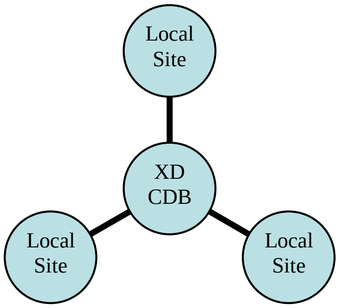
# AMIE in XSEDE

## Overview

This document describes the sets of AMIE transactions used by XSEDE. It is assumed that the reader has read the document *The AMIE Model* which describes the AMIE transaction model. The overview from that document says:

> The AMIE model consists of two sites and an agreed upon set of **transactions** that the two sites will use to send account management data to each other. A transaction consists of **packets** of data sent between the two sites. The site sending a packet is called the **local site** and the packet is known to the sending site as an **outgoing packet**. The receiving site is called the **remote site** and the packet received is known to the receiving site as an **incoming packet**. The site that creates the transaction (and sends the first packet) is also called the **originating site**.

From the AMIE perspective, XSEDE consists of a number of local sites that exchange account management information with the XSEDE Central Database (XDCDB).



This document is written from the perspective of a local site. The remote site is thus the XSEDE Central Database (XDCDB). Incoming packets are packets sent by XDCDB to the local site. Outgoing packets are packets sent to the XDCDB by the local site.

It is also assumed that the local sites will use the AMIE reference implementation described in the document *Implementing AMIE*.

## Transaction Packet Sequences

As described in the *Amie Model*, transactions consist of packets.

Each packet not only has data associated with it, but also, in most cases, an expected reply. The expected reply mechanism is used by AMIE implementations to specify **packet sequences** within transactions. For example, a *request_project_create* packet could specify that it expects a *notify_project_create* packet in reply, which could specify that it expects a *data_project_create* packet, which could specify that it expects an *inform_transaction_complete* packet in reply. The *inform_transaction_complete* packet is the only packet which can not have an expected reply. It would complete the sequence (and the transaction).

## Packet Data

In the descriptions of the packet sequences for transactions, the content of each is described in a table that corresponds to the *data_tbl* in the reference implementation. For example, packet data might look like:

| tag | subtag | seq | value |
|---|---|---|---|
| **ProjectID** | | | afm |
| ResourceList | | 0 | abc.nist.xsede[1] |
| Sfos | **Number** | 0 | 120 |
| Sfos | **Number** | 1 | 124 |
| **StartDate** | | | 2003-01-05 |

Required *tags* and *subtags* are shown in bold. An empty entry in the *subtag* column means that the *subtag* is a database NULL value. An entry in the *seq* column indicates that the *tag* represents a list which may contain many records (more or less than are shown in the examples). If there is no entry for *seq*, then *seq* is zero (0), but the *tag* does not represent a list.

In the above example, the *ProjectID* and *StartDate* are required. The *ResourceList* and *Sfos* are not required, but if the *Sfos* is present, the *Number* subtag is required. Also, the *ResourceList* and *Sfos* are lists.

### *XML Data Issues*

***Date***: Since AMIE uses XML as the representational form during information exchange, all date values must be formatted as valid XML dates. This format is *yyyy-mm-dd*, where *yyyy* is the 4-digit year, *mm* is the month (01 = jan, 02 = feb …, 12 = dec), and *dd* is the day (01, 02, …, 31). For example, Jan 5, 2003 would be 2003-01-05.

***Datetime***: XML datetime format is one of

*yyyy-mm-dd***T***hh***:***mm***:***ss***Z**
*yyyy-mm-ss***T***hh***:***mm***:***ss-hh***:***mm*
*yyyy-mm-ss***T***hh***:***mm***:***ss+hh***:***mm*

The *yyyy-mm-dd* before the **T** is the date part (same rules as Dates above). The *hh***:***mm***:***ss* is the time in hours (0-23), minutes (0-59), and seconds (0-59). The part following the time is a required timezone. Z indicates GMT. It can also be specified as an offset from GMT: -*hh***:***mm* (hours and minutes), or +*hh***:***mm*.

***Duration***: XML duration format is **PT***h***H***m***M***s***S**, where *h* is the number of hours, *m* is the number of minutes, and *s* is the number of seconds.

[1] The resource and sire names in this document may not be actual resource or site names. They are for illustrative purposes only. XSEDE sites and resource names will be defined in the XDCDB.

***Boolean***: XML Boolean values should be coded as 1 (true) or 0 (false).

***Spaces***: Whitespace in an XML file is considered part of the data. In particular, leading and trailing whitespace is not removed by AMIE or by the XDCDB. Any whitespace that is not a critical part of the data should be removed before it is inserted into the *data_tbl*. Furthermore, if, as a result of removing whitespace the result is an empty string, and the tag is optional, then data should not be inserted.

***Valid XML characters***: Only the tab, linefeed, and return characters (octal 011, 012, 015), and characters in the octal range 040-177 should be used. Other characters will be rejected by the XML parser.

## DN Handling

In order to insure that DNs are properly propagated within XSEDE, there are some requirements about how to handle DNs in the AMIE packets.

1. When constructing the *notify_project_create* and *notify_account_create* packets, sites must include **all** DNs that they have in their grid mapfiles for the user specified in the packet.
2. When processing *request_project_create*, *request_account_create*, *data_project_create,* and *data_account_create* packets, sites must add to their grid mapfiles **all** the DNs listed for the user in the packet.
3. There are two types of *request_user_modify* packet. For the *request_user_modify (replace)* packet, sites must add to their grid mapfiles **all** the DNs listed for the user in the packet. For the *request_user_modify (delete)* packet, sites must remove from their grid mapfiles the DNs listed for the user in the packet.

NOTE: in the documentation for these packets, the *DnList* tag is specified as optional. However, this should **not** be interpreted as meaning that providing the DNs is optional. The above rules must be adhered to. The tag is optional to support the (rare) case when a user does not have any DNs in the grid mapfiles at a site.

## Inform_transaction_complete

The final packet in a transaction is an *inform_transaction_complete* packet. This packet has two states: success and failure. A failure may be sent in response to any packet. A success may only be sent in reply to a packet that expects an *inform_transaction_complete* reply.

***inform_transaction_complete*** packet (success)

| tag | subtag | seq | value |
|---|---|---|---|
| **DetailCode** | | | 1 |
| **Message** | | | … success message |
| **StatusCode** | | | Success |

***inform_transaction_complete*** packet (failure)

| tag | subtag | seq | value |
|---|---|---|---|
| **DetailCode** | | | 2 |
| **Message** | | | … error message |
| **StatusCode** | | | Failure |

The *StatusCode* determines if the *inform_transaction_complete* is a success or failure (and its value must be one of "Success" or "Failure"). The *Message* is for human consumption and is mostly useful in the failure case. The *DetailCode* is not used (but must be a positive integer).

# Transactions

Transactions have two flavors – transactions that originate at the XDCDB and those that originate at the local site.

## *XDCDB Transactions*

Transactions that originate at the XDCDB are requests made upon the local sites to create projects and accounts[2], or requests to modify information about a project, account, or a user. Often this will mean that an event occurred at one site that the XDCDB then asks the other sites to also perform. For example, one site creates a project and notifies the XDCDB about the project (using a transaction that originates at the local site). The XDCDB then would send requests to all of the other sites to create that project at their site (using transactions that originate at the XDCDB).

For these types of transactions, local sites do not have to create a transaction. AMIE does this for them. Local sites need to process incoming packets and create the appropriate outgoing packets in reply to the incoming packets (including the packet data and the expected reply).

### ***request_project_create*** transaction

The purpose of this transaction is to ask a local site to create a project and an account for the PI of the project. The XDCDB sends the project and account information in a *request_project_create* packet. The local site creates the project and account and sends the site-specific project and account information back to the XDCDB in a *notify_project_create* packet. The XDCDB then sends additional information about the project to the local site in a *data_project_create* packet. The local site finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| request_project_create | incoming |
| notify_project_create | outgoing |
| data_project_create | incoming |
| inform_transaction_complete | outgoing |

[2] Account here means an association of a user with a project. It does not mean a login (ie, a Unix account). Users can have multiple accounts – ie, can be on multiple projects.

### ***request_project_create***

The *request_project_create* packet contains information to be used by a local site to create a project and an account for the PI on the project. The result of this request should be the creation of a local site project (with a local project ID) as well as an account for the PI (with a local person ID).

NOTE: a *request_project_create* implies project reactivation. If the project had been inactivated (by a *request_project_inactivate*), the site must reactivate the project as well as performing the other actions required by this transaction. The only account that should be reactivated would be the account for the PI in the *request_project_create*.

NOTE**:** The *PiGlobalID* is the internal ID for the PI within the XDCDB. It could be used to uniquely identify the PI.

*request_project_create*

| tag | subtag | seq | value |
|---|---|---|---|
| Abstract | | | Some text …. |

| | | | |
|---|---|---|---|
| AcademicDegree | **Degree**[3] | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| AllocatedResource | | | ecss.xsede |
| **AllocationType**[4] | | | new |
| ChargeNumber | | | TG-AST040002 |
| **EndDate** | | | 2013-12-31 |
| GrantType | | | Science Gateway |
| **GrantNumber** | | | AST040002 |
| NsfStatusCode[5] | | | CN |
| **PfosNumber** | | | 120 |
| PiBusinessPhoneComment | | | Business phone on campus |
| PiBusinessPhoneExtension | | | 7570 |

[3] Degrees are:

BA=Bachelor of Arts
BFA=Bachelor of Fine Arts
BS=Bachelor of Science
DO=Doctor of Osteopathic Medicine
JD=Juris Doctor
MA=Master of Arts
MBA=Master of Business Administration
MD=Doctor of Medicine
MFA=Master of Fine Arts
MS=Master of Science
PhD=Doctor of Philosophy
ScD=Doctor of Science

[4] *AllocationType* must be one of **new, renewal, extension, supplement, transfer, advance** or **adjustment**. See discussion under the *notify_project_create* transaction.

[5] *NsfStatusCodes* are:

CN=Center Non-Researcher Staff
CR=Center Researcher Staff
F=Faculty
G=Government User
GS=Graduate Student
HS=High School Studen
HT=High School Teache
I=Industrial User
N=Unaffiliated User
NP=Nonprofit User
O=Other User
PD=Postdoctorate
UG=Undergraduate Student
UK=Unknown
UN=University Non-Research Staff
UR=University Research Staff (excluding postdoctorates)

| | | | |
|---|---|---|---|
| PiBusinessPhoneNumber[6] | | | (217) 244-0072 |
| PiCitizenship | | | 9US |
| PiCity | | | Champaign |
| PiCountry | | | 9US |
| PiDepartment | | | N.C.S.A |
| PiDnList | | 0 | /C=US/O=NCSA/CN=Steven James Quinn |
| PiDnList | | 1 | /C=US/O=PSC/CN=Steven James Quinn |
| PiDnList | | 2 | /C=US/O=SDSC/CN=Steven James Quinn |
| PiEmail | | | squinn@ncsa.uiuc.edu |
| PiFax | | | (217)-244-1987 |
| **PiFirstName** | | | Steven |
| PiGlobalID | | | 70 |
| PiHomePhoneComment | | | Has Caller ID |
| PiHomePhoneExtension | | | none |
| PiHomePhoneNumber[7] | | | (217) 222-0000 |
| **PiLastName** | | | Quinn |
| PiMiddleName | | | James |
| **PiOrganization** | | | University of Illinois at Urbana-Champaign |
| **PiOrgCode** | | | 0017756 |
| PiPersonID[8] | | | 6751 |
| PiRequestedLoginList | | 0 | squinn |
| PiState | | | IL |
| PiStreetAddress | | | 131 CAB |
| PiStreetAddress2 | | | 605 E. Springfield Ave. |
| PiTitle | | | n/a |
| PiZip | | | 61820 |
| ProjectID[9] | | | aaa |
| ProjectTitle | | | Planetary Motion |
| **ResourceList** | | 0 | hc.ncsa.xsede |
| RoleList | | 0 | allocation_manager |
| **ServiceUnitsAllocated** | | | 99999 |
| Sfos[10] | **Number** | 0 | 122 |
| Sfos | **Number** | 1 | 124 |
| SitePersonId | **Site** | 0 | ANL |
| SitePersonId | **PersonID** | 0 | tg-squin |
| SitePersonId | **Site** | 1 | SDSC |
| SitePersonId | **PersonID** | 1 | 17761 |

[6] *PiBusinessPhoneNumber* is required if the *PiBusinessPhoneComment* or *PiBusinessPhoneExtension* is present.

[7] *PiHomePhoneNumber* is required if the *PiHomePhoneComment* or *PiHomePhoneExtension* is present.

[8] If the TGCDB already has the local site person ID for the PI, then the *PiPersonID* tag will be present. Otherwise it will not be present.

[9] The *ProjectID* will be the local project ID, if known.

[10] Due to implementation issues, the TGCDB may send *Sfos Number* with value of zero (0) which is not a valid number. The TGCDB may do this to indicate that there are no secondary fields of science associated with the project. Local sites must ignore *Sfos* with *Number* = 0.

| | | | |
|---|---|---|---|
| **StartDate** | | | 2003-12-16 |

***notify_project_create***

After having created a project and an account, the local site must reply with a *notify_project_create* packet. This packet must contain the *GrantNumber* from the *request_project_create*, the local project ID (*ProjectID*), the local ID for the PI (*PiPersonID*) which should be a locally unique identifier for the PI at the local site, and the login at the local site for the PI (*PiRemoteSiteLogin*). The *ResourceList* will only have one resource (and it will be a resource at the remote site).

The table below lists the tags that could be included in this packet (the ones in **bold *must*** be included). However, in a reply to a *request_project_create*, all the project-related tags are ignored except *GrantNumber, ProjectID, PfosNumber,* and *ResourceList*[11]. Also, if the PI is already known to the XDCDB (based on the *PiPersonID* or the *PiGlobalID*), then all the *Pi* tags are ignored, except *PiGlobalID, PiPersonID,* and *PiRemoteSiteLogin*. Otherwise the PI information is recorded at the XDCDB for the *PiPersonID.*

The *PiGlobalID* is the internal ID for the PI within the XDCDB.

The *NsfStatusCode* must be a valid NSF Status Code (as defined in the XDCDB). The *PiOrgCode* must be a valid NSF organization code (as defined in the XDCDB). The *PiCountry* and *PiCitizenship* must be valid NSF country codes (as defined in the XDCDB). The *PiState* must be a valid state abbreviation (as defined in the XDCDB).

The *AllocationType* must be one of **new, renewal, extension, supplement, transfer, advance** or **adjustment**.

- A **new** type creates a new project and creates a new allocation on a specific resource. New allocations for several resources can be created at the same time for the same new project. (Note that POPS may send the XDCDB an allocation for a specific resource, in which case the appropriate site will get an RPC for that resource. POPS may also send an allocation for a grid, in which case the affected sites will get RPCs for each site resource that is part of the grid.)
- A **renewal** is a continuing award for an existing project. In a sense, it "replaces" the previous year's allocation (as of the start date of the renewal). A renewal may be sent for a machine that did not previously have an allocation within the project, since a PI can renew onto different machines each year. In this case, the project may be unknown to the RP. A renewal for a resource not previously within the project should be handled by the RP as a new type, using the provided start/end dates.
- A **supplement** adds SUs to an existing allocation (without changing the start or end dates). A supplement can also be received for a project unknown to the RP or a machine not previously on the project, in which case it should be handled as a new type, using the provided SUs and start/end dates.

[11] *ResourceList* must be the same as the *ResourceList* in the *request_project_create*.

- A **transfer** type occurs as one of a pair of transactions – one negative, and one positive. A negative transfer for an unknown allocation is an error. Otherwise, the amount is deducted from the allocation. A positive transfer should be treated the same as a supplement.
- An **extension** extends the end date of an allocation. An extension should only be received for an existing allocation.
- An **advance** is an allocation given in anticipation of a new or renewal allocation. It should be treated as a supplement.
- An **adjustment** modifies an existing allocation for other reasons. It should be handled as a transfer type; but unlike a transfer, there is only one transaction. Currently an adjustment transaction is used to deduct usage charged against an advance from a renewal allocation.

***notify_project_create***

| tag | subtag | seq | value |
|---|---|---|---|
| Abstract | | | Some text …. |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| AllocationType[12] | | | |
| EndDate | | | 2013-12-31 |
| **GrantNumber** | | | AST040002 |
| NsfStatusCode | | | CN |
| **PfosNumber** | | | 120 |
| PiBusinessPhoneComment | | | Business phone on campus |
| PiBusinessPhoneExtension | | | 7570 |
| PiBusinessPhoneNumber[13] | | | (217) 244-0072 |
| PiCitizenship | | | 9US |
| PiCity | | | Champaign |
| PiCountry | | | 9US |
| PiDepartment | | | N.C.S.A |
| PiDnList | | 0 | /C=US/O=NCSA/CN=Steven James Quinn |
| PiDnList | | 1 | /C=US/O=PSC/CN=Steven James Quinn |
| PiDnList | | 2 | /C=US/O=SDSC/CN=Steven James Quinn |
| PiEmail | | | squinn@ncsa.uiuc.edu |
| PiFax | | | (217)-244-1987 |
| PiFirstName | | | Steven |
| PiGlobalID | | | 70 |
| PiHomePhoneComment | | | Has caller ID |
| PiHomePhoneExtension | | | none |
| PiHomePhoneNumber[14] | | | (217) 222-0000 |
| PiLastName | | | Quinn |

[12] If the NPC is not in reply to an RPC, then the *AllocationType* is required. Otherwise it is ignored.

[13] *PiBusinessPhoneNumber* is required if the *PiBusinessPhoneComment* or *PiBusinessPhoneExtension* is present.

[14] *PiHomePhoneNumber* is required if the *PiHomePhoneComment* or *PiHomePhoneExtension* is present.

| | | | |
|---|---|---|---|
| PiMiddleName | | | James |
| PiOrganization | | | University of Illinois at Urbana-Champaign |
| **PiOrgCode** | | | 0017756 |
| **PiPersonID** | | | 6751 |
| PiPosition | | | Research Programmer |
| **PiRemoteSiteLogin** | | | squinn |
| PiRequestedLoginList | | 0 | squinn |
| PiState | | | IL |
| PiStreetAddress | | | 131 CAB |
| PiStreetAddress2 | | | 605 E. Springfield Ave. |
| PiTitle | | | n/a |
| PiUID | | | 12456 |
| PiZip | | | 61820 |
| ProjectGID | | | 123 |
| **ProjectID** | | | afm |
| **ProjectTitle** | | | Planetary Motion |
| **ResourceList** | | 0 | hc.ncsa.xsede |
| ResourceLogin | **Resource** | 0 | portal.teragrid |
| ResourceLogin | **Login** | 0 | squinn |
| RoleList | | 0 | allocation_manager |
| **ServiceUnitsAllocated** | | | 99999 |
| Sfos | **Number** | 0 | 122 |
| Sfos | **Number** | 1 | 124 |
| **StartDate** | | | 2003-12-16 |

***data_project_create***

The XDCDB will reply with a *data_project_create* packet that contains all the DNs known by the XDCDB for the project PI.

***data_project_create***

| tag | subtag | seq | value |
|---|---|---|---|
| DnList | | 0 | /C=US/O=NCSA/CN=Steven James Quinn |
| DnList | | 1 | /C=US/O=PSC/CN=Steven James Quinn |
| DnList | | 2 | /C=US/O=SDSC/CN=Steven James Quinn |
| **PersonID** | | | 6751 |
| **ProjectID** | | | afm |

The local site must then reply with an *inform_transaction_complete* to complete the transaction.

## ***request_account_create*** transaction

The purpose of this transaction is to ask a local site to create an account for a user on a project. The XDCDB sends the account information in a *request_account_create* packet. The local site creates the account and sends the site-specific account information back to the XDCDB in a *notify_account_create* packet. The XDCDB then sends additional information about the account to the local site in a *data_account_create* packet. The local site finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| request_account_create | incoming |
| notify_account_create | outgoing |
| data_account_create | incoming |
| inform_transaction_complete | outgoing |

NOTE**:** although the XDCDB will not send a *request_account_create* packet before it has sent a corresponding *request_project_create* packet, it may send it before the *request_project_create* transaction has completed. Local sites must be prepared to delay processing of *request_account_create* packets until they have completed the corresponding *request_project_create* transaction.[15]

NOTE: If the project is active, a *request_account_create* implies account reactivation. If the account had been inactivated (by a *request_account_inactivate*), the site must reactivate the account as well as performing the other actions required by this transaction. The site will **not** receive a *request_account_reactivate* in addition to this transaction.

NOTE: If the project is inactive, a *request_account_create* may be sent to add a user to the project. The user should be added as inactive. A *request_account_inactivate* will follow upon completion of this transaction.

***request_account_create***

The *request_account_create* packet contains the *GrantNumber* for the related project as well as the information about the user for whom the account is to be created. If the XDCDB has been notified of the local project ID in a previous *request_project_create* transaction, then the *ProjectID* will be the local site project ID. Otherwise the *ProjectID* will not be present. This implies that sites must maintain a mapping between local projects and the *GrantNumber*.

**NOTE:** The *UserGlobalID* is the internal ID for the user within the XDCDB. It could be used to uniquely identify the user.

.

***request_account_create***

| tag | subtag | seq | value |
|---|---|---|---|
| **GrantNumber**[16] | | | AST040002 |
| NsfStatusCode | | | CN |
| ProjectID | | | afm |
| **ResourceList** | | 0 | oh.ncsa.xsede |

[15] A solution to this problem can be implemented in triggers on the transaction_tbl and packet_tbl. An example has been written for Postgres and can be found with the reference-implementation source ddl for Postgres.

[16] The *GrantNumber* must be used to tie the *request_acount_create* to the corresponding project since the *ProjectID* may not be present.

| | | | |
|---|---|---|---|
| RoleList | | 0 | co_pi |
| SitePersonId | **Site** | 0 | ANL |
| SitePersonId | **PersonID** | 0 | mshapiro |
| SitePersonId | **Site** | 1 | SDSC |
| SitePersonId | **PersonID** | 1 | 22998 |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| UserBusinessPhoneComment | | | Business phone on campus |
| UserBusinessPhoneExtension | | | 1514 |
| UserBusinessPhoneNumber[17] | | | (217) 244-0072 |
| UserCitizenship | | | 9US |
| UserCity | | | Champaign |
| UserCountry | | | 9US |
| UserDepartment | | | N.C.S.A |
| UserDnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| UserDnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| UserEmail | | | mshapiro@ncsa.uiuc.edu |
| UserFax | | | (217)-244-1987 |
| **UserFirstName** | | | Michael |
| UserHomePhoneComment | | | Has caller ID |
| UserHomePhoneExtension | | | none |
| UserHomePhoneNumber[18] | | | (217) 384-0000 |
| **UserLastName** | | | Shapiro |
| UserMiddleName | | | J. |
| **UserOrganization** | | | University of Illinois at Urbana-Champaign |
| **UserOrgCode** | | | 0017756 |
| UserPasswordAccessEnable[19] | | | 1 |
| UserPersonID[20] | | | 21619 |
| UserRequestedLoginList | | 0 | mshapiro |
| UserState | | | IL |
| UserStreetAddress | | | 137 CAB |
| UserStreetAddress2 | | | 605 E. Springfield Ave. |
| UserTitle | | | Senior Systems Engineer |
| UserZip | | | 61820 |

### *notify_account_create*

After having created an account, the local site must reply with a *notify_account_create* packet. This packet must contain the local project ID (*ProjectID*), the local ID for the account user

[17] *UserBusinessPhoneNumber* is required if the *UserBusinessPhoneExtension* is present.

[18] *UserHomePhoneNumber* is required if the *UserHomePhoneComment* or *UserHomePhoneExtension* is present.

[19] XML Boolean. Local sites should ignore this value. The TGCDB sets the value to **1**.

[20] If the TGCDB already has the local site person ID for the PI, then the *UserPersonID* tag will be present. Otherwise it will not be present.

(*UserPersonID*), and the login at the local site for the user (*UserRemoteSiteLogin*). It must also contain the *ResourceList* from the *request_account_create*. The table below lists the tags that could be included in this packet (the ones in **bold *must*** be included). However, in a reply to a *request_account_create,* if the user is already known to the XDCDB (based on the *UserPersonID* or the *UserGlobalID*), then all the *User* tags are ignored.

The *UserGlobalID* is the internal ID for the user within the XDCDB.

The *NsfStatusCode* must be a valid NSF Status Code (as defined in the XDCDB). The *UserOrgCode* must be a valid NSF organization code (as defined in the XDCDB). The *UserCountry* and *UserCitizenship* must be valid NSF country codes (as defined in the XDCDB). The *UserState* must be a valid state abbreviation (as defined in the XDCDB).

***notify_account_create***

| tag | subtag | seq | value |
|---|---|---|---|
| NsfStatusCode | | | CN |
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | viz.nist.xsede |
| ResourceLogin | **Resource** | 0 | portal.teragrid |
| ResourceLogin | **Login** | 0 | mshapiro |
| RoleList | | 0 | co_pi |
| StartDate[21] | | | 2003-12-16 |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| UserBusinessPhoneComment | | | Business phone on campus |
| UserBusinessPhoneExtension | | | 1514 |
| UserBusinessPhoneNumber[22] | | | (217) 244-0072 |
| UserCitizenship | | | 9US |
| UserCity | | | Champaign |
| UserCountry | | | 9US |
| UserDepartment | | | N.C.S.A |
| UserDnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| UserDnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| UserEmail | | | mshapiro@ncsa.uiuc.edu |
| UserFax | | | (217)-244-1987 |
| UserFirstName | | | Michael |
| UserGlobalID | | | 150 |

[21] An originating *notify_account_create* (i.e., one that is not sent in reply to a *request_account_create*) must specify the start date of the allocation to which the account will be associated.

[22] *UserBusinessPhoneNumber* is required if the *UserBusinessPhoneComment* or *UserBusinessPhoneExtension* is present.

| | | | |
|---|---|---|---|
| UserHomePhoneComment | | | Has caller ID |
| UserHomePhoneExtension | | | none |
| UserHomePhoneNumber[23] | | | (217) 384-0000 |
| UserLastName | | | Shapiro |
| UserMiddleName | | | J. |
| UserOrganization | | | University of Illinois at Urbana-Champaign |
| **UserOrgCode** | | | 0017756 |
| UserPasswordAccessEnable[24] | | | 1 |
| **UserPersonID** | | | 21619 |
| UserPosition | | | Research Programmer |
| UserRequestedLoginList | | 0 | mshapiro |
| **UserRemoteSiteLogin** | | | mshapiro |
| UserState | | | IL |
| UserStreetAddress | | | 137 CAB |
| UserStreetAddress2 | | | 605 E. Springfield Ave. |
| UserTitle | | | Senior Systems Engineer |
| UserUID | | | 12345 |
| UserZip | | | 61820 |

***data_account_create***

The XDCDB will reply with a *data_account_create* packet that contains all the DNs known by the XDCDB for the user.

***data_account_create***

| tag | subtag | seq | value |
|---|---|---|---|
| DnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| DnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |

The local site must then reply with an *inform_transaction_complete* to complete the transaction.

## ***request_project_inactivate*** transaction

The purpose of this transaction is to ask a local site to inactivate a project. The XDCDB sends the project information in a *request_project_inactivate* packet. The local site inactivates the project and all accounts on the project and sends a reply to the XDCDB in a *notify_project_inactivate* packet. The XDCDB finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| request_project_inactivate | incoming |
| notify_project_inactivate | outgoing |

[23] *UserHomePhoneNumber* is required if the *UserHomePhoneComment* or *UserHomePhoneExtension* is present.

[24] XML Boolean. The TGCDB ignores this tag.

| | |
|---|---|
| inform_transaction_complete | incoming |

***request_project_inactivate***

The *request_project_inactivate* packet contains information to be used by a local site to identify the project to be inactivated. The result of this request should be the inactivation of the project as well as inactivation of all accounts on the project.

***request_project_inactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | xx.ncsa.xsede |
| Comment | | | Project out of funds |
| AllocatedResource | | | staff.xsede |
| GrantNumber | | | AST040009 |
| StartDate | | | 2006-04-01 |
| EndDate | | | 2007-03-31 |
| ServiceUnitsAllocated | | | 50000 |
| ServiceUnitsRemaining | | | -2345 |

***notify_project_inactivate***

After having inactivated the project and all related accounts, the local site must reply with a *notify_project_inactivate* packet. This packet must contain the same project information as the incoming *request_project_inactivate*.

***notify_project_inactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | hc.ncsa.xsede |

## *request_project_reactivate* transaction

The purpose of this transaction is to ask a local site to reactivate a project that was previously inactivated. The XDCDB sends the project information in a *request_project_reactivate* packet. The local site reactivates the project and the account for the PI and sends a reply to the XDCDB in a *notify_project_reactivate* packet. The XDCDB finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| request_project_reactivate | incoming |
| notify_project_reactivate | outgoing |
| inform_transaction_complete | incoming |

***request_project_reactivate***

The *request_project_reactivate* packet contains information to be used by a local site to reactivate a project and the account for the PI on the project. No other accounts should be reactivated.

***request_project_reactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| PersonID[25] | | | 21619 |
| **ProjectID** | | | afm |
| Comment | | | PI begged for more time |
| **ResourceList** | | 0 | hc.ncsa.xsede |
| AllocatedResource | | | staff.xsede |
| GrantNumber | | | AST040009 |
| StartDate | | | 2006-04-01 |
| EndDate | | | 2007-03-31 |
| ServiceUnitsAllocated | | | 50000 |
| ServiceUnitsRemaining | | | 12345 |

***notify_project_reactivate***

After having reactivated a project, the local site must reply with a *notify_project_reactivate* packet. This packet must contain the same project information as the incoming *request_project_reactivate*.

***notify_project_reactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | hc.ncsa.xsede |

NOTE: an inactive project can also be reactivated with a *request_project_create* packet.

## ***request_account_inactivate*** transaction

The purpose of this transaction is to ask a local site to inactivate an account for a user on a project. The XDCDB sends the account information in a *request_account_inactivate* packet. The local site inactivates the account and sends a reply to the XDCDB in a *notify_account_inactivate* packet. The XDCDB finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| request_account_inactivate | incoming |
| notify_account_inactivate | outgoing |
| inform_transaction_complete | incoming |

***request_account_inactivate***

[25] The *PersonID* in this packet is the local site person ID for the PI on the project.

The *request_account_inactivate* packet contains information to be used by a local site to inactivate an account on a project.

***request_account_inactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |
| Comment | | | No longer on the project |
| **ResourceList** | | 0 | hc.ncsa.xsede |

***notify_account_inactivate***

After having inactivated an account, the local site must reply with a *notify_account_inactivate* packet. This packet must contain the same account information as the incoming *request_account_inactivate*.

***notify_account_inactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | hc.ncsa.xsede |

NOTE: it is important that the site inactivate the account, rather than remove it. It is possible that the account could be reactivated in the future. Also, account inactivation does not imply user inactivation. The *PersonID* (as well as the related login on the resource) must be retained for this user, since the XDCDB will retain them as well.

## ***request_account_reactivate*** transaction

The purpose of this transaction is to ask a local site to reactivate an account for a user on a project that was previously inactivated. The XDCDB sends the account information in a *request_account_reactivate* packet. The local site reactivates the account and sends a reply to the XDCDB in a *notify_account_reactivate* packet. The XDCDB finishes the transaction by sending an *inform_transaction_complete*.

NOTE: this transaction is included in this documentation for completeness only. The XDCDB currently will reactivate accounts using the *request_account_create* transaction.

The sequence of packets is

| | |
|---|---|
| request_account_reactivate | incoming |
| notify_account_reactivate | outgoing |
| inform_transaction_complete | incoming |

***request_account_reactivate***

The *request_account_reactivate* packet contains information to be used by a local site to reactivate an account on a project.

***request_account_reactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |
| Comment | | | Re-hired on the project |
| **ResourceList** | | 0 | hc.ncsa.xsede |

***notify_account_reactivate***

After having reactivated an account, the local site must reply with a *notify_account_reactivate* packet. This packet must contain the same account information as the incoming *request_account_reactivate*.

***notify_account_reactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | hc.ncsa.xsede |

## ***request_user_modify*** transaction

This transaction is used by the XDCDB to ask local sites to update all the information about a user. After a local site notifies the XDCDB with updates for a user, the XDCDB will send a *request_user_modify* packet to all the other sites with the updated information. After processing the updates, the local sites then send an *inform_transaction_complete* to the XDCDB to complete the transaction.

The sequence of packets is

| | |
|---|---|
| request_user_modify | incoming |
| inform_transaction_complete | outgoing |

***request_user_modify***

The *ActionType* will be one of **replace** or **delete**.

For **replace**, if a tag is present, the value associated with the tag at the local site must be replaced with the value in the packet; if a tag is not present, then the value associated with the tag at the local site must be deleted (or replaced with a null value). An exception is made for the *DnList* tag. If the *DnList* tag is present, DNs listed must be added to the grid mapfile; DNs in the grid mapfile which are not listed must be preserved. If the *DnList* tag is not present, all DNs in the grid mapfile must be preserved.

For **delete**, the only optional tag will be *DnList* which will specify DNs that must be deleted from the grid mapfile[26].

***request_user_modify***

| tag | subtag | seq | value |
|---|---|---|---|
| **ActionType** | | | replace |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| BusinessPhoneComment | | | Business phone on campus |
| BusinessPhoneExtension | | | 1514 |
| BusinessPhoneNumber[27] | | | (217) 244-0072 |
| City | | | Champaign |
| Country | | | 9US |
| Department | | | N.C.S.A |
| DnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| DnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| Email | | | mshapiro@ncsa.uiuc.edu |
| Fax | | | (217)-244-1987 |
| FirstName | | | Michael |
| HomePhoneComment | | | Has caller ID |
| HomePhoneExtension[28] | | | none |
| HomePhoneNumber[29] | | | (217) 384-0000 |
| LastName | | | Shapiro |
| MiddleName | | | J. |
| NsfStatusCode | | | CN |
| Organization | | | University of Illinois at Urbana-Champaign |
| OrgCode | | | 0017756 |
| **PersonID** | | | 21619 |
| State | | | IL |
| StreetAddress | | | 137 CAB |
| StreetAddress2 | | | 605 E. Springfield Ave. |
| Title | | | Senior Systems Engineer |
| Zip | | | 61820 |

The local site must then reply with an *inform_transaction_complete* to complete the transaction.

[26] Only the specific DN strings listed should be deleted. Other DNs that some applications may consider to be equivalent, but are not listed should not be deleted. For example, even though the following DNs may be considered equivalent by some applications, the TGCDB considers them as distinct:
/C=US/O=Pittsburgh Supercomputing Center/OU=PSC Kerberos Certification Authority/CN=jsmith/USERID=jsmith/Email=jsmith@PSC.EDU
/C=US/O=Pittsburgh Supercomputing Center/OU=PSC Kerberos Certification Authority/CN=jsmith/UID=jsmith/emailAddress=jsmith@PSC.EDU

[27] *BusinessPhoneNumber* is required if *BusinessPhoneComment* or *BusinessPhoneExtension* is present.

[28] It is not necessary to specify the extension if the phone number does not have an extension.

[29] *HomePhoneNumber* is required if the *HomePhoneComment* or *HomePhoneExtension* is present.

# Local Site Transactions

Transactions that originate at a local site are notifications to the XDCDB of project and account creation, or modifications to information about a project, account, or a user. Often this will then cause the XDCDB to send requests to the other sites to do the same. For example, one XSEDE site creates a project and notifies the XDCDB about the project. The XDCDB then would send requests to all the other sites to create that project at their site.

For these types of transactions, local sites will have to create a transaction and put an initial outgoing packet into the transaction[30]. Then, when an incoming reply packet arrives, it would be processed and an outgoing packet generated in reply[31].

## ***notify_project_create*** transaction

The purpose of this transaction is for a local site to notify the XDCDB of the creation of an XSEDE project and related account for the PI of the project. Projects are never created by XDCDB. They must first be created at one of the local sites. That site then notifies the XDCDB of the project.

For any given project, one site must be designated as the originating site for the project. For example, a given site may be designated as the originating site for the peer-reviewed projects from a review board meeting. Which site is designated is an XSEDE policy decision that is outside the scope of this document.

The local site creates the project and account and sends the site-specific project and account information to the XDCDB in a *notify_project_create* packet[32]. The XDCDB then sends additional information about the project to the local site in a *data_project_create* packet. The local site finishes the transaction by sending an *inform_transaction_complete*.

NOTE: a *notify_project_create* also implies project reactivation. If a site receives a *notify_project_create* packet for an inactive project, the site must reactivate the project in addition to performing the other actions required by this packet. In this case, the site will **not** receive a *request_project_reactivate.*

The sequence of packets is

| | |
|---|---|
| notify_project_create | outgoing |
| data_project_create | incoming |
| inform_transaction_complete | outgoing |

[30] The details of both transaction and packet creation as implementation specific, but a sketch of how these are done can be found in the *Implementing AMIE* document.
[31] Unless the incoming packet does not expect a reply (which is the case for *inform_transaction_complete* packets).
[32] The local site must first create a transaction before it can create the *notify_project_create* packet.

### *notify_project_create*

After creating a project (and an account for the PI), the local site sends a *notify_project_create* packet to the XDCDB. This packet must contain the *GrantNumber* from the *request_project_create*, the local project ID (*ProjectID*), the local ID for the PI (*PiPersonID*) which should be a locally unique identifier for the PI at the local site, and the login at the local site for the PI (*PiRemoteSiteLogin*). The *ResourceList* must have a single item (even though it is a list).

The table below lists the tags that could be included in this packet (the ones is **bold *must*** be included). If the PI is already known to the XDCDB (based on the *PiPersonID* or the *PiGlobalID*), then all the *Pi* tags are ignored, except *PiPersonID, PiGlobalID,* and *PiRemoteSiteLogin*. Otherwise the PI information is recorded at the XDCDB for the *PiPersonID*.

The *PiGlobalID* is the internal ID for the PI within the XDCDB.

The *NsfStatusCode* must be a valid NSF Status Code (as defined in the XDCDB). The *PiOrgCode* must be a valid NSF organization code (as defined in the XDCDB). The *PiCountry* and *PiCitizenship* must be valid NSF country codes (as defined in the XDCDB). The *PiState* must be a valid state abbreviation (as defined in the XDCDB).

***notify_project_create***

| tag | subtag | seq | value |
|---|---|---|---|
| Abstract | | | Some text …. |
| **AccountActivityTime** | | | 2003-12-16T08:00:00Z |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| **BoardType** | | | NRAC |
| EndDate | | | 2013-12-31 |
| GrantType | | | Science Gateway |
| **GrantNumber** | | | AST040002 |
| NsfStatusCode | | | CN |
| **PfosNumber** | | | 120 |
| PiBusinessPhoneExtension | | | 7570 |
| PiBusinessPhoneNumber[33] | | | (217) 244-0072 |
| PiCitizenship | | | 9US |
| PiCity | | | Champaign |
| PiCountry | | | 9US |
| PiDepartment | | | N.C.S.A |
| PiDnList | | 0 | /C=US/O=NCSA/CN=Steven James Quinn |
| PiDnList | | 1 | /C=US/O=SDSC/CN=Steven James Quinn |
| PiDnList | | 2 | /C=US/O=PSC/CN=Steven James Quinn |
| PiDnList | | 3 | /C=US/O=ANL/CN=Steven James Quinn |
| PiEmail | | | squinn@ncsa.uiuc.edu |
| PiFax | | | (217)-244-1987 |

[33] *PiBusinessPhoneNumber* is required if the *PiBusinessPhoneExtension* is present.

| | | | |
|---|---|---|---|
| **PiFirstName** | | | Steven |
| PiGlobalID | | | 70 |
| PiHomePhoneExtension | | | none |
| PiHomePhoneNumber[34] | | | (217) 222-0000 |
| **PiLastName** | | | Quinn |
| PiMiddleName | | | James |
| **PiOrganization** | | | University of Illinois at Urbana-Champaign |
| **PiOrgCode** | | | 0017756 |
| **PiPersonID** | | | 6751 |
| PiPosition | | | Research Programmer |
| **PiRemoteSiteLogin** | | | squinn |
| PiRequestedLoginList | | 0 | squinn |
| PiState | | | IL |
| PiStreetAddress | | | 131 CAB |
| PiStreetAddress2 | | | 605 E. Springfield Ave. |
| PiTitle | | | n/a |
| PiUID | | | 12456 |
| PiZip | | | 61820 |
| ProjectGID | | | 123 |
| **ProjectID** | | | afm |
| **ProjectTitle** | | | Planetary Motion |
| **RecordID** | | | 123456 |
| **ResourceList** | | 0 | staff.xsede |
| RoleList | | 0 | allocation_manager |
| **ServiceUnitsAllocated** | | | 99999 |
| Sfos | **Number** | 0 | 122 |
| Sfos | **Number** | 1 | 124 |
| **StartDate** | | | 2003-12-16 |

***data_project_create***

The XDCDB will reply with a *data_project_create* packet that contains all the DNs known by the XDCDB for the project PI.

***data_project_create***

| tag | subtag | seq | value |
|---|---|---|---|
| DnList | | 0 | /C=US/O=NCSA/CN=Steven James Quinn |

[34] *PiHomePhoneNumber* is required if the *PiHomePhoneExtension* is present.

| | | | |
|---|---|---|---|
| DnList | | 1 | /C=US/O=PSC/CN=Steven James Quinn |
| DnList | | 2 | /C=US/O=SDSC/CN=Steven James Quinn |
| DnList | | 3 | /C=US/O=ANL/CN=Steven James Quinn |
| **ProjectID** | | | afm |

The local site must then reply with an *inform_transaction_complete* to complete the transaction.

## *notify_account_create* transaction

The purpose of this transaction is for a local site to notify the XDCDB of the creation of an account for an XSEDE project. Accounts are never created by XDCDB. They must first be created at one of the local sites. That site, then, notifies the XDCDB of the account.

For any given account, one site must be designated as the originating site for the account. In principle, any site could create an account. In practice it may be the site that created the project which also creates the accounts for the project. Which site is designated is an XSEDE policy decision that is outside the scope of this document.

The local site creates an account and sends the site-specific account information to the XDCDB in a *notify_account_create* packet[35]. The XDCDB then sends additional information about the account to the local site in a *data_account_create* packet. The local site finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| notify_account_create | outgoing |
| data_account_create | incoming |
| inform_transaction_complete | outgoing |

### *notify_account_create*

After having created an account, the local site sends a *notify_account_create* packet to the XDCDB. This packet must contain the local project ID (*ProjectID*) and the local ID for the account user (*UserPersonID*) and the login at the local site for the user (*UserRemoteSiteLogin*). The table below lists the tags that could be included in this packet (the ones in **bold *must*** be included). However, if the user is already known to the XDCDB (based on the *UserPersonID* or the *UserGlobalID*), then all the *User* tags are ignored.

The *UserGlobalID* is the internal ID for the user within the XDCDB.

[35] The local site must first create a transaction before it can create the *notify_account_create* packet. It must also make sure that the related *notify_project_create* transaction has been completed. The transaction dependency mechanism (described in Implementing AMIE) can be used to delay the transmission of the *notify_account_create* until the completion of the related *notify_project_create* transaction.

The *NsfStatusCode* must be a valid NSF Status Code (as defined in the XDCDB). The *UserOrgCode* must be a valid NSF organization code (as defined in the XDCDB). The *UserCountry* and *UserCitizenship* must be valid NSF country codes (as defined in the XDCDB). The *UserState* must be a valid state abbreviation (as defined in the XDCDB).

***notify_account_create***

| tag | subtag | seq | value |
|---|---|---|---|
| **AccountActivityTime** | | | 2003-12-16T08:00:00Z |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| NsfStatusCode | | | CN |
| **ProjectID** | | | afm |
| **ResourceList** | | 0 | staff.xsede |
| RoleList | | 0 | co_pi |
| UserBusinessPhoneExtension | | | 1514 |
| UserBusinessPhoneNumber[36] | | | (217) 244-0072 |
| UserCitizenship | | | 9US |
| UserCity | | | Champaign |
| UserCountry | | | 9US |
| UserDepartment | | | N.C.S.A |
| UserDnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| UserDnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| UserEmail | | | mshapiro@ncsa.uiuc.edu |
| UserFax | | | (217)-244-1987 |
| **UserFirstName** | | | Michael |
| UserGlobalID | | | 150 |
| UserHomePhoneExtension | | | none |
| UserHomePhoneNumber[37] | | | (217) 384-0000 |
| **UserLastName** | | | Shapiro |
| UserMiddleName | | | J. |
| **UserOrganization** | | | University of Illinois at Urbana-Champaign |
| **UserOrgCode** | | | 0017756 |
| UserPasswordAccessEnable[38] | | | 1 |
| **UserPersonID** | | | 21619 |
| UserPosition | | | Research Programmer |
| UserRequestedLoginList | | 0 | mshapiro |
| **UserRemoteSiteLogin** | | | mshapiro |
| UserState | | | IL |
| UserStreetAddress | | | 137 CAB |
| UserStreetAddress2 | | | 605 E. Springfield Ave. |

[36] *UserBusinessPhoneNumber* is required if the *UserBusinessPhoneExtension* is present.
[37] *UserHomePhoneNumber* is required if the *UserHomePhoneExtension* is present.
[38] XML Boolean. The TGCDB ignores this tag.

| | | | |
|---|---|---|---|
| UserTitle | | | Senior Systems Engineer |
| UserUID | | | 12345 |
| UserZip | | | 61820 |

***data_account_create***

The XDCDB will reply with a *data_account_create* packet that contains all the DNs known by the XDCDB for the user.

***data_account_create***

| tag | subtag | seq | value |
|---|---|---|---|
| DnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| DnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |

The local site must then reply with an *inform_transaction_complete* to complete the transaction.

## *notify_project_inactivate* transaction

The purpose of this transaction is for a local site to notify the XDCDB of the inactivation of an XSEDE project.

The local site inactivates a project and all accounts on the project and sends the site-specific project information to the XDCDB in a *notify_project_inactivate* packet. The XDCDB then finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| notify_project_inactivate | outgoing |
| inform_transaction_complete | incoming |

***notify_project_inactivate***

After having inactivated a project, the local site sends a *notify_project_inactivate* packet to the XDCDB. This packet must contain the local project ID (*ProjectID*), a single resource (*ResourceList*) and the time that the account was inactivated (*AccountActivityTime*). The table below lists the tags that could be included in this packet (the ones in **bold *must*** be included).

***notify_project_inactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **AccountActivityTime** | | | 2004-10-21T09:00:00Z |
| **ProjectID** | | | afm |
| Comment | | | Project out of funds |
| **ResourceList** | | 0 | hc.ncsa.xsede |

## ***notify_project_reactivate*** transaction

The purpose of this transaction is for a local site to notify the XDCDB of the reactivation of an XSEDE project.

The local site reactivates a project and the account for the PI and sends the site-specific project information to the XDCDB in a *notify_project_reactivate* packet. The XDCDB then finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| notify_project_reactivate | outgoing |
| inform_transaction_complete | incoming |

***notify_project_reactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **AccountActivityTime** | | | 2004-12-21T09:00:00Z |
| **ProjectID** | | | afm |
| Comment | | | PI needs more time |
| **ResourceList** | | 0 | hc.ncsa.xsede |

NOTE: projects can also be reactivated via *request_project_create* transactions.

## ***notify_account_inactivate*** transaction

The purpose of this transaction is for a local site to notify the XDCDB of the inactivation of an account for an XSEDE project.

The local site inactivates an account and sends the site-specific account information to the XDCDB in a *notify_account_inactivate* packet. The XDCDB then finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| notify_account_inactivate | outgoing |
| inform_transaction_complete | incoming |

***notify_account_inactivate***

After having inactivated an account, the local site sends a *notify_account_inactivate* packet to the XDCDB. This packet must contain the local project ID (*ProjectID*), the local ID for the account user (*PersonID*), and the allocated resource (*ResourceList*). The table below lists the tags that could be included in this packet (the ones in **bold *must*** be included).

***notify_account_inactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |
| Comment | | | No longer on the project |
| **ResourceList** | | 0 | staff.xsede |

## ***notify_account_reactivate*** transaction

The purpose of this transaction is for a local site to notify the XDCDB of the reactivation of an account for an XSEDE project.

The local site reactivates an account and sends the site-specific account information to the XDCDB in a *notify_account_reactivate* packet. The XDCDB then finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| notify_account_reactivate | outgoing |
| inform_transaction_complete | incoming |

***notify_account_reactivate***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **ProjectID** | | | afm |
| Comment | | | Re-hired on the project |
| **ResourceList** | | 0 | staff.xsede |

NOTE: accounts can also be reactivated via *notify_account_create* transactions.

## ***notify_project_usage*** transaction

On a regular basis, the local sites must send usage information for XSEDE projects to the central database. The local site sends project usage information to the XDCDB in a *notify_project_usage* packet[39]. The XDCDB processes the usage information and finishes the transaction by sending an *inform_transaction_complete* to the local site.

The sequence of packets is

| | |
|---|---|
| notify_project_usage | outgoing |
| inform_transaction_complete | incoming |

[39] The local site must first create a transaction before it can create the *notify_project_usage* packet.

***notify_project_usage***

This packet is used to send job information, as well as credits and debits. The *UsageType* indicates the purpose of the packets. The *normal* usage type is for a single job. Credit types are *credit*, *refund* and *storage-credit*. Debit types are *debit*, *reservation*, and *storage-debit.*

The following tags are allowed in the *notify_project_usage* job usage packet:

***notify_project_usage (job)***

| tag | subtag | seq | value |
|---|---|---|---|
| **UsageType** | | | **normal** |
| **ProjectID** | | | afm |
| **Charge** | | | 23.5 |
| CpuDuration | System | | PT00H23M45S |
| CpuDuration | User | | PT10H34M01S |
| **EndTime** | | | 2004-02-05T08:02:40Z |
| JobIdentity | **LocalJobID**[40] | | 55145 |
| JobIdentity | GlobalJobID[41] | | 12345TG |
| JobName | | | myjob |
| **MachineName**[42] | | | tg.sdsc.edu |
| Memory | | | 12345000 |
| **NodeCount**[43] | | | 1 |
| Processors | | | 1 |
| **Queue** | | | xque |
| **RecordIdentity** | **CreateTime** | | 2004-02-05T00:02:40Z |
| **RecordIdentity** | **RecordID**[44] | | 55145.tg-master@tg.sdsc.edu |
| **StartTime** | | | 2004-02-05T08:02:33Z |
| **SubmitTime** | | | 2004-02-05T08:01:34Z |
| **UserLogin** | | | ux453932 |
| **WallDuration** | | | PT00H00M01S |
| Attribute | **Name** | 0 | user |
| Attribute | **Value** | 0 | steve |

Charges are computed by the central database based on formulas associated with the *MachineName*[45]. The *ExecHost* list should contain the complete list of hosts that the job ran on. This list is only required if *Charge* tag is not present and the central database needs the list of hosts to compute the charges. Moreover, the number of processors and reserved memory on each host may also be needed to compute the charges.

[40] The *LocalJobID* should be the job ID issued by the local job resource manager.

[41] The *GlobalJobID* could be the Globus job ID.

[42] The *MachineName* is the name of an XSEDE resource which was used to initiate the job. It can be a real machine or it can refer to a cluster of machines. It must be a valid resource in the TGCDB.

[43] NodeCount must be greater than zero.

[44] The *RecordID* can be anything at all. It is not recorded in the TGCDB.

[45] The TGCDB must be configured before usage is sent. Sites must coordinate with the TGCDB administrators to get their resources and formulas configured for usage, be it based on the *Charge* tag or a based on *WallDuration*, etc.

The following tags are allowed in the *notify_project_usage* credit/debit usage packet:

***notify_project_usage (credit/debit)***

| tag | subtag | seq | value |
|---|---|---|---|
| **UsageType** | | | **refund** |
| **ProjectID** | | | afm |
| **Charge** | | | 23.5 |
| Comment | | | comments … |
| **RecordIdentity** | **RecordID** | | credit-debit-123 |
| **MachineName** | | | tg.sdsc.edu |
| StartTime | | | 2004-02-05T08:02:33Z |
| UserLogin | | | ux453932 |

The *Charge* field is used to specify the amount of the credit or debit. It must be positive. The *RecordID* field is used to uniquely identify this credit/debit on the *MachineName.* The *StartTime* is used to select the allocation to receive the credit/debit. If *StartTime* is missing, then the current allocation is selected. The *UserLogin* identifies the person who is given the credit/debit. If *UserLogin* is missing, then the PI for the project is given the credit/debit.

NOTE: If a credit is received for project currently inactivated due to being out of funds, the credit may cause the project to be reactivated. A debit may similarly inactivate a project. In general, credits and debits should be sent and processed as soon as reasonably possible so that XDCDB and the services that rely on it, such as the TG User Portal and tgusage, can present accurate information.

However, sites should note two implications of project reactivation via credits. First, small refunds on projects that are at or near their allocation limits may cause a "flip-flop" cycle of project reactivation by credits and inactivation as subsequent job charges put the project over its allocation limit. Similarly, refunds on inactivated projects that are near the end of an allocation period (i.e., projects that may expire within a few days) may reactivate the project, only to have it inactivated when it expires.

The following "best practices" for credits/refunds are recommended:

- Sites should send credits/refunds for the largest appropriate amount in the fewest packets. For example, send one *refund* NPU for 2000 SUs rather than 10 *refund* NPUs for 200 SUs.
- For refunds on overspent projects that are very close to expiration, it may be practical to re-activate the project locally until the expiration date has passed, then send up a "back-dated" refund or credit to apply to the allocation. For grid allocations, sites should use their best judgment in these situations, since the credit/debit reactivation/inactivation may affect a project's ability to run at another site.

## ***notify_user_modify*** transaction

The purpose of this transaction is to update information about a user in the central database.

The local site sends project usage information to the XDCDB in a *notify_project_usage* packet. The XDCDB processes the usage information and finishes the transaction by sending an *inform_transaction_complete* to the local site.

The sequence of packets is

| | |
|---|---|
| notify_project_usage | outgoing |
| inform_transaction_complete | incoming |

If a local site has updated information about an XSEDE user, it sends the updates to the XDCDB in a *notify_user_modify* packet. The XDCDB updates the information about the user and finishes the transaction by sending an *inform_transaction_complete*.

The sequence of packets is

| | |
|---|---|
| notify_user_modify | outgoing |
| inform_transaction_complete | incoming |

***notify_user_modify***

The following tags are allowed in this packet. Note that the *ActionType* and *PersonID* must be specified.

***notify_user_modify***

| tag | subtag | seq | value |
|---|---|---|---|
| **ActionType**[46] | | | replace |
| AcademicDegree | **Degree** | 0 | MS |
| AcademicDegree | **Field** | 0 | Math |
| AcademicDegree | **Degree** | 1 | PhD |
| AcademicDegree | **Field** | 1 | Math |
| BusinessPhoneComment | | | Business phone on campus |
| BusinessPhoneExtension | | | 1514 |
| BusinessPhoneNumber[47] | | | (217) 244-0072 |
| City | | | Champaign |
| Country | | | 9US |
| Department | | | N.C.S.A |
| DnList | | 0 | /C=US/O=NCSA/CN=Michael Shapiro |
| DnList | | 1 | /C=US/O=SDSC/CN=Michael Shapiro |
| Email | | | mshapiro@ncsa.uiuc.edu |
| Fax | | | (217)-244-1987 |
| FirstName | | | Michael |
| HomePhoneComment | | | Has caller ID |
| HomePhoneExtension[48] | | | none |

[46] *ActionType* must be one of **add, delete**, or **replace**.
[47] *BusinessPhoneNumber* is required if *BusinessPhoneComment* or *BusinessPhoneExtension* is present.
[48] It is not necessary to specify the extension if the phone number does not have an extension.

| | | | |
|---|---|---|---|
| HomePhoneNumber[49] | | | (217) 384-0000 |
| LastName | | | Shapiro |
| MiddleName | | | J. |
| Organization | | | University of Illinois at Urbana-Champaign |
| OrgCode | | | 0017756 |
| **PersonID** | | | 21619 |
| State | | | IL |
| StreetAddress | | | 137 CAB |
| StreetAddress2 | | | 605 E. Springfield Ave. |
| Title | | | Senior Systems Engineer |
| Zip | | | 61820 |

The following table defines the behavior for the ActionTypes (add, replace, delete) for each of the allowed tags:

| tag | **add** or **replace** | **delete** |
|---|---|---|
| BusinessPhoneNumber | All business phone numbers are replaced with the single number, extension, comment | All business phone numbers are deleted |
| HomePhoneNumber | All home phone numbers are replaced with the single number, extension, comment | All home phone numbers are deleted |
| Fax | All fax numbers are replaced with the single fax number | All fax numbers are deleted |
| Email | All email addresses are replaced with the single email address | All email addresses are deleted |
| StreetAddress, StreetAddress2, City, State, Zip, Country | All addresses are replaced with the single address | All addresses are deleted |
| Title | Title set to value | Title set to null |
| Department | Department set to value | Department set to null |
| LastName | Last name set to value | **Illegal** – transaction failure |
| FirstName | First name set to value | First name set to null |
| MiddleName | Middle name set to value | Middle name set to null |
| OrgCode | OrgCode set to value | OrgCode set to null |
| DnList | New DNs are added[50] | Matching DNs are deleted |
| AcademicDegree | New degrees are added | Ignored |

Note that, while the XDCDB supports multiple phone numbers, addresses and email, AMIE does not. Thus the implementation of AMIE for the XDCDB also only allows one phone number (of each type – business, home, fax), one postal address, and one email address.

[49] *HomePhoneNumber* is required if the *HomePhoneComment* or *HomePhoneExtension* is present.

[50] Only if the *ActionType* is **add**. The *ActionType* **replace** is illegal and will cause a transaction failure.

Also, if the ActionType is **delete**, it is only necessary that the tag be present. The value of the tag is ignored (except for DnList which requires the deleted DNs to match existing DNs).

The XDCDB will reply with an *inform_transaction_complete* to complete the transaction.

## Duplicate People

Occasionally, multiple entries for the same person get created in the XDCDB[51]. The three transactions described in this section (*notify_person_duplicate*, *request_person_merge*, and *notify_person_ids*) are designed to automated the process of merging duplicate people into a single person.

One implication of this design is that a person in the XDCDB now can have multiple local person IDs (one of which is the primary local person ID) and multiple usernames on any resource. The sites can use any of the local person IDs or usernames in the packets they send up (*notify_account_create*, *notify_project_usage*, etc). The XDCDB will only use the primary local person ID in packets sent to sites (*request_account_create*, *data_account_create*, etc), except for the *request_user_modify* (which will be sent for each local person ID, primary as well as non-primary).

### ***notify_person_duplicate*** transaction

The purpose of this transaction is to notify the central database that there are duplicate people in the database.

A site sends the local IDs of two people that are duplicates to the XDCDB in a *notify_person_duplicate* packet[52]. The XDCDB will merge the two people and finish the transaction by sending an *inform_transaction_complete* to the site.

The sequence of packets is

| | |
|---|---|
| notify_person_duplicate | outgoing |
| inform_transaction_complete | incoming |

***notify_person_duplicate***

| tag | subtag | seq | value |
|---|---|---|---|
| GlobalID1 | | | 125 |
| PersonID1 | | | 21619 |
| GlobalID2 | | | 133 |
| PersonID2 | | | 21620 |

The PersonID1 and PersonID2 values are the local person IDs for the two people. The GlobalID1 and GlobalID2 are the XDCDB internal ids for the two people[53]. The XDCDB will merge the two

[51] This can happen when the vetting process for a new account does not detect the person in the database.

[52] The local site must be trusted by the TGCDB to properly vet users and detect duplicates.

people into a single person. The XDCDB makes the determination about how to merge the two people. One will be kept and the other deleted.[54]

## ***request_person_merge*** transaction

The purpose of this transaction is to inform sites that the central database has merged two duplicate people into a single person.

The XDCDB sends the IDs of the two people that were merged into a single person via a *request_person_merge* packet. The site finishes the transaction by sending an *inform_transaction_complete* to the XDCDB.

The sequence of packets is

| | |
|---|---|
| request_person_merge | incoming |
| inform_transaction_complete | outgoing |

***request_person_merge***

| tag | subtag | seq | value |
|---|---|---|---|
| **KeepGlobalID** | | | 1234 |
| **KeepPersonID** | | | 21620 |
| KeepPortalLogin | | | mshapiro |
| **DeleteGlobalID** | | | 431 |
| **DeletePersonID** | | | 21619 |
| DeletePortalLogin | | | mshapiro2 |

The KeepPersonID and DeletePersonID values are the local person IDs for the two people. The KeepGlobalID and DeleteGlobalID are the XDCDB internal person IDs for the two people. The KeepPortalLogin and the DeletePortalLogin are the portal usernames for the two people. The XDCDB has merged DeleteGlobalID into KeepGlobalID and deleted DeleteGlobalID. The accounts and usage for the deleted person are transferred to the kept person.

The local site identifiers (local person ID, usernames on resources) from the deleted person are transferred to the kept person (which implies that the kept person now has multiple local IDs and multiple usernames). The local person ID for the kept person is the primary local person ID.

The XDCDB will ensure that the kept person has a portal login. If the kept person has a portal login, then it will be retained. If not, but the deleted person did, then the portal login for the deleted person will be transferred to the kept person. The KeepPortalLogin (if present) indicates the portal login retained. The DeletePortalLogin (if present) indicates the portal login deleted.

[53] Even though the GlobalIDs and the PersonIDs are specified as optional, one of GlobalID1 and PersonID1 must be specified, as well as one of GlobalID2 and PersonID2.

[54] Once the merge is completed at the TGCDB, *request_person_merge* transactions will be sent to each site informing them of the person kept and the person deleted.

Sites can handle the merge locally as they wish. If they decide to keep both people as distinct people, they can use any of the local IDs and usernames to refer to the single (merged) person. If they wish to merge the people locally, they can notify the XDCDB of the local ID(s) and username(s) they wish the XDCDB to have for the single person via a *notify_person_ids* transaction. The XDCDB will use the primary local person ID in all packets sent to sites (except for *request_user_modify* packets – one will be sent for each local person ID).

NOTE: The XDCDB will not send this packet to a site if the XDCDB does not have a local person ID for both people (i.e., when the site has not added one of the people to their local accounting system.)

## ***notify_person_ids*** transaction

The purpose of this transaction is to notify the central database of the local site person ID(s) and username(s) to be associated with a person. It can be used after processing a *request_user_merge* or independently to change the local person ID or usernames at any time.

A site sends a primary local ID (and possibly other IDs) as well as the username(s) on the site's resources to be associated with the person. The XDCDB will assign the IDs and usernames to the specified person and finish the transaction by sending an *inform_transaction_complete* to the site.

The sequence of packets is

| | |
|---|---|
| notify_person_ids | outgoing |
| inform_transaction_complete | incoming |

***notify_person_ids***

| tag | subtag | seq | value |
|---|---|---|---|
| **PersonID** | | | 21619 |
| **PrimaryPersonID** | | | 21619 |
| PersonIdList | | 0 | 21620 |
| PersonIdList | | 1 | 21800 |
| RemoveResourceList | | 0 | hc.ncsa.xsede |
| ResourceLoginList | **Resource** | 0 | hc.ncsa.xsede |
| ResourceLoginList | **Login** | 0 | mshapiro |
| ResourceLoginList | UID | 0 | 21619 |
| ResourceLoginList | **Resource** | 1 | hc.ncsa.xsede |
| ResourceLoginList | **Login** | 1 | mshapiro2 |
| ResourceLoginList | UID | 1 | 21620 |

The PersonID is any local person ID that identifies the person in the TGCDGB. The PrimaryPersonID is the local person ID that is to be used as the primary. If a PersonIdList is provided, then these person IDs are also associated with the person. Any other person ID that the XDCDB has that are not the primary and not listed in the PersonIdList will be deleted.

The RemoveResourceList lists resources from which all usernames for this person are to be deleted.

The ResourceLoginList[55] lists usernames (and optional UIDs) to be associated with the person on the specified resources.

[55] The ResourceLoginList only adds usernames (or updates the UID). It does not remove existing usernames. To remove existing usernames, the RemoveResourceList must be specified.