

August 2001 DRAFT
HRIS Follow-up Task Report

Return completed document to Judy.Thai@ucop.edu by August 30, 2001

ITEM#: 2	Revamp Action Code Process
----------	-----------------------------------

Team #: 2	Reported by: Robert Merryman (UCSD)
-----------	--

Definition Statement:

This relates primarily to the project comments, in the context of requirement **20.1**, and does not attempt to address the issues of history data element detail, level of summary, method of capture, etc, all of which are within the larger scope of Project # **1** - History Enhancement, although these two efforts should be tightly coupled.

- (1) Determine need to add new action codes or redefine use of current codes.
Possible re-engineering required.
- (2) Reexamine association of given actions with appropriate structures and levels of Employee Data, especially between Appointments and Distributions. Possibly reposition more actions to be stored in association with Distribution changes. In addition, entry of actions should be accommodate in ways and at levels which are most intuitively comfortable for the preparer, even if the action is recorded at another level or stored against multiple actual database records. In particular, where possible, the preparer should not have to repetitively enter a code representing the same action against multiple involved records.
- (3) Tighten action code editing to always require entry or derivation of non-default codes that are consistent with the data changes occurring within the scope of the current update. Issue error message if inconsistent code(s) entered and show list of codes consistent with changes if no entry made, and allow preparer to choose from the list. This should definitely be implemented in the context of Project # **3** - Appt/Dist Restructuring to achieve maximum benefit. In addition, this will naturally be impacted by the results of any re-engineering in item (1).
- (4) In order to avoid loss of action detail, determine how to support proper representation and entry/creation of multiple sequential and/or parallel actions, especially when these are executed retroactively. Some issues are specific to batch-generated actions, such as ranges and merits, while the more difficult issues involve supporting departmental preparers in entering such changes and not simply 'jumping' to the final status. Finding a way to ensure a high rate of compliance in accurate online entry of complex retroactive actions is likely to be extraordinarily difficult to achieve. This is greatly exacerbated by errors in the past, or even in the course of making the retroactive entry. Even if this is made easy to do, compliance is by no means guaranteed. One possible approach might be to integrate this process with the generation of retroactive pay adjustments, along with a sophisticated set of checks against the pay history.

Re-engineering Required? Y/N: **Yes (Minor-Medium)**

Size/Scope (person months)	Small < 1 month	Medium 1-3 months	Large 3-6 months	Extra Large > 6 months	Super Size > 1 Year
Requirements Definition & Development		(2)	(1) (3)	(4) * =>	(4) *
Programming		(1) (2)	(3)	(4) * =>	(4) *
Campus Implementation		(1) (2) (3)		(4) * =>	(4) *

Comments/Risk Factors:

- Parenthesized numbers in **Size/Scope** matrix refer to the Item groupings given in the Definition Statement section.
- Reengineering and training impact is relatively low on (3) since it would be implemented within the scope of the current workflow. Item (4) would require significant alteration to input screens and procedures.
- Regarding Item (4), Requirements, Programming, and Implementation could range (*) between Extra Large and Super Size, based on actual requirements. The lower estimate is for building new structures to support and facilitate layered reconstruction of retroaction changes and corrections, whereas the higher estimate involves a framework that would (in some unspecified way) attempt to coerce proper entry.
- Risks involved with Items (3) and (4) include political backlash from campus departments due to real or perceived increase in workload without commensurate payback at their level – especially true of (4).