

# CorpTech

## Functional Specification (Enhancement)

### H2R 2.2.1 E-1207 Entitlement Ratios

**Key Result Area:** Improvement  
**Program Name:** Shared Service Solutions  
**Sub-program:** HR  
**Phase:** Solution Design

**Version:** 1.0  
**Revision Date:** 16/05/2013  
**Issued by:** Roster Management

**Commercial in Confidence**

The material within this document has been developed solely for the internal business purposes of CorpTech on behalf of the Government of Queensland. The material shall be kept confidential and shall not be disclosed or made available directly or indirectly to other parties.

## Document Control Page

### Management

Version	Role	Name	Position	Signature	Date
1.0	Submitted by				
1.0	Accepted by				

### Revision History

Version	Date	Additions/Amendments	Author	Review By
0.1	15/09/06	Started Original Draft	Brad Swatogor	
0.2	18/09/06	Updates	Bill Grimson	Brad Swatogor
0.3	29/09/2006	Updates	Bill Grimson	Brad Swatogor
0.4	29/09/2006	Minor modifications	Brad Swatogor	
0.5	24/10/2006	Change Balance Sum Design to allow for more than 3 balances to be summed.	Brad Swatogor	Joseph Sullivan
0.6	01/03/07	Updated based on feedback from tech assessment	Brad Swatogor	Roslyn Ricoine, Brad Mammino, Darren Schooth
1.0	23/05/07	Updates from HRBS Review	S. Euston	

### Distribution List for Current Version

Name	Title
Jason Cameron	
Sara Simpson	
Indu Ranjan	
Jasen Floyd	
Nathan Hulse	

### Reference Documents

Document Title	Reference
HRBS Rostering Solution design Blueprint Balances and Entitlements Appendix	4.3 Long Service Leave (LSL) Balances 5.1 Entitlement Policies

# Table of Contents

<b>1</b>	<b>Introduction.....</b>	<b>5</b>
1.1	Background .....	5
1.2	Reference Information .....	5
1.3	Purpose and Scope .....	6
1.4	Justification.....	6
1.4.1	Recommended Approach.....	7
1.4.2	Alternatives .....	7
1.4.3	Risks .....	7
1.4.4	Assumptions .....	7
1.4.5	Issues Outstanding .....	7
1.5	Further Considerations .....	7
<b>2</b>	<b>Enhancement Specification – Ordinary Hours.....</b>	<b>8</b>
2.1	General Information.....	8
2.2	Security .....	8
2.3	Processing.....	8
2.3.1	Logic .....	8
2.3.2	Calculations .....	8
2.3.3	Updates.....	9
2.3.4	Error Handling.....	9
2.4	Comments .....	9
<b>3</b>	<b>Enhancement Specification – Standard Weekly Hours.....</b>	<b>10</b>
3.1	General Information.....	10
3.2	Security .....	11
3.3	Processing.....	11
3.3.1	Logic .....	11
3.3.2	Calculations .....	11
3.3.3	Updates.....	11
3.3.4	Error Handling.....	11
3.4	Comments .....	11
<b>4</b>	<b>Enhancement Specification – Balance Sum .....</b>	<b>12</b>
4.1	General Information.....	12
4.2	Security .....	12
4.3	Processing.....	12
4.3.1	Logic .....	12
4.3.2	Calculations .....	14
4.3.3	Updates.....	14
4.3.4	Error Handling.....	14
4.4	Comments .....	14
<b>5</b>	<b>Enhancement Specification – Cap at N Months.....</b>	<b>15</b>
5.1	General Information – Required for release 7 .....	15
5.2	Security .....	16
5.3	Processing.....	16
5.3.1	Logic .....	16
5.3.2	Calculations .....	18
5.3.3	Updates.....	19
5.3.4	Error Handling.....	19

5.4	Comments .....	19
-----	----------------	----

# 1 Introduction

## 1.1 Background

Balance accruals are performed within Workbrain using the Entitlements functionality. Many entitlements can be accomplished by pure configuration. However custom ratios are used in order to accrue a balance using complex business rules. Custom ratios are Java-based components that can be "plugged-in" to the entitlements engine in order to calculate the number of units to be accrued at a given point in time.

For more information on the manner in which entitlement ratios are used within the solution, refer to the *HRBS Rostering Blueprint Balances and Entitlements Appendix*.

The SSS Glossary contains definitions of many of the terms used in this document.

DBR/RICEF ID	Description	Fit or Gap ID
266B-WB	The system shall automatically cease/commence the accrual of leave according to changes in the employee's Terms of Employment.	Fit
1418B-WB	The system shall automate the payment and reduction of leave loading balances once per user-defined period according to the employee's Terms of Employment.	Fit
277B-WB	The system shall automatically cease/recommence the accrual of leave including loading when an employee starts/stop a leave exclusion absence.	Fit
640H-WB	The system shall generate payments, leave accruals and leave debits according to an employee's roster.	Fit
883B-WB	The system shall enable the user to configure date-effective leave accrual rules according to the employee's Terms of Employment.	Fit
884B-WB	The system shall automatically execute the user defined leave accrual rules via leave accrual processes.	Fit
888B-WB	The system shall automatically recognise a continuous period of user defined absences for the purpose of including but not limited to: * Leave accrual exclusions (eg unpaid leave). * Other Validations.	Gap
1207	Entitlement Ratios	

This development also incorporates the following child RICEF items:

Child RICEF ID	Description
N/A	

## 1.2 Reference Information

<b>Business Process Owner</b>	Nathan Hulse	
<b>Functional Owner</b>	Brad Swatogor	
<b>Application Environment</b>	Workbrain	
<b>Cross Reference</b>	<b>Framework Reference</b>	2.2 Manage Attendance and Leave



	Fit/Gap ID				n/a
	BPR Document(s)				n/a
	Detailed Process Design Report(s)				2.2.1 Maintain Leave and Entitlements
	Functional Specification(s)				H2R 2.2.1 E-1208 Entitlement Conditions H2R 2.2.1 e-1209 Entitlement Actions
	Configuration Document				TBA
	Testing document(s)				TBA
Solution Elements	Report	Extension	Input Form	Workflow	
	<input type="checkbox"/>	<input checked="" type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>	
	Conversion	Interface	Portal		
	<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>		

### 1.3 Purpose and Scope

This document describes the functional specifications for the following Workbrain entitlement ratios to be developed for Rostering Agencies.

1. **Ordinary Hours:** The ratio multiplies the entitlement value by the number of ordinary hours in the employee's timesheet on a given day. The entitlement value will be configured as an entitlement per ordinary hour. When that value is multiplied by this ratio, the full daily accrual amount is applied.
2. **Standard Weekly Hours:** The ratio multiplies the entitlement value by the employee's "standard weekly hours". The "standard hours" is the standard (or average) number of hours an employee works per week (as sent to Workbrain from SAP).
3. **Balance Sum:** The ratio sums up the values of the LSL1, LSL2, and LSL3 balances on the given day. The amount accrued to the LSL balance is equal to the difference between the current LSL balance and the sum of the other 3.
4. **Cap at N Months:** Applies the same logic as the "Hours Worked" ratio, but caps the entitlement at a configurable number of months of accrual. Once the employee has reached the specified number of months of balance, no more will accrue.

### 1.4 Justification

The justification for each ratio is described below:

- **Ordinary Hours:** The ratio is required in order to prorate entitlements based on the ordinary hours in the employees timesheet. It would not be possible to run daily accruals without this. In addition, this ratio allows part time employees to be assigned to the same entitlement policy as full time employees since the entitlements are automatically prorated based on ordinary hours.
- **Standard Weekly Hours:** The ratio is required in order to accommodate advance sick and recreation leave requirements for Resident Medical Officers and other employee groups that receive advanced entitlements on hire dates. Using this ratio allows standard hours to be converted into hourly leave entitlements.

- **Balance Sum:** This ratio is required in order to ensure that LSL accrued prior to 16 August 1993 or 15 August 1978 can be tracked separately but without displaying a separate balance to the employee. Without this ratio, 3 separate LSL balances would be displayed to employees.
- **Cap at N Months:** This ratio is required in order to cap the accrual of Professional Development Leave for Senior Medical Officers and Visiting Medical Officers within Queensland Health.

#### **1.4.1 Recommended Approach**

Entitlement ratios will be developed to satisfy the business requirements specified.

#### **1.4.2 Alternatives**

No other alternatives have been identified.

#### **1.4.3 Risks**

None identified.

#### **1.4.4 Assumptions**

The following assumptions have been made in the preparation of this document:

- A non-null value for standard hours will be imported from SAP for all employees who are advanced sick or recreation leave in their first year.
- Standard Hours will be imported from SAP as a fortnightly number.
- Any calculation group that has one or more VMOs will contain only VMOs.
- Any calculation group that has one or more SMOs will contain only SMOs.
- The definition of "ordinary hours" is the same within scheduled compliance rules as for entitlement rules.
- The definition of "ordinary hours" is the same across agencies and employee groups

#### **1.4.5 Issues Outstanding**

The following issues remain outstanding:

1. The final set of calculation groups has not yet been defined.
2. A decision is still pending on how prior service will be handled. If SAP is to handle all prior service, then the 3 LSL balances will not be required, meaning the "Balance Sum" (Section 4 of this document) will not be required.

### **1.5 Further Considerations**

## 2 Enhancement Specification – Ordinary Hours

### 2.1 General Information

Name	Ordinary Hours Entitlement Ratio
Application Environment	Workbrain
Application Modules	Leave Entitlements
Activation	

### 2.2 Security

Not applicable.

### 2.3 Processing

#### 2.3.1 Logic

The ratio calculates the number of “ordinary hours” in the employee’s timesheet on the day on which the entitlement is executed. Ordinary hours would be defined as a period of time where both the time code (TCODE\_FLAG2 = Y) and the hour type (HTYPE\_FLAG1 = Y) are flagged as ordinary hours.

These flags would exclude other periods of time from the timesheet such as overtime, TOIL and Flex time.

#### 2.3.2 Calculations

The calculation sums the number of minutes in the timesheet records (WORK\_DETAIL.WRKD\_MINUTES), including those records that meet all of the following criteria:

- The work detail employee is the same as the employee for which the entitlement is being applied
- The work detail date is the same as the date for which the entitlement is being applied
- The work detail time code is an “Ordinary Hours” time code (TCODE\_FLAG2 = Y)
- The work detail hour type is an “Ordinary Hours” hour type (HTYPE\_FLAG1 = Y)

The result of the ratio is the sum of WRKD\_MINUTES for all work details meeting the criteria above, divided by 60. The result will be rounded to 5 decimal places.

The following is an example of how this would work:

- Employee ( whose standard day is 7.25 hrs = **435 minutes**) works a total of 675 minutes on one timesheet day
- Workbrain time codes will split the total of 675 minutes into: 435 minutes ordinary time, 60 minutes flexible hours accrued and 180 minutes overtime/Toil. **The ordinary hour accrual of 435 minutes would be divided by 60 to convert to hours and fraction of hours (maximum 5 decimal spaces).** This will then equal the ratio – see example below:



Sample Timesheet					
Date	Start Time	End Time	Duration	Time Code	Hour Type
26/09/2006	9:00	12:00	3:00	WRK	REG
	12:00	12:45	0:45	BRK	UNPAID
	12:45	17:00	4:15	WRK	REG
	17:00	18:00	1:00	FLEX_A	UNPAID
	18:00	21:00	3:00	WRK	OT1.5

In the example above, the period from 9:00 – 12:00 and 12:45 – 17:00 would be considered “ordinary hours”. Unpaid breaks, flex time banked, and overtime would not be included in the ordinary hours calculation.

### 2.3.3 Updates

Not applicable.

### 2.3.4 Error Handling

None identified.

## 2.4 Comments

Currently TCODE\_FLAG2 is also being used within Schedule Compliance rules to represent Ordinary Hours. Although the definition of Ordinary Hours within the two components appears to be identical, it may be necessary to use different flags if design constraints necessitate the use of different time codes across the different components.

## 3 Enhancement Specification – Standard Weekly Hours

### 3.1 General Information

Name	Standard Weekly Hours Entitlement Ratio
Application Environment	Workbrain
Application Modules	Leave Entitlements
Activation	Entitlements Engine

Several categories of Employees in Workbrain agencies have an entitlement for their first year of accumulations for leave types such as sick and recreation to be advanced. These advances are conditional on their appointment being deemed permanent from their date of hire.

Workbrain will need to import the standard weekly hours for the employees concerned so that their leave advances can be calculated. A summary of the entitlement policies that include advanced entitlements is provided in the table below:

Policy Name	Policy Description	Entitlement Name	Entitlement Description
Sick Advance-4	Employees who earn 4 weeks of Recreational Leave and are advanced one year of sick leave on their hire date. <b>(permanent Hire only)</b>	1 Dec Leave Loading - 4 Week	Dec 1 Leave Loading - 4 Weeks Per Year
		LSL Accrual	Long Service Leave Accrual
		LSL Display	Updating the LSL Display balance based on the sum of LSL1, LSL2, and LSL3
		Rec - 4 Week	Recreation Leave - 4 Weeks Per Year
		Sick Advance	Advanced Sick Leave on Hire Date
		Sick Starting 2nd Year	Daily Sick Leave Accrual Starting in the Second Year of Employment
Sick Advance-5	Employees who earn 5 weeks of Recreational Leave and are advanced one year of sick leave on their hire date. <b>(permanent Hire only)</b>	1 Dec Leave Loading - 5 Week	Dec 1 Leave Loading - 5 Weeks Per Year
		LSL Accrual	Long Service Leave Accrual
		LSL Display	Updating the LSL Display balance based on the sum of LSL1, LSL2, and LSL3
		Rec - 5 Week	Recreation Leave - 5 Weeks Per Year
		Sick Advance	Advanced Sick Leave on Hire Date
		Sick Starting 2nd Year	Daily Sick Leave Accrual Starting in the Second Year of Employment
Sick Advance-6	Employees who earn 6 weeks of Recreational Leave and are advanced one year of sick leave on their hire date. <b>(permanent Hire only)</b>	1 Dec Leave Loading - 6 Week	Dec 1 Leave Loading - 6 Weeks Per Year
		LSL Accrual	Long Service Leave Accrual
		LSL Display	Updating the LSL Display balance based on the sum of LSL1, LSL2, and LSL3
		Rec - 6 Week	Recreation Leave - 6 Weeks Per Year
		Sick Advance	Advanced Sick Leave on Hire Date
		Sick Starting 2nd Year	Daily Sick Leave Accrual Starting in the Second Year of Employment
RMO-5	Remote Medical Officers who earn 5 weeks of Recreational Leave.	LSL Accrual	Long Service Leave Accrual
		LSL Display	Updating the LSL Display balance based on the sum of LSL1, LSL2, and LSL3
		PDL - SMO	Professional Development Leave - SMO
		Rec Advance - 5 Week	Advanced Recreation Leave on Hire Date - 5 Weeks
		Sick Advance	Advanced Sick Leave on Hire Date
		Sick Starting 2nd Year	Daily sklv Accrual Starting in the 2nd Year of Emp.

Policy Name	Policy Description	Entitlement Name	Entitlement Description
RMO-6	Remote Medical Officers who earn 6 weeks of Recreational Leave.	LSL Accrual	Long Service Leave Accrual
		LSL Display	Updating the LSL Display balance based on the sum of LSL1, LSL2, and LSL3
		PDL - SMO	Professional Development Leave - SMO
		Rec Advance - 6 Week	Advanced Recreation Leave on Hire Date - 5 Weeks
		Sick Advance	Advanced Sick Leave on Hire Date
		Sick Starting 2nd Year	Daily Sick Leave Accrual Starting in the Second Year of Employment

## 3.2 Security

Not applicable

## 3.3 Processing

### 3.3.1 Logic

The ratio returns the fortnightly standard hours value imported from SAP, stored in EMP\_VAL18 on the date which the entitlement is run. If a null value is encountered it is interpreted as a zero.

### 3.3.2 Calculations

The ratio returns the standard hours value imported from SAP, stored in EMP\_VAL18 on the date in which the entitlement is run. This value is divided by 2, to give standard weekly hours.

### 3.3.3 Updates

Not Applicable

### 3.3.4 Error Handling

None identified

## 3.4 Comments

Use the EMPLOYEE\_HISTORY table to retrieve effective dated employee data eg. Standard hours.



## 4 Enhancement Specification – Balance Sum

### 4.1 General Information

<b>Name</b>	Balance Sum Entitlement Ratio
<b>Application Environment</b>	Workbrain
<b>Application Modules</b>	Leave Entitlements
<b>Activation</b>	Entitlement Engine

The primary purpose of this ratio is to sum the LSL1, LSL2, and LSL3 balances, as well as the LSL1\_VMO, LSL2\_VMO, and LSL3\_VMO balances. In order to make the class more re-usable, it has been designed as a general balance sum class, calculating the sum of all balances in the specified Balance Sum Group.

### 4.2 Security

Not applicable

### 4.3 Processing

#### 4.3.1 Logic

The ratio performs the following steps:

1. Determine the Balance Sum Group being summed. The group is based on a value configured in the entitlement detail record (ENTITLEMENT\_DETAIL. ENTDET\_UDF1). If the value is null, generate an exception.
2. Find all balances in the same balance sum group as that identified in the previous step. A balance is in the same balance sum group if each of the following conditions are met:
  - a. The balance is associated with the calculation group of the employee for whom the entitlement is being run. This association is identified by the existence of a record referencing the balance and the calculation group in the IAG\_BALANCE\_CALCGRP table.
  - b. The value of the Balance Sum Group (IAG\_BALANCE\_CALCGRP. IBCG\_UDF2) is the same as that of the entitlement (ENTITLEMENT\_DETAIL. ENTDET\_UDF1).
3. Calculate the sum of all balances retrieved for the employee on the date for which the entitlement is being run (this should include any entitlements against those balances for that day).
4. Determine the difference between the value of the balance for which the entitlement is being run and the sum of balances retrieved in the previous step.

The ratio will apply the difference between the current value of the balance for which the entitlement is being run, and the sum of balances in the group. By this method, the balance will be incremented or decremented each day to ensure that it always equals the sum of the other balances.

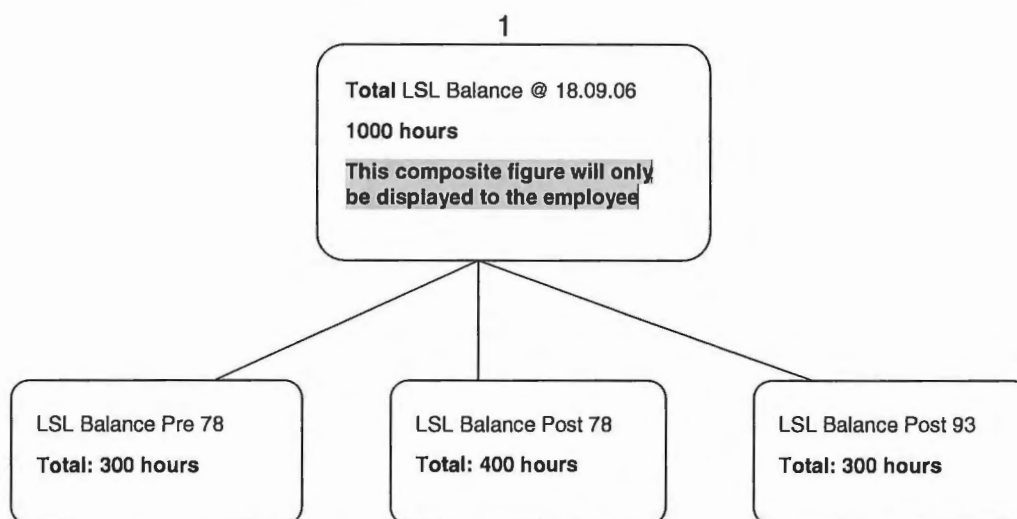


## Examples

The following two diagrams explain the layout of the employee LSL balance and how their LSL balance would both display and accrue over two consecutive working days eg. 18.09.06 and 19.09.06. This will reflect a daily accrual of **0.18125** hours

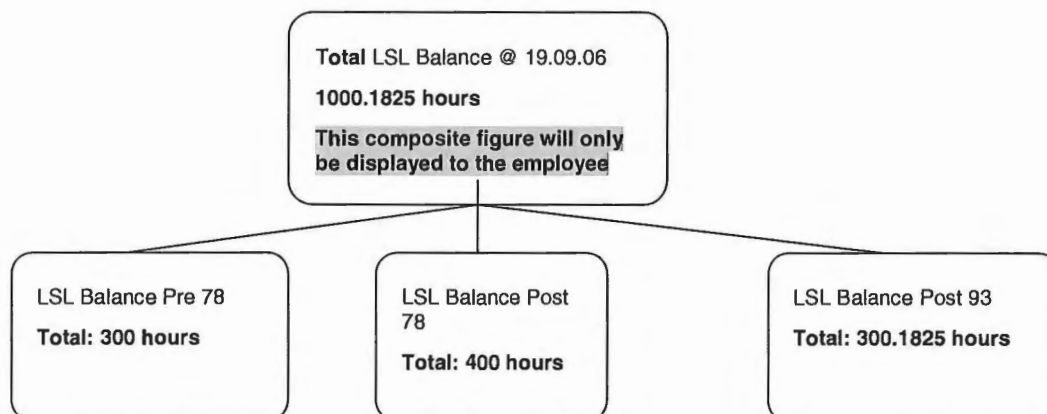
### Diagram 1:

How An Employee's Total LSL Balance of 1000 hours would initially be displayed on 18.09.06



### Diagram 2:

How an employees balance of 1000.1825 hours would be displayed on 19.09.06



The calculations above will also be applied to LSL balances in LSL1\_VMO, LSL2\_VMO and LSL3\_VMO

#### **4.3.2 Calculations**

See logic section.

#### **4.3.3 Updates**

Not applicable

#### **4.3.4 Error Handling**

##### **No Balance Sum Group Specified**

If a null value exists in the Balance Sum Group of the entitlement, generate an exception.

#### **4.4 Comments**

## 5 Enhancement Specification – Cap at N Months

### 5.1 General Information

Name	Cap At N Weeks Entitlement Ratio
Application Environment	Workbrain
Application Modules	Leave Entitlements
Activation	Entitlement Engine

There are three categories of Health Department employees who accrue Professional Development Leave (**PDL**). Each type of employee has a cap on the amount of PDL they may retain in their balance at any given time. The amount of the cap is not a fixed amount, but is based on the amount of PDL accrued over a fixed period of time. To use a simple example, an employee who accrues PDL at a rate of 1 hour per week may have a cap of 10 weeks' accrual. This means that the employee could have a maximum balance of 10 hours of PDL at any given time.

One additional complexity exists, the accrual policy is based on the number of hours worked, rather than just a flat weekly accrual. To extend the previous example, the employee works 40 hours per week and accrues 1 hour of PDL for every 40 hours worked. If the employee's hours were reduced to 20 hours per week, the 10 week cap would also be reduced. However, the cap is based on the number of hours worked over the past 10 weeks. If the employee had worked the previous 10 weeks at 20 hours per week, the cap would be 5 hours of PDL (half of 10). If in the past 10 weeks the employee worked half the time at 20 hours per week and half the time at 40 hours per week, the cap would be 7.5 as of the end of that period. In other words, the cap may change daily if the employee's working hours have varied in the previous ten week period.

The three categories of employees who receive PDL, and the applicable cap are as follows:

- VMO – 4 weeks per year with a cap of 16 weeks (**4 years**)
- SMO – 3.6 weeks per year with a cap of 36 weeks (**10 years**)
- RMO – 1 week per year with a cap of 2 weeks (**2 years**) - accrued only during service periods determined by pay rules **nb – leave for RMO employees can be taken in advance of normal accrual unlike VMO and SMO employees who can only draw down positive accruals**

Workbrain logic will need to ensure that the cap for each category of **PDL** employee is not reached until minimum periods of service of 4 years, 10 years and 2 years respectively are reached.

Workbrain Logic would also have to factor in periods of termination (eg for Locum Medical Staff). Staff in these circumstances would have their remaining PDL balance zeroed as at the date of termination. If re-employed after the date of termination they would start their accrual period afresh with a zero balance.

This rule will also apply to medical staff who change occupation (eg A VMO becoming a SMO).

## 5.2 Security

Not applicable.

## 5.3 Processing

### 5.3.1 Logic

The following is a summary of the overall logic performed by the ratio:

Retrieves the number of months of accrual at which to cap from (ENTITLEMENT\_DETAIL. ENTDET\_UDF4). **eg. 4 years = 48 months**

The "Ordinary Hours" ratio is then applied, and the cap subsequently applied. If the accrual would cause the balance to exceed the cap, the accrual is adjusted to ensure that the balance does not exceed the cap.

Workbrain Logic will have to accommodate the fact that the balance caps as per award conditions are expressed as weeks and not as specific balances in hours.

The logic will need to be able to determine what the capped balance in hours will be for individual employees after factoring in changes to working hours (eg fulltime to part-time hours and subsequent changes to part-time hours) and also periods of leave without pay with accruals being affected (same accrual rules that apply to sick/recreation leave).

Periods of time from where the Cap balance has been reached (and the balance not subsequently drawn down) will have a work summary (timesheet) flag set against them so that period of time is not considered for future accruals of PDL.

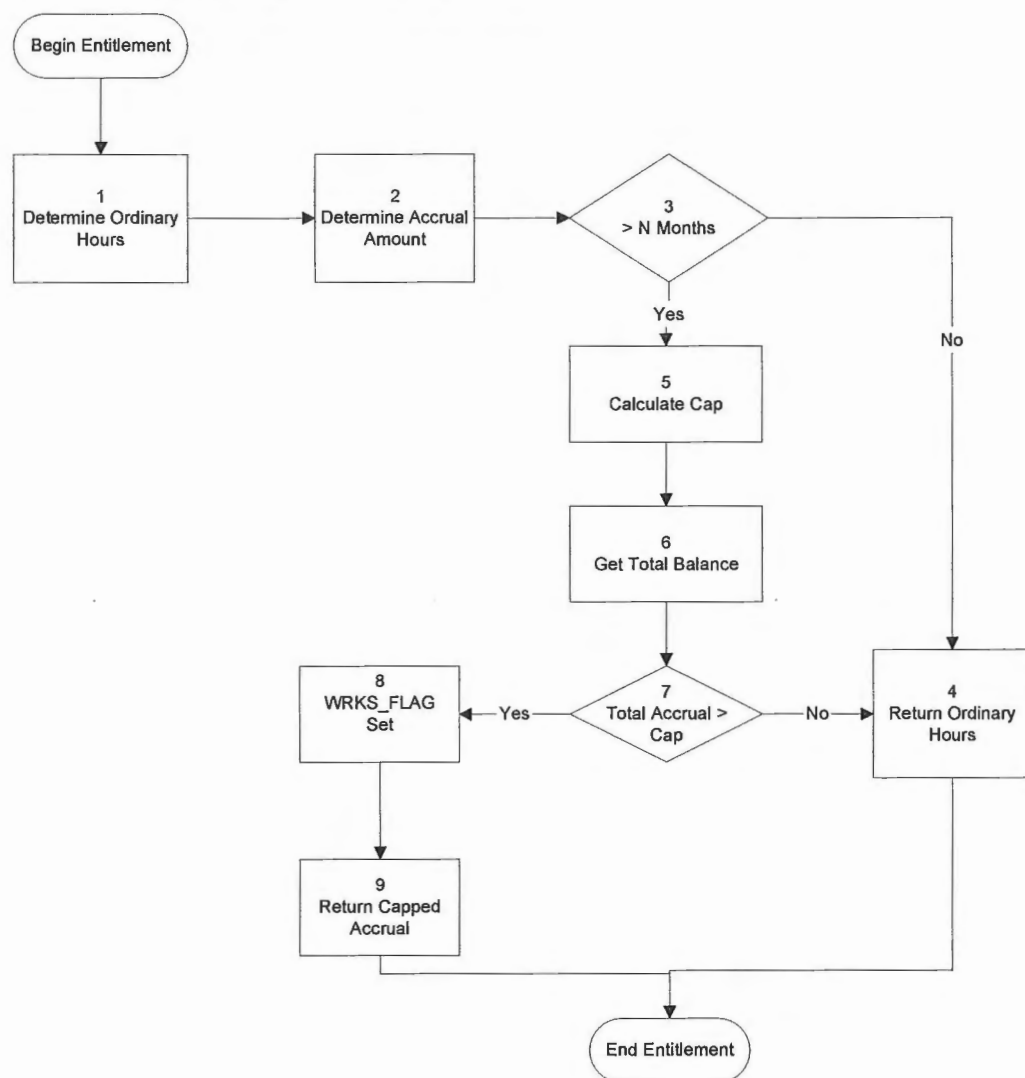
This flag will also be set against any periods of leave without pay greater than three months.

Workbrain would also not be able to look at hire dates to calculate the year accrual periods (eg 4 years for VMO) as a change of occupation invalidates prior PDL entitlement both for service and residual accruals. As such Workbrain logic would ignore hire and termination dates and calculate accrual years and balance increments based on the period of time

Workbrain Logic will determine that the entitlement cap will not be calculated until a minimum period of service has been reached (eg.- VMO – 4 years). At the end of this period an entitlement cap will be flagged (conditional on no leave having been drawn down) irrespective of what PDL balance has accrued (a variance in hours would be caused by a change in full-time hours to part-time hours).



The following diagram explains the Logic:



The Logic steps are as follows:

Step	Name	Type	Description
1	Determine Ordinary Hours	Process	Get the result of the ordinary hours ratio
2	Determine Accrual Amount	Process	Multiply ordinary hours by entitlement value (ENTDET_UNITS) and then determine accrual
3	> 3 N Months	Decision	Determine if employee has been in their current "accrual group" for more than the maximum accrual period (eg 48 months) – <b>See Calculations Section</b>
4	Return Ordinary Hours	Process	Where employee has been in entitlement group less than maximum accrual period then ordinary hours are

			returned
5	Calculate Cap	Process	Calculate the capped accrual balance – <b>See Calculations Section</b>
6	Get Total Balance	Process	Calculate total PDL balance including that days accrual
7	Total Accrual > Cap	Decision	Determine if the accrual amount is > than the cap amount calculated in step 5
8	WRKS_FLAG Set	Process	Flag set to prevent further accrual until balance is reduced by leave taking
9	Return Capped Accrual	Process	After WRKS_FLAG set, the capped accrual balance will be returned

### 5.3.2 Calculations

#### Has Employee Been In Accrual Group for N Months

The period of “N Months” begins as of the date an employee begins in a new “accrual group”. An accrual group is a group of entitlement policies that are treated as a single unit with respect to this entitlement cap. For example, an employee may change from one entitlement policy to another policy within the same accrual group. In that scenario, the count of N Months does not start over. However, if an employee changes policies to one in a different accrual group, the count of N months begins anew as of the new policy start date.

To determine if the employee has been in their current accrual group for N Months, do the following:

1. Get the accrual group associated with the employee’s current policy (ENTITLEMENT\_POLICY. ENTPOL\_UDF1) as of the date for which the entitlement is being run. If no value is assigned then the accrual group is null. This is a valid value.
2. Get the start date of the employee’s current assignment to that policy (ENT\_EMP\_POL. ENTEMPPOL\_START\_DATE). Set that as the start date of the current accrual group.
3. If the employee has an entitlement policy assignment that immediately precedes the current policy (there can be no gap between the policy assignments), and the accrual group is the same (note that null is a valid value) then get the start date of that policy. Set that as the start date of the current accrual group.
4. Repeat step 3 until the no previous policies are identified or until a policy in a different accrual group is reached. The start date retrieved from step 3 is the start date of the current accrual group.
5. Determine if the start date retrieved is greater than N Weeks prior to the date for which the entitlement is being run. This is done by comparing the difference between the entitlement date with the date retrieved from step 3 and the number of weeks configured in ENT\_DETAIL.ENTDET\_UDF4.

#### Accrual Cap

The calculation of the cap amount is as follows:

Retrieve the sum of all balance log entries meeting the following criteria:

1. The employee is the employee for which the entitlement is being run
2. The balance is the balance for which the entitlement is being run

3. The balance log date is between N Weeks prior to the day for which the entitlement is being run, and the day prior to the day for which the entitlement is being run. The value of N is taken from ENT\_DETAIL.ENTDET\_UDF4.
4. The accrual is a positive number (EBLOG\_DELTA > 0)

Effectively this defines N weeks of accrual as the exact amount accrued in the balance over the previous N week period.

### 5.3.3 Updates

WRKS\_FLAG

### 5.3.4 Error Handling

If a null value is found in ENT\_DETAIL.ENTDET\_UDF4 then raise the following exception:

“Cannot have a null value in ENTDET\_UDF4: N Weeks.”

## 5.4 Comments

Consideration will have to be given to **calculating the cap when employees are first loaded** onto Workbrain after the transition. This has been covered in the Leave Balances Conversion specification.