

# Prearchive Statuses and Permissions

The Prearchive is constantly listening for incoming sessions, and is very sensitive to the status of a given session. As such, it makes sense to restrict what actions a user can perform on a given session (or group of sessions) based on its status.

Here is a list of statuses and actions, cross-referenced for easy reference. It should be noted that bulk actions are processed asynchronously, so there is an fine distinction between "Delete Pending" and "Deleting Now,"

Action / Status	Archive	Review and Archive*	Change Project	Delete	Rebuild	Cancel****
RECEIVING	No	No	No	Yes	Yes	No
READY	Yes	Yes	Yes	Yes	Yes	No
CONFLICT **	No	Yes	No	Yes	Yes	No
ERROR	No	No	No	Yes	Yes	No
(UNASSIGNED) ***	No	No	Yes (by Admin only)	Yes (by Admin only)	Yes (by Admin only)	No
<i>"Pending" Statuses - Action has been selected and is queued for processing</i>						
ARCHIVE PENDING	No	No	No ?	No ?	No ?	Yes
BUILD PENDING	No	No	No ?	No ?	No ?	Yes
DELETE PENDING	No	No	No ?	No ?	No ?	Yes
MOVE PENDING	No	No	No	No ?	No ?	Yes
<i>"Locked" Statuses - Action is currently in process</i>						
ARCHIVING NOW	No	No	No	No ?	No ?	No
BUILDING NOW	No	No	No	No ?	No ?	No
DELETING NOW	No	No	No	No ?	No ?	No
MOVING NOW	No	No	No	No ?	No ?	No

## \* Review and Archive

\*\* The conflict status reflects a session in the Prearchive that has the same name and project as an existing session in the archive.

\*\*\* Unassigned sessions do not appear in the UI for non-Admin users. These sessions have malformed headers caused by improper data entry at the scanner.

\*\*\*\* There is REST support for an "Undo" or "Cancel" action, but no reflection of this in the UI. This may not appear until XNAT 2.0.

Items marked with ? indicates an open question for XNAT developers.

- Items in a Pending state should not have any other actions allowed, other than an "Undo" function. However, there is not support for this in the UI yet.
- Items in a Locked state should not have any actions allowed, period.

## Actions Logic Flow

```
MOVABLE : function (session) {
return !prearchiveui.Utils.isLocked(session)
&& session.status != "MOVING" && session.status != "RECEIVING"
&& session.status != "ERROR";
},
ARCHIVABLE : function (session)
{
return !prearchiveui.Utils.isLocked(session) && (session.status ==
"READY" || session.status == "CONFLICT") && session.project !=
prearchiveui.Utils.UNASSIGNED;
},
DELETABLE : function (session)
{
return true;
},
RESETABLE : function (session) {
return true;
}
```

## Statuses Logic Flow

```
if(internalStatus=="BUILDING" || internalStatus=="QUEUED" ||
internalStatus=="_QUEUED" || internalStatus=="QUEUED_BUILDING" ){
return
"Build pending";
}
else if (internalStatus=="_BUILDING" ){
return
"Building now";
}
else if (internalStatus=="ARCHIVING" || status ==
"QUEUED_ARCHIVING"){
return "Archive pending";
}
else if
(internalStatus=="_ARCHIVING" ){
return "Archiving now";
}
else if
(internalStatus=="MOVING" ){
return "Move pending";
}
else if
(internalStatus=="_MOVING" ){
return "Moving now";
}
else if
(internalStatus=="DELETING" ){
return "Delete pending";
}
else if
(internalStatus=="_DELETING" ){
return "Deleting now";
}
else if
(internalStatus=="RECEIVING" || internalStatus=="_RECEIVING"){
return
"Receiving";
}
else if (internalStatus=="READY" ){
return
"Ready";
}
else if (internalStatus=="CONFLICT" ||
internalStatus=="_CONFLICT"){
return "Conflict";
}
else if
(internalStatus=="ERROR"){
return "Error";
}
else{
return "Unknown
Status";
}
```