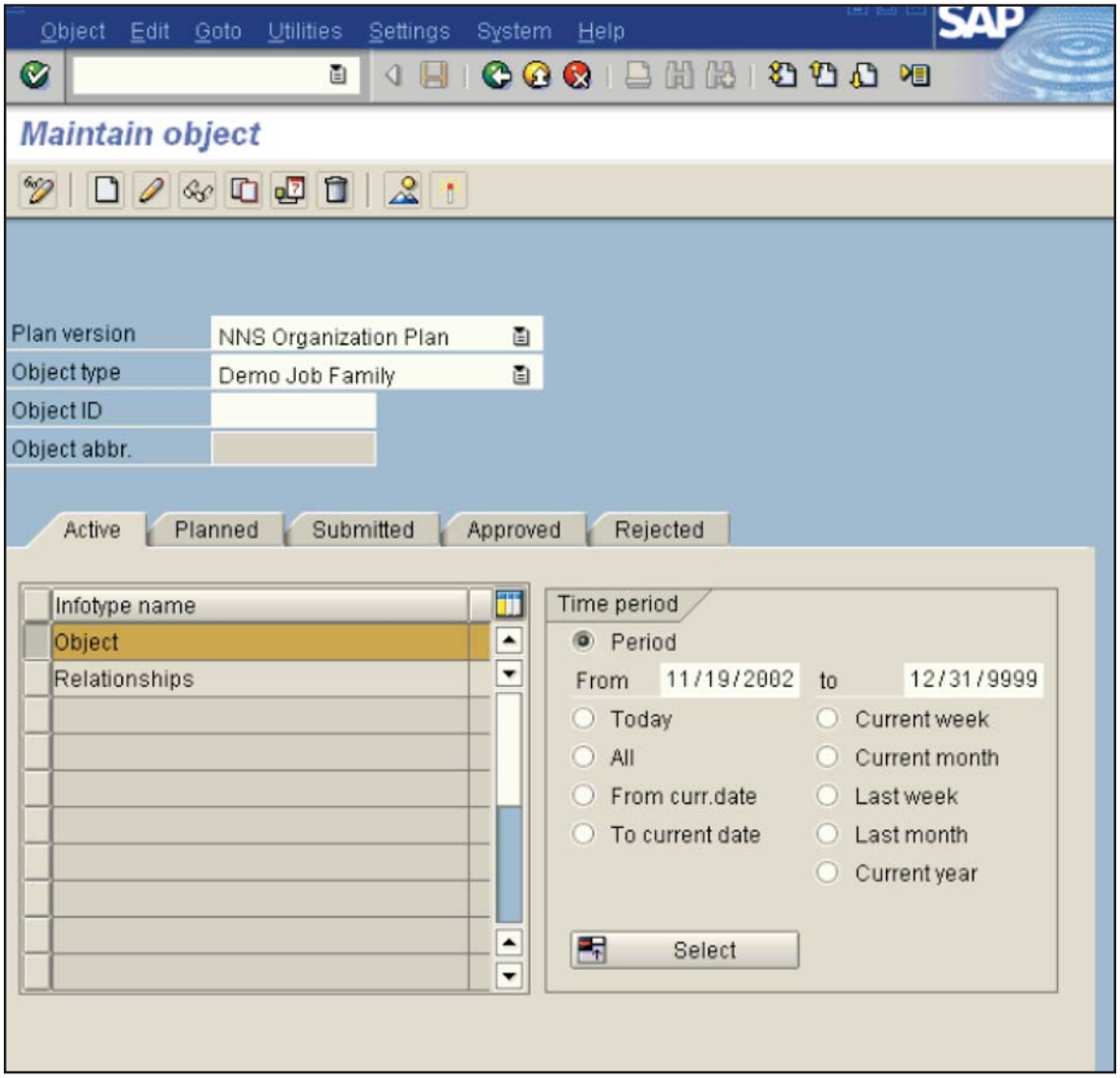

Figure 8: Object maintenance

SAPtips

Optimizing continued from page 7

You can create custom objects using future, past, or present dates

Please note the validity dates. You are able to create the objects using future, past, or present dates. Object creation generates an object number according to the system settings for the object number ranges. This number will be unique for the object with all the characteristics of the custom object associated with it. Set the validity desired and click the white "paper" icon or the "create" icon. This creates the object and brings up the screen to name the object and create the object text. After the object is created and named, you will need to create the relationships to the desired jobs. We will associate three jobs to this new object. Again, use PP01 but enter the object number of the new Demo Job Family object or choose the correct object from the Demo Job Family search function. (F4)

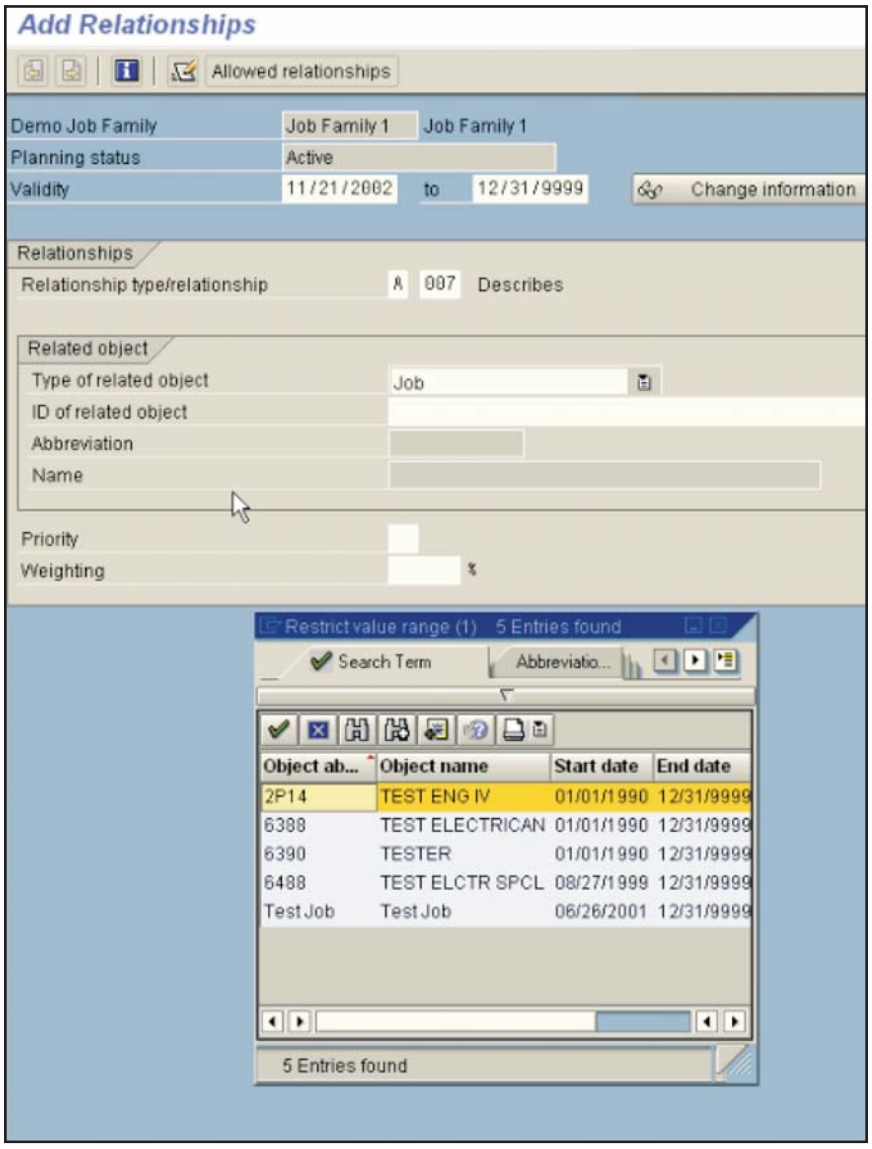

Figure 9: Relating custom objects to existing jobs

Choose the "relationships info type" and click on the "create" button. You will bring up the "Add Relationships" screen. Click on the "Allowed relationships" button. You will be shown the A007 "describes job" relationship as your only choice. Select this relationship. Click in the ID of related object field for a search function for jobs. Here you can enter a job name that you wish to include, and the system will find the correct object number. See Figure 9.

Select the correct job for the family. Enter and save. Repeat this for all the jobs in the job family. Now the job family will have all the associated jobs linked to it. It is available for reporting on, or for further ABAP development. Here is a sample report using standard Organizational Management reporting. See Figure 10.

This completes the basic steps for utilizing custom objects.

What I've just described is a basic view of the custom objects creation process. There are many "follow on" activities that we have not covered here due to time and space limitations. As a next step in custom object usage, explore the evaluation path functionality and object based reporting. These areas will open up further functionality for custom objects.

Optimizing continued on page 9

SAPtips

Optimizing continued from page 8

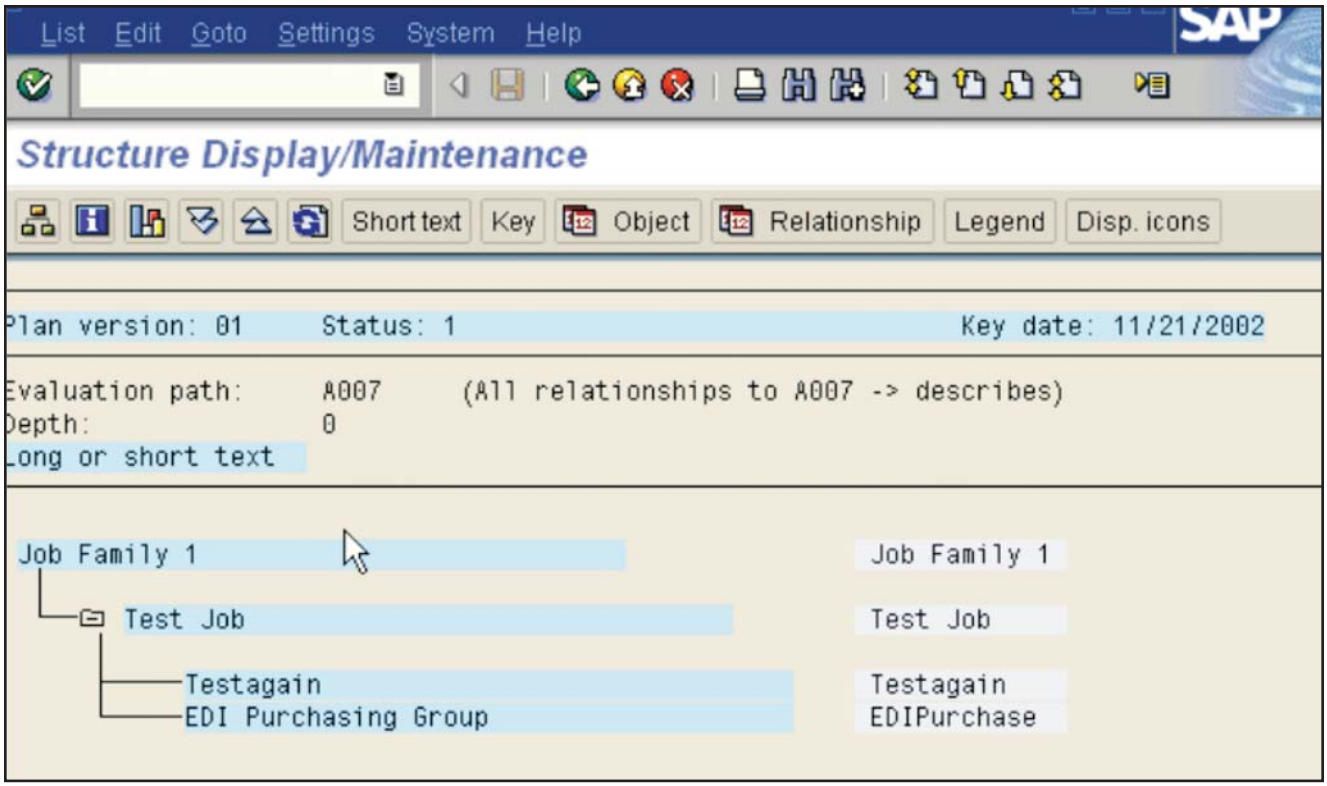

Figure10: Report displaying all jobs in the new Job Family 1

Here is a listing of the command line transaction codes used during the configuration of custom objects.

Transaction Code List

PPOME	Maintain organizational structure for version 4.6 and greater
PPOM	Maintain organizational structure for version 4.6 and earlier
PPOSE	Display organizational structure for version 4.6 and greater
PPOS	Display organizational structure for version 4.6 and earlier
SE16N	Data browser for tables
SPRO	Access implementation guide
PP01	Maintain objects

Here are the key configuration tables updated during the configuration of custom objects.

Configuration Tables

T778O	object type definition
T778T	object info type table
T777ZIT	time constraint tables for object info types
T777I	info types per object table
T778V	relationship characteristics table

Greg Robinette, *Robinette Industries and Consulting.* Greg is an SAP certified human resources application consultant. His most recent projects have involved de-coupling the HR module, upgrading from 3.1 to 4.6, and developing output management and reporting strategies. He has led projects in payroll in SAP and other software packages. Greg has a varied background including packaging system design, service, and installation, hotel engineering management, and business development for mechanical contracting and veterinary hospitals. Greg's email address is Greg.Robinette@SAPtips.com

Optimizing continued on page 10

SAPtips

Optimizing continued from page 9