User Defined Categories in PSNext 3.0

02/15/09 – Revision 1.0

Sciforma

Sciforma Europe: 10 rue Vercingétorix 75014 Paris – France
Tel : +33 (0) 1 56 54 83 00 – www.sciforma.fr – info@sciforma.fr

Contents

A -Introduction

PSNext power relies in the complete set of Project Management tools it provides to corporations. One of PSNext's major assets relies on the easy way users can customize it to fulfill their own needs.

Since PSNext was first released, all of the Project Management objects are represented as “categories” such as Project, Resource, Task, User, etc. For each existing “category”, a set of built-in fields is provided and users are able to create their own fields (user defined fields) to customize the definition of each category as well as the spreadsheets and forms displaying them.

In PSNext 3.0 the customization possibilities go far beyond creating fields within a built-in category; also users are now able to create their own categories to describe any object they would like to manage such as risks, invoices, purchase orders, etc.

User Defined Categories have all the power of a built-in category; new fields can be added, spreadsheet listings can be designed, reports can be used to consult their records, etc.

The data model structure is now open for users to extend it. Since records of a user defined category are saved in a readable database table, it is really simple to provide full integration capabilities with other corporate IT systems.

Understanding of Data Views in PSNext is essential before you keep reading this paper. For further information on this subject you can read the “Data Views in PSNext 3.0” white paper.

B -Definition

Users are now able to extend the PSNext's data model by creating their own categories. A User Defined Category is based on a “Data View Definition”. Every existing “Data View Definition” saved within the PSNext domain's database (domain is not overridden in the data view definition) is now considered to be a user defined category (UDC).

1 -UDC's fields

As done within a built-in Category, within a UDC, users can create user defined fields (UDF) of any type (cost, rate, date, etc) as well as formulas and filters. Dated and Distributed fields can be created for a UDC by using View Based Fields (see "Fields Enhancements for further details). Users can also set each field's formatting options and control access to their values through permissions.

Fields of User Defined Categories can be managed in System/Fields, every available UDC is listed next to the Built-in categories.

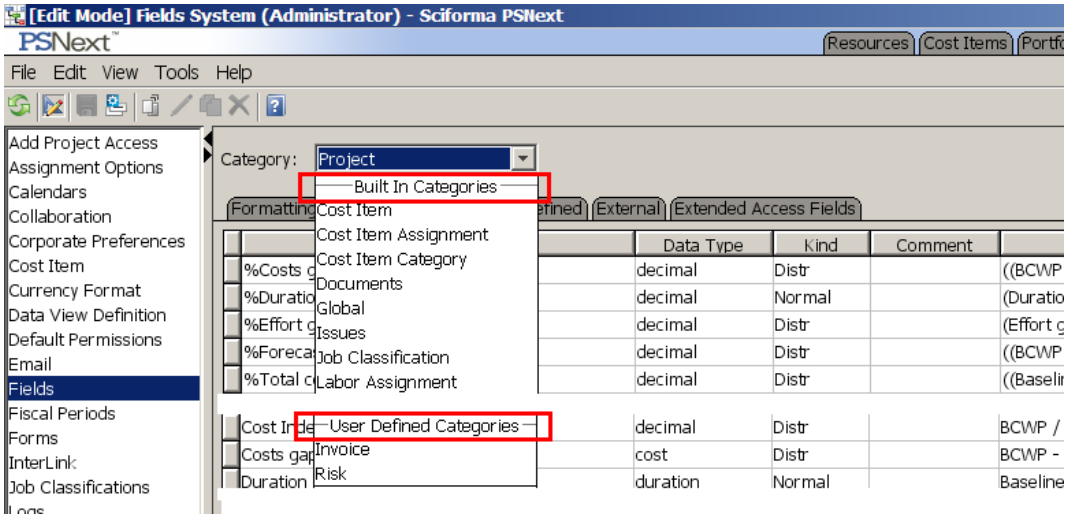

Fig 1. User Defined Categories Fields can be managed in System/Fields

Note that any existing column in the related "Data View definition" marked as "Visible" will already be considered to be a User Defined Field of the UDC.

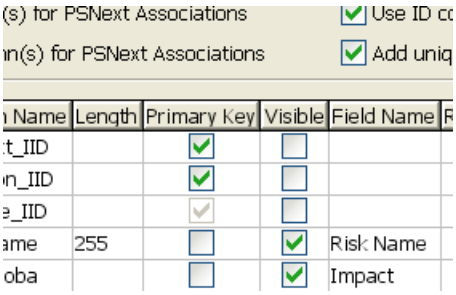

Fig 2. Columns marked as "Visible" in the View definition dialog box.

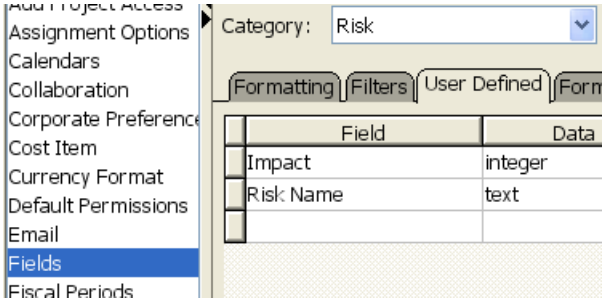

Fig 3. UDFs of a UDC correspond to a View definition's visible column.

Additional User defined fields can either be created from System/Fields or from the related Data View definition (System/Data View Definitions).

Only those columns marked as visible will be available as fields elsewhere in PSNext.

2 -Relationships

User Defined Categories are used to describe an object that does not exist as a PSNext built-in structure such as an Invoice, a Purchase order, a Risk, a Customer, etc.

Besides describing the properties (fields) of a UDC, it is essential to determine the relationship it will have with other data structures of PSNext. A UDC can rather be related to a built-in category or to another UDC.

2.1 -Relationships with built-in Categories

PSNext built-in categories such as Projects, Tasks and Resources have a related database table describing them. Within these tables some columns are used by PSNext to clearly identify the record.

For instance a Project record will contain an Internal_ID and a Version_ID so that PSNext knows what project and version the record is representing. These fields are generally defined as keys in the Table's definition and will allow you to establish relationships between a built-in object and a UDC.

When creating the Data View Definition of a UDC you can include the IDs (internal or external) of a built-in category to provide the UDC with enough information of the built-in object it will be related to.

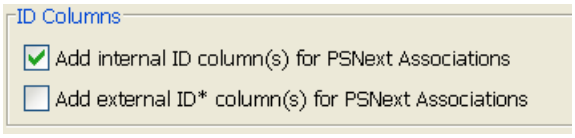

Fig 4. Including internal and/or external Ids into the view definition create a relationship with a built-in category.

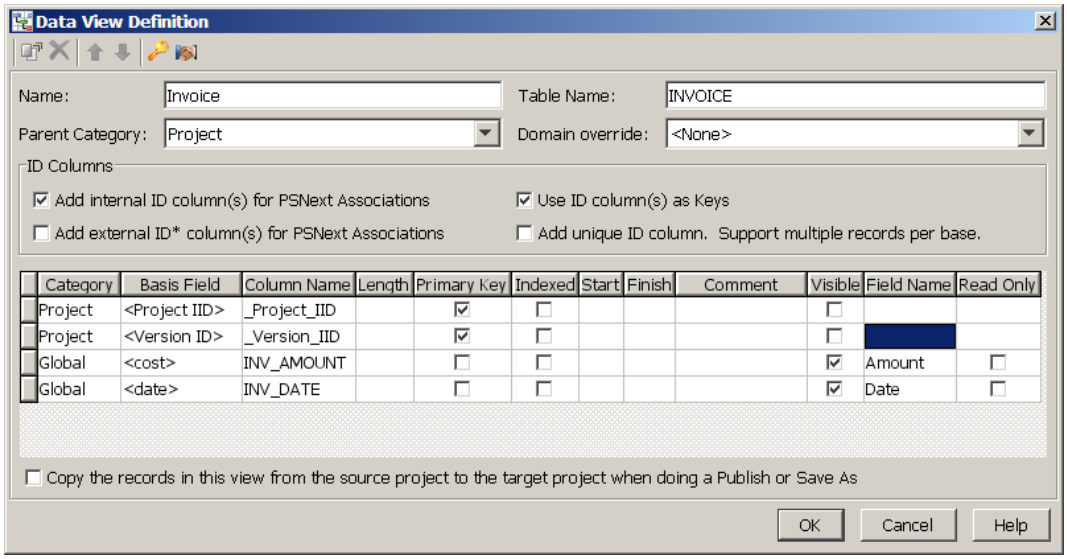

Fig 5. The Invoice UDC could be related to the Project category since it contains information on the Project and Version Ids

2.2 -Reference fields

Reference fields allow users to consult data from related categories. For example from the “Project” category users can access the “Tasks” related to the “Project” by using the [Task] reference field in a Project formula (reference fields are normally surrounded by squared brackets)

As soon as a UDC has enough information about a built-in category (the related ID columns are included), PSNext will automatically create the reference fields in both the built-in category and the UDC so that access can be done from one to the other (from the UDC to the built-in category and vice versa).

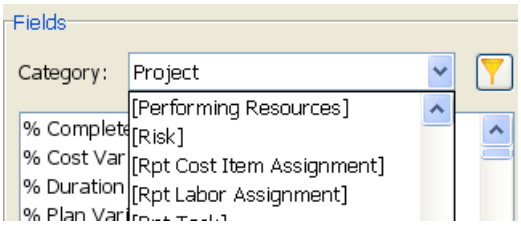

Fig 6. Reference fields from “Project” to the “Risk” UDC.

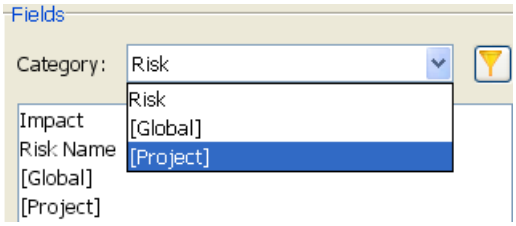

Fig 7. Reference fields from the “Risk” UDC to the “Project” built-in category.

Table 1 is a summary of the required Ids that a UDC's data view definition should include establish a relationship with a built-in category. Since a view definition can only have a single “Parent category”, a UDC can only include the Ids of a single built-in category.

Table 1: Required Ids to establish a relationship with a built-in category

Built-in category	Internal IDs	External IDs
Project	Project_IID	Project id
Task	Project_IID, Task_IID	Project id,Task Id
Labor Assignment	Project_IID, Task_IID, Res_IID	Project id, Task id, Resource id
Resource	Resource_IID	Resource id
Cost Item	Cost_IID	Cost id
Cost Item Assignment	Project_IID, Task_IID, Cost_IID	Project id, Task id, Cost id
Timesheet	Project_IID, Task_IID, Res_IID	Project id, Task id, Resource id

2.3 -1 TO 1 RELATIONSHIPS

We can say that a UDC is related with a built-in category in a 1 to 1 relationship when there is one record in the UDC for each PSNext built-in record.

Lets assume that each “Resource” has a related “Contract”. A 1 to 1 relationship would exist if there is a single record in the “Contract” UDC table for each Resource ID. A resource might have a single [Contract]. The 1 to 1 relationship is ensured by the fact that the Resource ID is the single Key within the the “Contract” UDC Data View Definition.

2.4 -1 TO N RELATIONSHIPS

These relationships are an extension of the 1 to 1 relationship. In fact we can say that there is a 1 to n relationship when there could be many records of a UDC related to the same built-in category. There is a 1 to n relationship when a UDC definition contains more keys that those required for establishing a 1 to 1 relationship with the built-in category.

Lets consider the following example: A project has one invoice per month. In this example, the “Invoice” definition would contain a “Project ID” (to link the “Invoice” category to the “Project” category) plus a “Month” value defined as a key (to ensure the uniqueness of the “Invoice” record per month.

Furthermore if a “Project” had one monthly invoice per provider, an additional column and key “Provider” should be added to the “Invoice” data view definition.

You can easily identify a 1 to n relationship when using reference fields from one category to another. In fact if you were to access an “Invoice” record from a “Project” formula, PSNext will ask for a summary operator such as sum, list, min, etc. because multiple records exist for the given project and they should be summarized before you can access them from a “Project”. For instance the “Project” formula: sum([Invoice].Price will return the total Price of every invoice of every month of every provider related to the project.

2.5 -PROJECT VERSIONS

Relationships with Project objects (both 1 to 1, or 1 to n) might be a little harder to understand due to version management. A Project in PSNext might exist in multiple versions (working, published, working 2, working 3, objective, objective 2, etc). A separate record is created for each project version in the PSNext built-in table containing projects. When relating a UDC to the “Project” category it is then important to decide on whether each project version will have a related UDC record(s) or not.

Lets pretend that there is an “Invoice” UDC that is related to the “Project” category (“Projects” have some related “Invoices”). Are “Invoice” records different for each project version? Or should every project, regardless its version, have a common set of “invoices”?. This decision is determined by including or not Project version information in the UDC's Data View Definition.

Category	Basis Field	Column Name	Length	Primary Key	Indexed	St
Global	<cost>	INV_AMOUNT		☐	☐	
Global	<date>	INV_DATE		☐	☐	
Project	<Project IID>	_Project_IID		☑	☐	
Project	<Version ID>	_Version_IID		☑	☐	
Project	<Portfolio IID>	_Portfolio_IID		☐	☐	

Fig 8. Version ID is included and marked as a key in the Data View Definition

Relationships to the following Project related categories can support versions: Project, Task, Task Step, Labor assignment, and Cost item assignment

If ever you need to include a version id in the definition of a UDC (and manage then multiple versions for each UDC record) PSNext will allow you to determine how to update these records. Since a new version of a project is created by using the Save as... command (except for the Published version that requires a "publish" to be done), it is then possible to determine if the records of the source version should rewrite those of the target version when executing any of those actions or not.

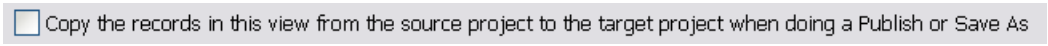

Fig 9. Control if records of a version-enabled UDC should be rewritten by the source records.

3 - UDC to UDC relationships

User defined categories can be related to each other such as built-in categories are. Projects are related to Tasks while Tasks are related to Labor Assignments, and Labor Assignments are related to Resources.

In order for a User Defined Category to be related with another User Defined Category, the common criteria relating them should be identified.

Tasks are related to Projects because in their internal definition they include the Project ID and Version ID of the Project the Task belongs to. This is how PSNext is able to identify the related tasks of a given project.

UDC to UDC relationships are created and managed in the Data View Definition of the UDC.

The "View to view" icon in the "Edit View definition" dialog box brings up the "View to View relationships" dialog box (Fig 10).

From this window users can create, edit, and delete relationships of the view with other existing views (UDCs).

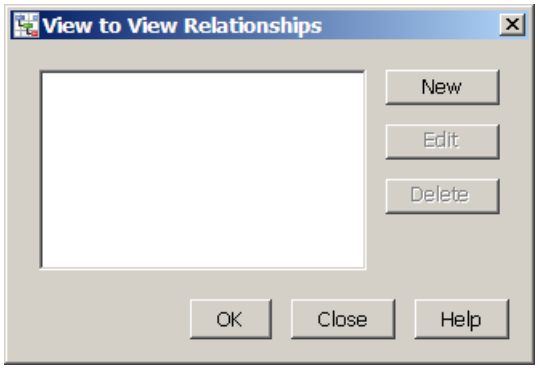

Fig 10. View to View relationships for the selected view

For each View to View relationship the field's mapping can be done. The "View to View reference field definition" dialog box allows the user to map the fields establishing the relationship between both Views. To help the user set the right field mappings PSNext will list by default all the table columns that are marked as keys in both views.

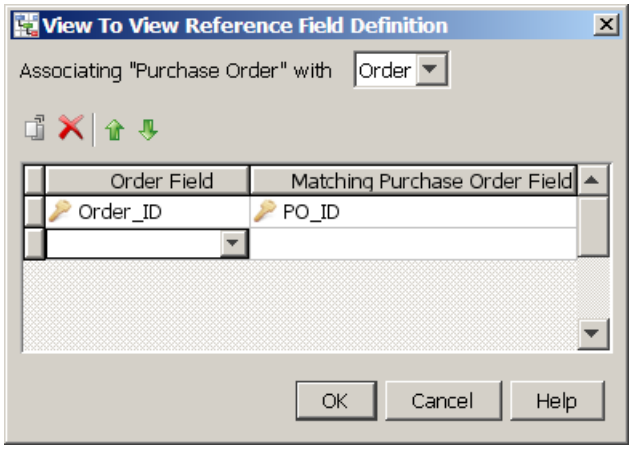

Fig 11. Keys are mapped in both views to establish the relationship

As a result of creating a View to View relationship, both views will now have Reference Fields pointing to each other. These reference fields can be used in formulas to access the content of the related view. If the referenced View had many entries per owner (1 to n relationship), then summary operators should be used to access the referenced data (sum, min, max, etc.).

4 - UDC records

Once a UDC has been declared as well as its relationships, you might want to provide a way to create, update, consult, and delete the records of the UDC. Create new Invoices, Purchase Orders, etc.

4.1 - Creating new UDC records

Remember that UDCs rely on a Data View Definition. The records could be created by a third party application into the related database table containing the UDC records.

“Invoice” records might be created by an Invoice system and would be readable from the PSNext interface by using the [Invoice] reference field to create project formulas referring to Invoice data and display them in Project interfaces (spreadsheets, forms) as well as in reports.

PSNext 3.0 provides as well the Grid control to create new UDC records. The Grid is inserted into a Form and is able to loop on any of the User Defined Categories providing a spreadsheet with the relevant UDC fields (as columns) and a blank line at the end of the spreadsheet for users to create a new record (as users are used to create a task in the Gantt spreadsheet).

To learn more about Grid Controls read “PSNext 3.0 User Interface”.

Note that when creating records of a UDC that contains a “Unique ID” column, a 1 to n relationship to the UDC is automatically created. Consider this column as an internally generated unique ID for each UDC record that is created.

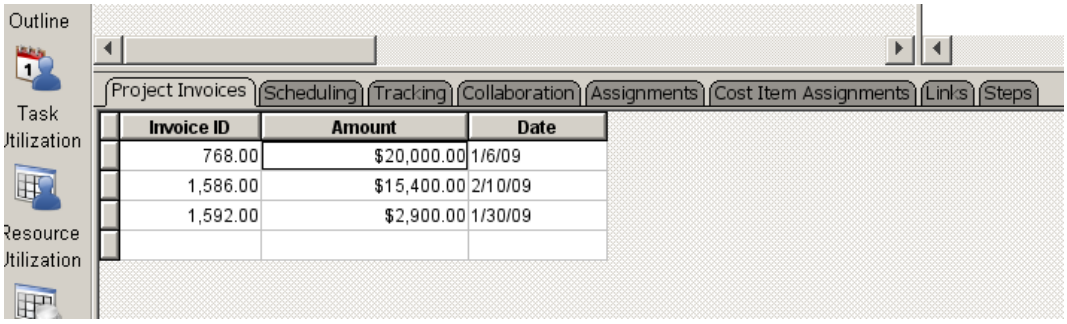

Fig 12. A Grid control inserted in a Project form is used to create UDC records.

4.2 -Update existing UDC records

No matter how the UDC record was created (a third-party system did or a PSNext user did by using a grid), it can be updated as well in many different ways. A grid listing the UDC records can be used to update the record's fields. Since the grid is inserted into a form that is already "typed" (project form, task form, resource form, etc.) the listed items (if related to the form's type) will already be filtered. For instance in a task form grid, only the UDC records for the selected task will be displayed if the UDC has a relationship with the Task category (see Relationships with built-in categories in this paper).

It is important to say that field permissions (ACLs) are honored when they are set for a given UDC's field. This means that even if the grid displays a field (column), a user might not be able to read or write its value. Additionally, when fields are to be updated only by a third-party system there is a property in the data view definition that can restrict a read-only access to a given field (no matter the permissions on the field).

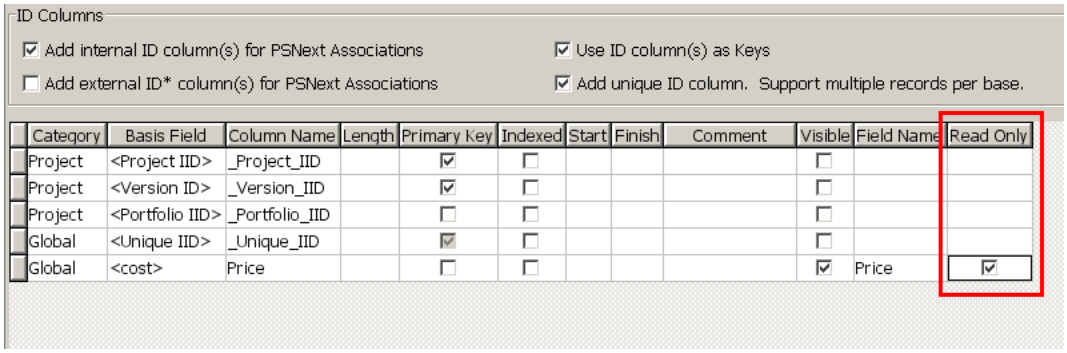

Fig 13. The "Price" field is marked as read-only in the data view definition.
It can only be updated directly in the database table.

4.3 -Consult UDC records

Depending on the purpose of the consultation, PSNext users can decide on the consult method that is more appropriate to handle the records of a UDC.

- Reports: UDCs are categories, and as so they are available in Reports design screens (text blocks, graphs, bubble charts, etc.). It is then really easy to display for instance all the purchase orders f a given project or graph the monthly invoices of all the projects within a Portfolio.
- Grids: As already detailed, grids are an easy way to display the records of a UDC within a Form.
- Database queries: UDC records are saved in a database table, queries to the database are a good way to consult UDC records by third-party systems.

4.4 -Delete records

Deletion of a UDC record within the PSNext interface can only be done by using a grid. A right click on a given row will bring up the context menu where you can choose the “Delete Record” command.

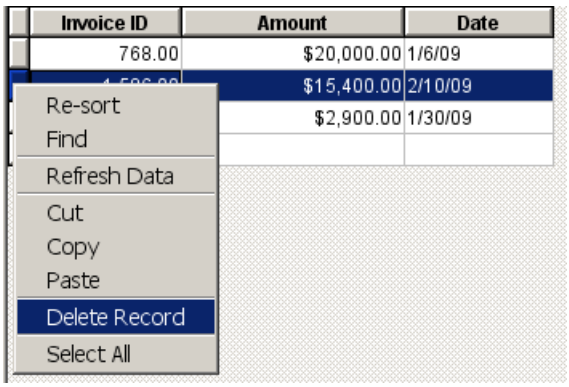

Fig 14. The context menu of a grid provides a “Delete Record” command

Otherwise the deletion of the record can be directly done in the UDC's related database table by a third-party querying the database.

Another important command of the context menu of the Grid control is the “Refresh data” command. Since UDC data behaves the same way “built-in data” does, once the project is open, refreshing the component (F5 or View/Refresh) will not reread data from the database. The "refresh data" grid's command provides a way to get fresh data from the data view if the user wants to get rid of all edits or absolutely knows that new data (perhaps from an external update) is now available. When data is to be updated or deleted directly in the database, this command allows the user to refresh the content of the grid by querying the database again.

C -Summary

The intention of this paper is to present the main technical features that PSNext 3.0 provides in terms of data model extension and customization. They are easy to use tools that allow users to design and implement into PSNext brand new objects according to their specific Project and Portfolio Management needs.

User defined categories have been designed to behave the closest as possible to a built-in category so that users can easily customize the related User Interfaces for data entry, fields customization as well as reports with the techniques PSNext users are used to work with.

Integrating UDCs with third-party systems is very simple since records are clearly exposed in a database table. Data exchanges are done in real time by keeping fresh data at any time.