Time sheets Add-On Solution

Concepts

Highlights

- Time Sheet Positions (relationship) are used to enter Activity descriptions, a date and the hours worked

- TimeSheet Report presents a summary of all recorded hours.

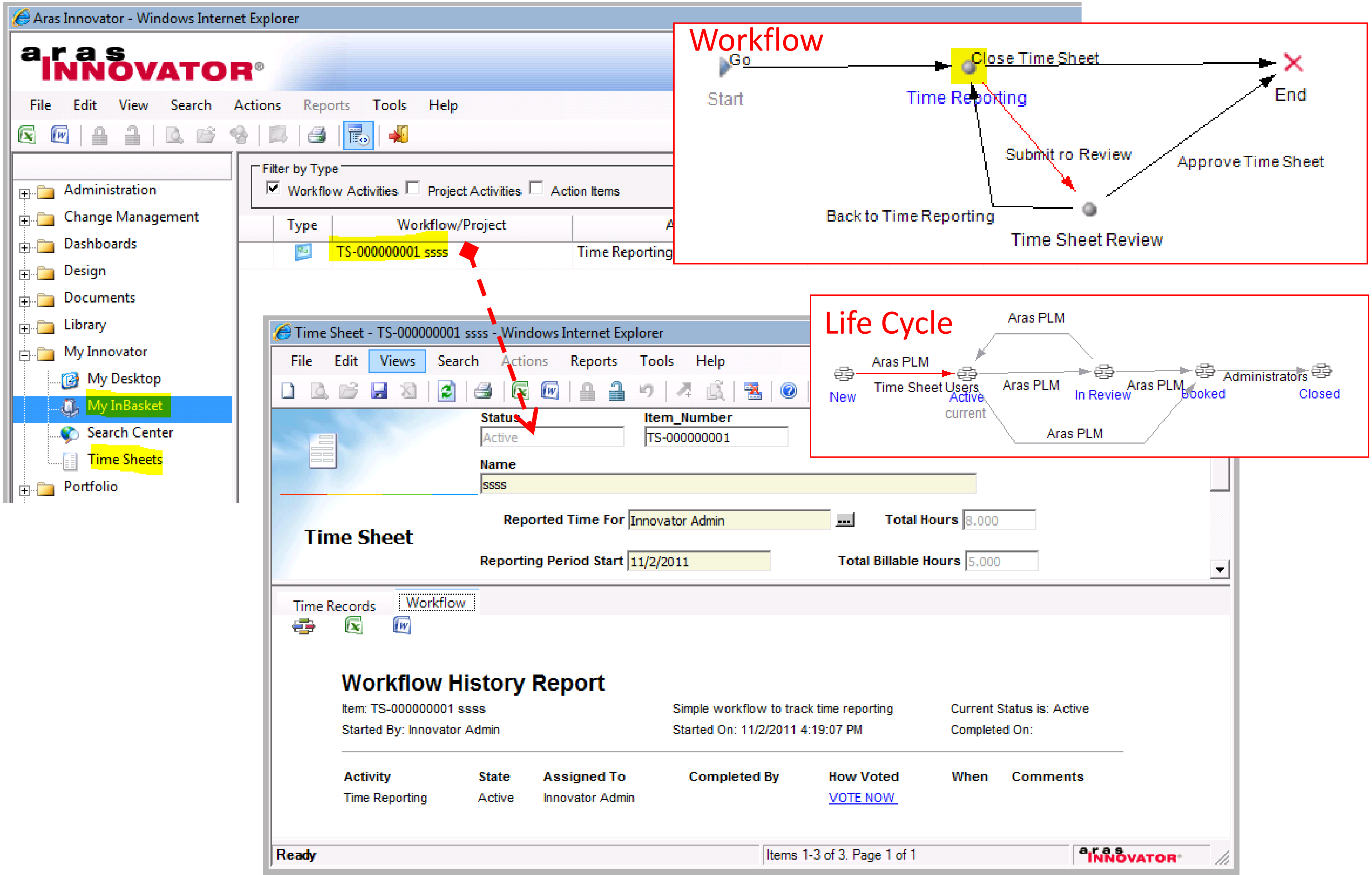
- Workflow drives the life cycle of a time sheet. Owner can submit it to review. WF activity will show in inBasekt of „Owner“ and „Reviewers“

- Auto creation of regular (ie. Weekly) time sheets for a defined list of users (members of Identity: „**Time Sheet Auto Create Weekly**“
  - Aras Innovator Service must be configured to trigger method „**Time Sheet AutoCreate Weekly**“ i.e. every Sunday. Method can be run manually by Administrators, as well

Installation

- Use the PackageImportExportUtilities to import the package „ Simple Time Sheet vX-X“
- Extract package to directory accessible to the import tool.
- Start the import tool, enter the URL to your Aras environment, and log on with „admin“.
- There are 2 import steps to run. Find the manifest file in folder „import1“, and „import2“ in the directory to which the package got extracted.

  → You MUST use the „Merge“ option.



Filter by Type
☑ Workflow Activities ☐ Project Activities ☐ Action Items

Type	Workflow/Project	A
	TS-000000001 ssss	Time Reporting

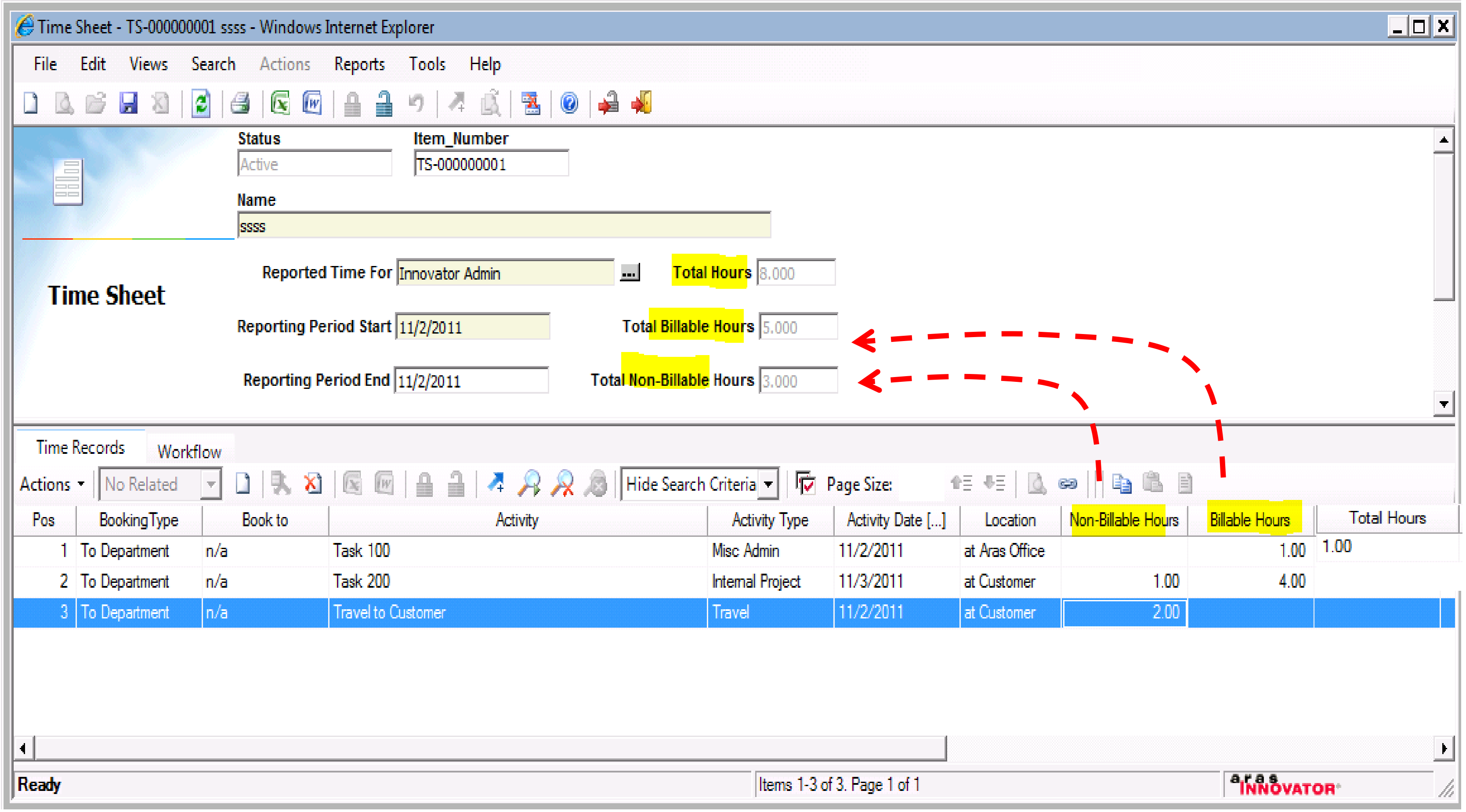
Status: Active
Item_Number: TS-000000001
Name: ssss
Reported Time For: Innovator Admin
Total Hours: 8.000
Reporting Period Start: 11/2/2011
Total Billable Hours: 5.000

Workflow History Report

Item: TS-000000001 ssss
Simple workflow to track time reporting
Current Status is: Active
Started By: Innovator Admin
Started On: 11/2/2011 4:19:07 PM
Completed On:

Activity	State	Assigned To	Completed By	How Voted	When	Comments
Time Reporting	Active	Innovator Admin		VOTE NOW		

Sample Solution – Time Sheet



Time Sheet - TS-000000001 ssss - Windows Internet Explorer
File Edit Views Search Actions Reports Tools Help
Time Sheet
Time Records Workflow
Actions No Related Hide Search Criteria Page Size:
Ready
aras INNOVATOR

Status: Active

Item_Number: TS-000000001

Name: ssss

Reported Time For: Innovator Admin

Total Hours: 8.000

Reporting Period Start: 11/2/2011

Total Billable Hours: 5.000

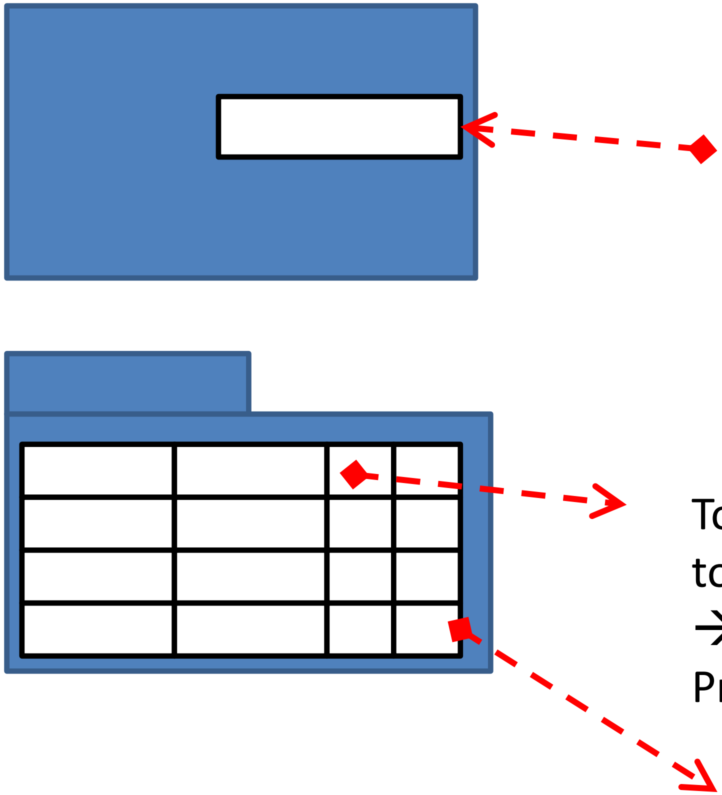
Reporting Period End: 11/2/2011

Total Non-Billable Hours: 3.000

Pos	BookingType	Book to	Activity	Activity Type	Activity Date [...]	Location	Non-Billable Hours	Billable Hours	Total Hours
1	To Department	n/a	Task 100	Misc Admin	11/2/2011	at Aras Office		1.00	1.00
2	To Department	n/a	Task 200	Internal Project	11/3/2011	at Customer	1.00	4.00	
3	To Department	n/a	Travel to Customer	Travel	11/2/2011	at Customer	2.00		

Items 1-3 of 3. Page 1 of 1

Using Grid Events to calculate values from a relationship grid

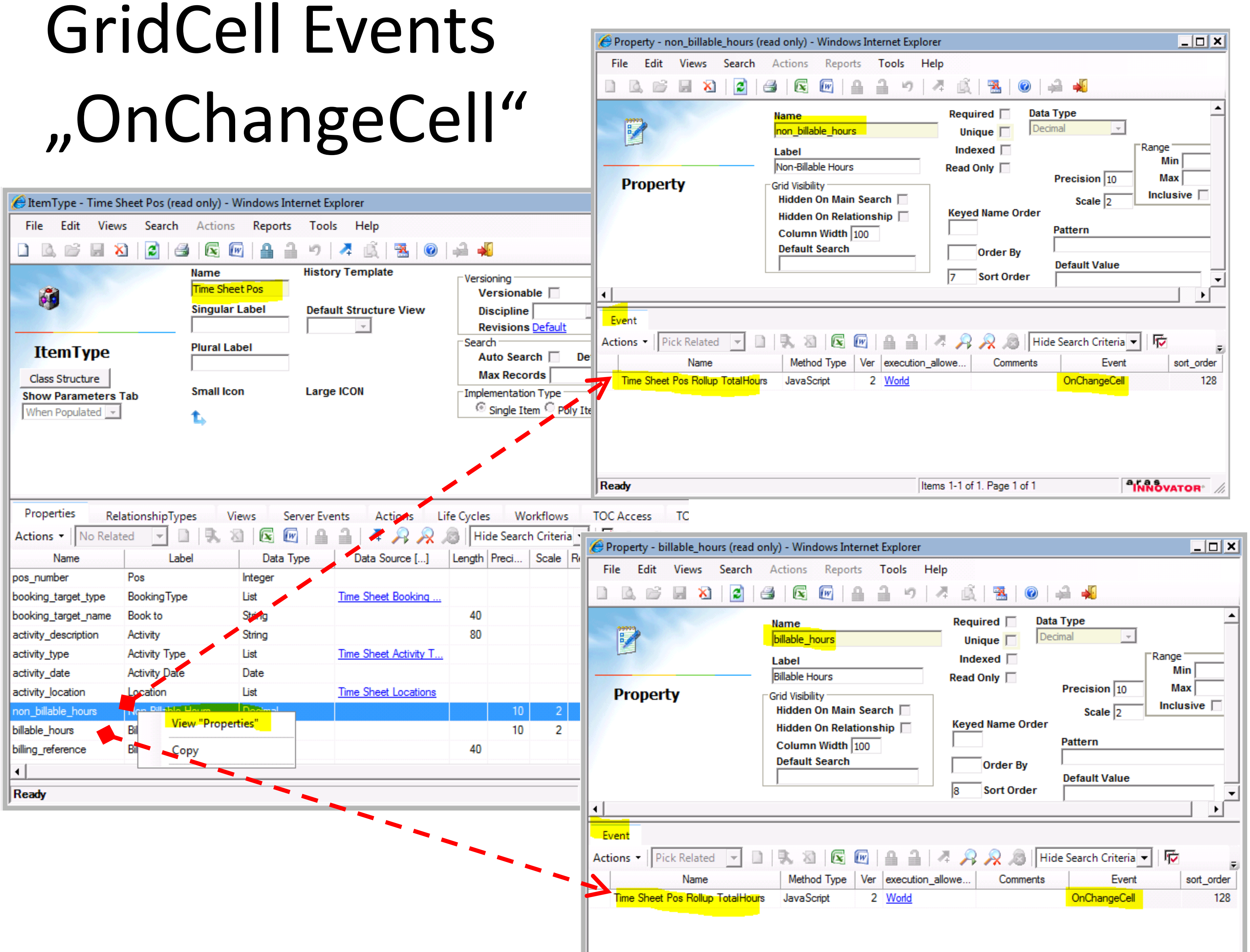


Update date field on top form from methods triggered by Grid Events

To cover new values in new or existing rows and to cover updates on cell values,
→ Use GridCell event „**onChangeCell**“ on column Properties to trigger calculation method

To cover deleting an entire row with ist values,
→ Use GridRow event „**onDeleteRow**“ on RelatinshipType definitions

GridCell Events „OnChangeCell“



ItemType - Time Sheet Pos (read only) - Windows Internet Explorer

ItemType

Class Structure

Name: Time Sheet Pos

History Template

Singular Label

Default Structure View

Plural Label

Versioning
Versionable □
Discipline
Revisions Default

Search
Auto Search □
Max Records

Show Parameters Tab: When Populated

Small Icon

Large ICON

Implementation Type
○ Single Item ○ Poly Ite

Name	Label	Data Type	Data Source [...]	Length	Preci...	Scale
pos_number	Pos	Integer				
booking_target_type	BookingType	List	Time Sheet Booking ...			
booking_target_name	Book to	String		40		
activity_description	Activity	String		80		
activity_type	Activity Type	List	Time Sheet Activity T...			
activity_date	Activity Date	Date				
activity_location	Location	List	Time Sheet Locations			
non_billable_hours	Non Billable Hours	Decimal			10	2
billable_hours	Bi				10	2
billing_reference	Bi			40		

Property - non_billable_hours (read only) - Windows Internet Explorer

Property

Name: non_billable_hours

Label: Non-Billable Hours

Required □
Unique □
Indexed □
Read Only □

Data Type: Decimal

Range
Min
Max
Inclusive □

Precision: 10

Scale: 2

Grid Visibility
Hidden On Main Search □
Hidden On Relationship □
Column Width: 100
Default Search

Keyed Name Order

Order By

Sort Order: 7

Pattern

Default Value

Event

Name	Method Type	Ver	execution_allowe...	Comments	Event	sort_order
Time Sheet Pos Rollup TotalHours	JavaScript	2	World		OnChangeCell	128

Ready — Items 1-1 of 1. Page 1 of 1

Property - billable_hours (read only) - Windows Internet Explorer

Property

Name: billable_hours

Label: Billable Hours

Required □
Unique □
Indexed □
Read Only □

Data Type: Decimal

Range
Min
Max
Inclusive □

Precision: 10

Scale: 2

Grid Visibility
Hidden On Main Search □
Hidden On Relationship □
Column Width: 100
Default Search

Keyed Name Order

Order By

Sort Order: 8

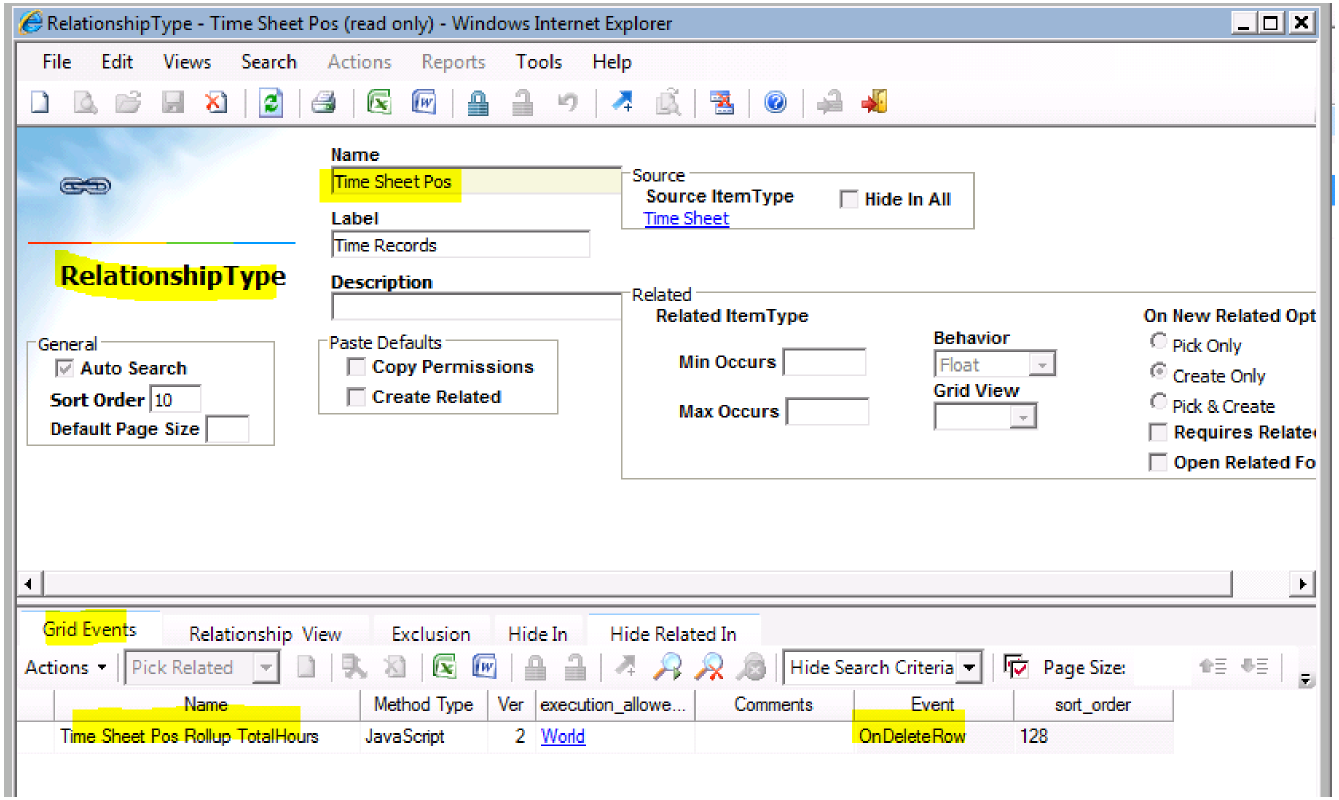
Pattern

Default Value

Event

Name	Method Type	Ver	execution_allowe...	Comments	Event	sort_order
Time Sheet Pos Rollup TotalHours	JavaScript	2	World		OnChangeCell	128

GridRow Event – OnDeleteRow



Name: Time Sheet Pos

Label: Time Records

Description:

Source

Source ItemType: Time Sheet

□ Hide In All

General

☑ Auto Search

Sort Order: 10

Default Page Size:

Paste Defaults

□ Copy Permissions

□ Create Related

Related

Related ItemType

Min Occurs:

Max Occurs:

Behavior: Float

Grid View:

On New Related Opt

○ Pick Only

● Create Only

○ Pick & Create

□ Requires Relate

□ Open Related Fo

Name	Method Type	Ver	execution_allowe...	Comments	Event	sort_order
Time Sheet Pos Rollup TotalHours	JavaScript	2	World		OnDeleteRow	128

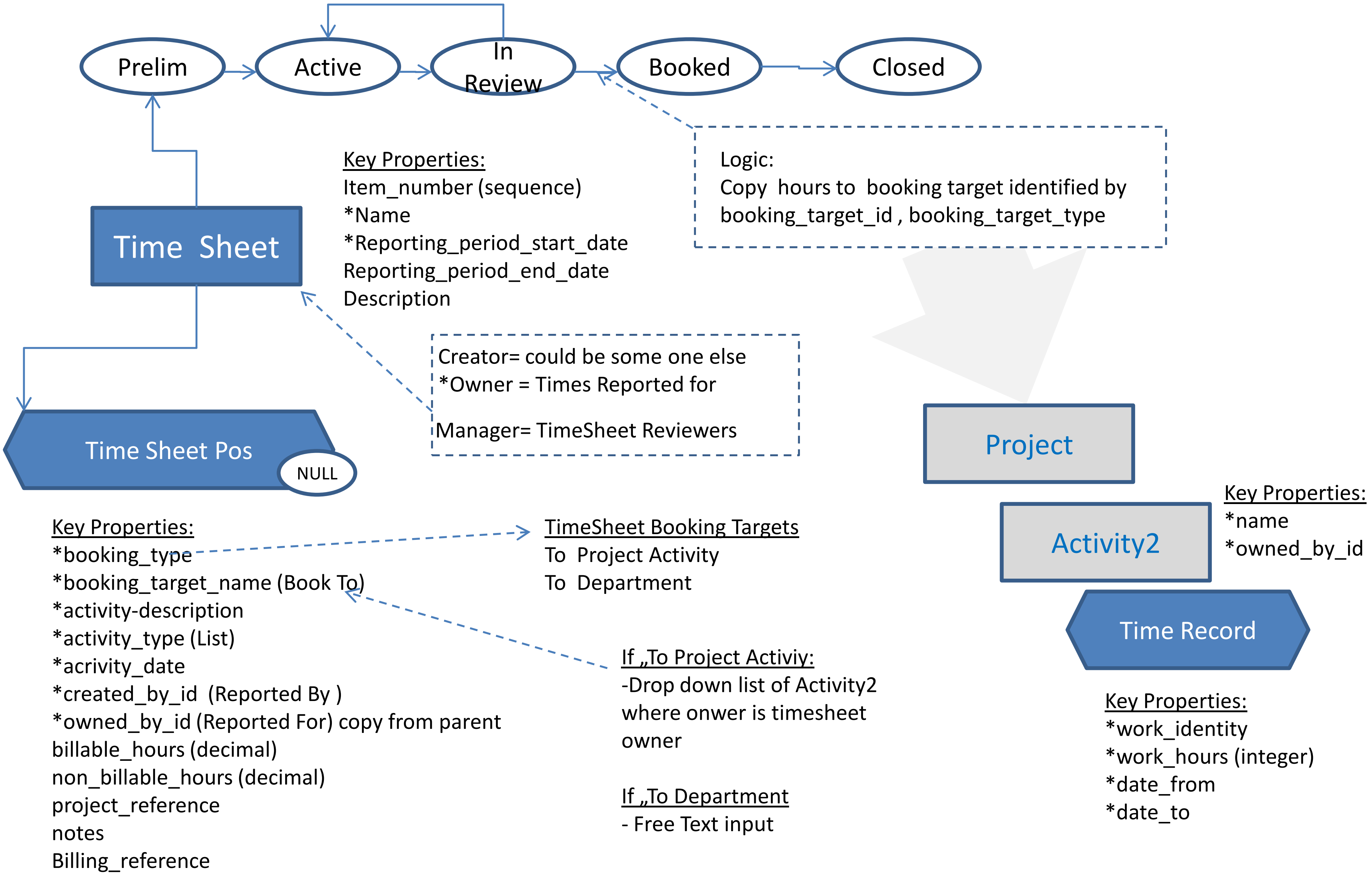
JS-Method (1): **Time Sheet Pos Rollup TotalHours**

```
//
// Grid Event: OnChangeCell, OnDeleteRow
// ItemType:  Time Sheet Pos  (relationship grid)
// Properties: billable_hours, non_billable_hours

// easiest approach is to clear and refresh the field every change

var inn = parent.thisItem.getInnovator();
var cntx = inn.getI18NSessionContext();
var gridRelationshipName = "Time Sheet Pos";

var billableHours = 0.0;
var nonBillableHours = 0.0;
var totalHours = 0.0;
var totalBillableHours = 0.0;
var totalNonBillableHours = 0.0;
var NumStr = "";
var thisRow;

// now query the DOM to get the values

// if standard grid argument "propertyName" is not set, then this method was called from onDeleteRow
if (typeof(propertyName) == "undefined"  || propertyName === "") {

  // get current totals
  NumStr = top.aras.getItemProperty(top.frames[1].document.item,"total_billable_hours","");
  if (NumStr===""){NumStr="0";}
  totalBillableHours = parseFloat(NumStr);
  NumStr = top.aras.getItemProperty(top.frames[1].document.item,"total_non_billable_hours","");
  if (NumStr===""){NumStr="0";}
  totalNonBillableHours = parseFloat(NumStr);

  thisRow = parent.thisItem.getItemsByXPath("Relationships/Item[@type='" + gridRelationshipName + "' and @id='" + relationshipID + "']");

  // get the values of deleted row (to be subtracted)
  NumStr = thisRow.getProperty("billable_hours","0.0");
  totalBillableHours -= parseFloat(cntx.ConvertFromNeutral(NumStr,"decimal",""));

  NumStr = thisRow.getProperty("non_billable_hours","0.0");
  totalNonBillableHours -= parseFloat(cntx.ConvertFromNeutral(NumStr,"decimal",""));
}
else
{
  // on Cell Change recalculate all totals
  totalBillableHours = 0.0;
  totalNonBillableHours = 0.0;
  //debugger;
  var relRows = parent.thisItem.getItemsByXPath("Relationships/Item[@type='" + gridRelationshipName + "']");
```

JS-Method (2): **Time Sheet Pos Rollup TotalHours**

```
 // loop through all positions and calculate toal of billable of non-billable hours
  for (var i=0; i< relRows.getItemCount(); i++) {
    thisRow= relRows.getItemByIndex(i);
    var thisRowID = thisRow.getID();

    // do not calculated values of rows marked as deleted
    if (thisRow.getAttribute("action","") !== "delete") {

      // calculate with new values or get old values
      NumStr = thisRow.getProperty("billable_hours","");
      if (NumStr===""){NumStr="0";}
      billableHours=parseFloat(cntx.ConvertFromNeutral(NumStr,"decimal",""));
      billableHours = billableHours.toFixed(2);

      NumStr = thisRow.getProperty("non_billable_hours","");
      if (NumStr===""){NumStr="0";}
      nonBillableHours=parseFloat(cntx.ConvertFromNeutral(NumStr,"decimal",""));
      nonBillableHours = nonBillableHours.toFixed(2);

      // build column sums
      totalBillableHours += parseFloat(billableHours);
      totalNonBillableHours += parseFloat(nonBillableHours);

      // build sum on selected row and display on grid
      if (relationshipID == thisRowID)
      {
                                        totalHours = parseFloat(billableHours) + parseFloat(nonBillableHours);
        NumStr = cntx.ConvertFromNeutral(totalHours.toFixed(2).toString(),"decimal","");
                                        // set num value in context
        thisRow.setProperty("total_hours",NumStr);
        // display num value immediately on grid in locale specific number format
                                        // #9 is column number on the grid (starting at 0)
        gridApplet.cells(thisRowID,9).SetValue(NumStr);
      }
    }
  }
}

// set new totals on top form
NumStr = cntx.ConvertFromNeutral(totalBillableHours.toFixed(2).toString(),"decimal","");
top.frames[1].handleItemChange("total_billable_hours",NumStr);
NumStr = cntx.ConvertFromNeutral(totalNonBillableHours.toFixed(2).toString(),"decimal","");
top.frames[1].handleItemChange("total_non_billable_hours",NumStr);

totalHours = totalBillableHours + totalNonBillableHours;

NumStr = cntx.ConvertFromNeutral(totalHours.toFixed(2).toString(),"decimal","");
top.frames[1].handleItemChange("total_hours",NumStr);
```

Data Model & Life Cycle



Key Properties:
Item_number (sequence)
*Name
*Reporting_period_start_date
Reporting_period_end_date
Description

Logic:
Copy hours to booking target identified by booking_target_id , booking_target_type

Creator= could be some one else
*Owner = Times Reported for

Manager= TimeSheet Reviewers

Key Properties:
*booking_type
*booking_target_name (Book To)
*activity-description
*activity_type (List)
*acrivity_date
*created_by_id (Reported By)
*owned_by_id (Reported For) copy from parent
billable_hours (decimal)
non_billable_hours (decimal)
project_reference
notes
Billing_reference

TimeSheet Booking Targets
To Project Activity
To Department

If „To Project Activiy:
-Drop down list of Activity2 where onwer is timesheet owner

If „To Department
- Free Text input

Key Properties:
*name
*owned_by_id

Key Properties:
*work_identity
*work_hours (integer)
*date_from
*date_to