AMIE RabbitMQ Implementation for XSEDE SSO Hub

Table of Contents

[AMIE Introduction 2](#_Toc525541)

[AMIE RabbitMQ Transfer Setup Instructions 3](#_Toc525542)

[AMIE RabbitMQ Transfer Reference Software Installation 3](#_Toc525543)

[Installation 4](#_Toc525544)

[Existing AMIE users 7](#_Toc525545)

[AMIE Transactions Implementation 8](#_Toc525546)

[Project Create transaction 8](#_Toc525547)

[Example Project Create Transaction 9](#_Toc525548)

[Account Create Transaction 12](#_Toc525549)

[Example Account Create Transaction 13](#_Toc525550)

[Error Handling 17](#_Toc525551)

[RabbitMQ 17](#_Toc525552)

[RabbitMQ Message Format 17](#_Toc525553)

[Appendix 1: Packet Definitions 18](#_Toc525554)

[Request Project Create definition 18](#_Toc525555)

[Notify Project Create Definition 20](#_Toc525556)

[Data Project Create Definition 23](#_Toc525557)

[Inform Transaction Complete Definition 23](#_Toc525558)

[Request Account Create Definition 23](#_Toc525559)

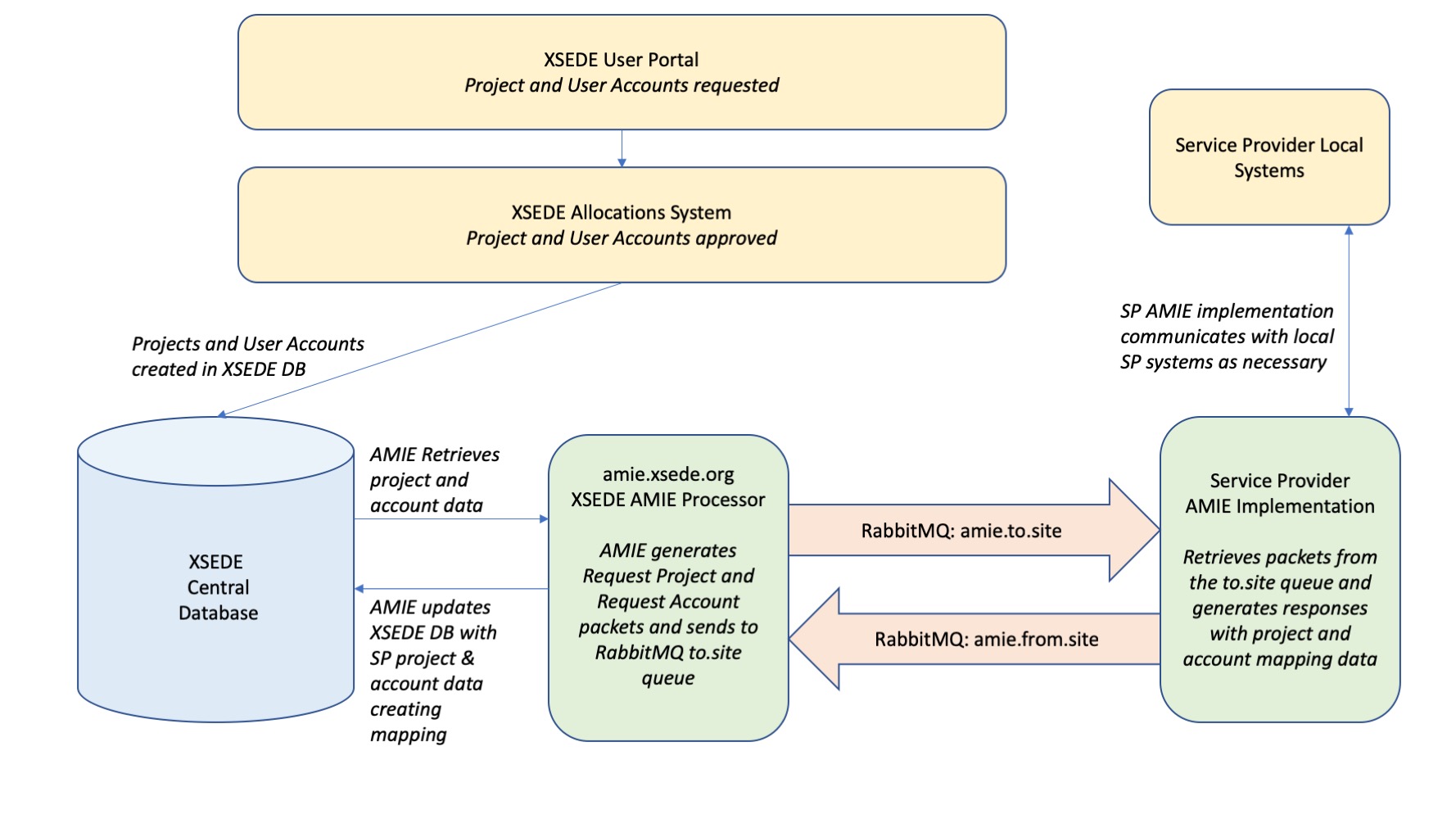
[Notify Account Create Definition 25](#_Toc525560)

[Data Account Create Definition 26](#_Toc525561)

# AMIE Introduction

XSEDE uses the Account Management Information Exchange (AMIE) system to communicate project and user information with local service providers and create mappings between local accounts and XSEDE accounts. Once mappings are established AMIE can also be used to communicate project usage and project allocation status.

Projects and user accounts are initially requested via the XSEDE User Portal. Requested projects and user accounts are then processed by XSEDE Administrators through the XSEDE Resource Allocation System. Approved projects and accounts are then communicated to the local service providers via AMIE. Finally the local service providers reply via AMIE with local mapping information for the projects and users.



AMIE uses an XML protocol used to send project and account information between XSEDE and Service Providers. Historically these packets have been transferred via ssh. This document describes using RabbitMQ as the data transfer mechanism.

AMIE has defined transaction sequences of data packets to accomplish different account management tasks. There are two packet transaction sequences that you will need to implement for SSO. More may be needed later if your site utilizes more account management functions through XSEDE.

The two transaction sequences that you will need to implement to start are Project Create and Account Create. Project Create will create a project with a Project Lead (Principal Investigator). Account Create will add additional users to the project. These packet sequences allow XSEDE to map XSEDE user accounts with your local user accounts to allow for SSO via XSEDE. Project create transactions occur on initial creation of a project and when the project is renewed or extended.

XSEDE provides a reference implementation to handle the RabbitMQ transfers that you are free to use. The implementation puts incoming data packets into a received directory for you and will read and send your responses from an outbox directory. You will still be responsible for generating the responses.

# AMIE RabbitMQ Transfer Setup Instructions

These are the steps needed to interact with XSEDE via AMIE

1. Request XSEDE AMIE setup.
   1. You will need an AMIE name for your organization.
      1. Send request for an AMIE name to help@xsede.org
   2. You will need RabbitMQ queues set up for your organization
      1. Authentication to RabbitMQ is done using an SSL certificate. When you have an SSL Certificate ready to use please send [help@xsede.org](mailto:help@xsede.org) a request for AMIE rabbitmq queues and include the Subject DN of the certificate which can be found with the command.

**openssl x509 -noout -in <certificate file name with full path> -subject**

1. Install the AMIE RabbitMQ Transfer Reference Software as explained in the next section.
   1. If your site is a past AMIE site and still has your AMIE installation and implementation you can follow the install section and then the Existing AMIE users section and your setup will be complete as your previous implementation of packet handling can still be used.
2. Implement AMIE packet handling as explained in the AMIE Transactions Implementation section.

# AMIE RabbitMQ Transfer Reference Software Installation

We have provided a reference implementation for interacting with the RabbitMQ queues. The reference implementation is available at <https://github.com/ResearchComputing/amie_rabbitmq>. It will read the packet XML data from the read queue and write it to xml files in a directory you define on your file system. It will also read your response XML files from a directory you define and publish them to the write queue. You will still be responsible for generating the response packets correctly however you would like.

The reference implementation is written in Ruby and tested in Ruby 2.4.1. It requires the ruby gems daemons, bunny, nokogiri, and nori.

The reference implementation will run as a daemon on your server listening for incoming messages and monitoring an outgoing directory to send your responses. Because RabbitMQ won’t reprocess failed transfers with the same running process and sometimes the listener loses connection to the RabbitMQ queue we recommend having a cron job restart the daemon hourly.

### Installation

1. Install Ruby
   * The amie\_rabbitmq implementation was tested with Ruby 2.4.1. The latest Ruby 2.4.\* should be fine to use. Newer versions of Ruby will also likely work, but aren’t tested.
   * Ruby can be downloaded at <https://www.ruby-lang.org/en/downloads/>
   * For simple Ruby install management installing ruby using rvm <https://rvm.io/> is recommended for installing Ruby.
2. Install Ruby gems.
   * Gems are libraries for Ruby. They are installed using the gem command which is installed along with ruby.
   * Run the commands
     + gem install daemons
     + gem install bunny
     + gem install nokogiri
     + gem install nori
3. Install git.
   * If git is not already installed it should be available in your OS’s package manager
4. Clone the git repo
   * In the directory you would like to install amie\_rabbitmq (ie. ‘cd /usr/local’ to install to /usr/local/amie\_rabbitmq)
   * git clone <https://github.com/ResearchComputing/amie_rabbitmq.git>
5. Configure the software
   * The config file is in config/settings.yml.example. Copy this to settings.yml and edit as necessary. The folders you define for received\_folder and out\_folder are where you will read and write packets respectively.
   * The production host for RabbitMQ is now infopub.xsede.org and a failover server is infopub-alt.xsede.org. The reference implementation does not provide automatic connections to the failover at this time.

*# rabbit mq authentication***message\_queue:  
 host:** infopub.xsede.org  
 **port:** 5671  
 **vhost:** xsede  
  
 **tls:** true

# These are the certificate files corresponding to the one used in the RabbitMQ setup step earlier

*# If using X.509 Authentication (preferred)  
 # filename with full path of crt file* **tls\_cert:** /path/to/host.xsede.org.crt  
  
 *# filename with full path of key file* **tls\_key:** /path/to/host.xsede.org.key  
  
 *# filenames with full path of ca certificate chain* **tls\_ca\_certificates:** - /path/to/ca.crt  
 **auth\_mechanism:** EXTERNAL  
 **verify\_peer:** true  
  
**amie:** *# TGCDB for central AMIE instance or local site name as defined in AMIE for local instance. Instructions for obtaining an AMIE SITE NAME is in the AMIE section of this document.* **local\_site:** ~AMIE SITE NAME~  
 *# TGCDB for local instance or remote site name as defined in AMIE for central instance* **remote\_site:** TGCDB  
 *# name of the rabbitmq exchanges provided by XSEDE as described in the RabbitMQ section of this document.* **write\_exchange:** amie.to.~site~.xsede.org  
 **read\_exchange:** amie.from.~site~.xsede.org  
  
 *# folders for transfer, processing, and logging of AMIE XML packets. Use full path  
 # in\_folder is where you will read incoming packets from*

*# out\_folder is where you will write your response packets*

*# packets that can’t be sent are moved into the respective failed folders*

*# when outgoing packets are sent they are moved to the wait folder, when XSEDE has*

*# responded they are moved to the done folder*

**in\_folder:** /path/rabbitmq\_in  
 **received\_folder:** /path/received  
 **out\_folder:** /path/outbox  
 **done\_folder:** /path/done  
 **wait\_folder:** /path/wait  
 **out\_failed\_folder:** /path/out\_failed  
 **in\_failed\_folder:** /path/in\_failed  
 **log\_folder:** /path/logs  
  
*# List of all valid packet types. To prioritize a packet type give a higher number from 1 to 10. Priority is not turned on in the RabbitMQ server, so just leave this section alone.***valid\_packet\_types:  
 data\_account\_create:** 1  
 **data\_project\_create:** 1  
 **notify\_account\_create:** 1  
 **notify\_account\_inactivate:** 1  
 **notify\_account\_reactivate:** 1  
 **notify\_person\_duplicate:** 1  
 **notify\_person\_ids:** 1  
 **notify\_project\_create:** 1  
 **notify\_project\_inactivate:** 1  
 **notify\_project\_modify:** 1  
 **notify\_project\_reactivate:** 1  
 **notify\_project\_resources:** 1  
 **notify\_project\_usage:** 1  
 **notify\_user\_create:** 1  
 **notify\_user\_modify:** 1  
 **notify\_user\_reactivate:** 1  
 **notify\_user\_suspend:** 1  
 **request\_account\_create:** 1  
 **request\_account\_inactivate:** 1  
 **request\_account\_reactivate:** 1  
 **request\_person\_merge:** 1  
 **request\_project\_create:** 1  
 **request\_project\_inactivate:** 1  
 **request\_project\_modify:** 1  
 **request\_project\_reactivate:** 1  
 **request\_project\_resources:** 1  
 **request\_user\_create:** 1  
 **request\_user\_modify:** 1  
 **request\_user\_reactivate:** 1  
 **request\_user\_suspend:** 1  
 **inform\_transaction\_complete:** 1  
 **response:** 1

* + You also need to edit the rabbit\_transfer\_control.rb script with the path to the rabbit\_transfer.rb script.

1. Run the software
   * When the configuration is complete and ruby and the required ruby gems are installed you can run the script like:
   * ruby rabbit\_transfer.rb –c config/settings.yml
2. Setup Cron to restart the software regularly. Something similar to the below will take of this.

0 \* \* \* \* /path/to/ruby /path/to/app/rabbit\_transfer\_control.rb restart -- -c /path/to/app/conf/settings.yml

* + That should all be one line. The double dash tells the control script to pass the following parameters to the rabbit\_transfer.rb script.

### Existing AMIE users

If your site has already implemented AMIE processing using the existing AMIE reference implementation you do not need to redo that work. You can change the transfer mechanism of AMIE to RabbitMQ and use the amie\_rabbitmq software to handle the transfers

1. Stop your current AMIE implementation from running if it is currently running
2. Update to the latest version of the AMIE reference implementation at <https://software.xsede.org/production/amie/amie.tar>
   * Documentation for the AMIE reference implementation is at <https://software.xsede.org/production/amie/amie-docs.tar>
     + Updating AMIE is simply replacing the part installed to /amie in the AMIE installation documentation.
3. In the amie.config file (standard location is ~amieuser/amielocal/conf/amie.config)
   * Change the “send.~SITE~.TGCDB ssh…” line to “send.~SITE~.TGCDB rabbitmq
4. Install amie\_rabbitmq as explained above
5. In the settings.yml file point the directory parameters to the corresponding AMIE reference implementation folders
   * If your folders follow the AMIE reference implementation instructions the config should be like:

in\_folder: /home/amieuser/amielocal/xfer/incoming/rabbitmq\_in

received\_folder: /home/amieuser/amielocal/xfer/incoming/received

out\_folder: /home/amieuser/amielocal/xfer/outgoing/outbox

done\_folder: /home/amieuser/amielocal/xfer/outgoing/done

wait\_folder: /home/amieuser/amielocal/xfer/outgoing/wait

out\_failed\_folder: /home/amieuser/amielocal/xfer/outgoing/failed

in\_failed\_folder: /home/amieuser/amielocal/xfer/incoming/failed

log\_folder: /home/amieuser/amielocal/logs

* Send mail to help@xsede indicating you are changing your AMIE installation to use RabbitMQ
* When you hear back that XSEDE is ready you can resume running AMIE as you have in the past.

# AMIE Transactions Implementation

AMIE implements many transaction types. For getting started with XSEDE Campus SSO integration only two of these transactions are needed initially. Project Create and Account Create. These transactions are described in the detail in this document. Other transactions are detailed in the full set of AMIE documentation available at <https://software.xsede.org/production/amie/amie-docs.tar>

You need to implement the packet handling process to work at your site creating users and projects as described in this section. You will read incoming packets from the in\_folder and write your responses to the out\_folder as defined in the configuration file from the software install step if you are using the reference installation.

## Project Create transaction

Project create transactions begin with a project request created through the XSEDE Portal and approved by an XSEDE admin. This process is beyond the scope of this document.

After the project has been approved by an XSEDE admin a Request Project Create message will be sent to your site in the amie.to.site\_name.xsede.org queue.

The Project Create transaction has a 4 packet sequence:

1. Request Project Create
   1. Sent from XSEDE to your site.
   2. Provides data about the project and user account information about the PI / Project Lead.
   3. [Request Project Create Definition Appendix](#Request Project Create definition)
2. Notify Project Create
   1. Sent from your site back to XSEDE
   2. Contains data about how the project will be identified locally as well as local user account info for the PI / Project lead so that the XSEDE Portal user account can be linked to your local user account for the person.
   3. [Notify Project Create Definition Appendix](#Notify Project Create Definition)
3. Data Project Create
   1. Sent from XSEDE to your site
   2. Verifies that the Notify Project Create packet was accepted and processed and provides the current list of DN’s for the Project Lead user account
   3. [Data Project Create Definition Appendix](#Data Project Create Definition)
4. Inform Transaction Complete
   1. Sent from your site to XSEDE
   2. Finalizes the Project Create transaction
   3. [Inform Transaction Complete Definition Appendix](#Inform Transaction Complete Definition)

### Example Project Create Transaction

#### Request Project Create

* The X-PORTAL site\_person\_id will be the XSEDE username used to log into your site from the SSO hub.
* If person\_id is included in this packet then your local user is already mapped to the XSEDE user and the notify response must have the same person\_id

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <request\_project\_create>

    <header>

      <date>2017-08-01T14:13:07Z</date>

      <expected\_reply\_list>

        <expected\_reply>

          <timeout>30240</timeout>

          <type>notify\_project\_create</type>

        </expected\_reply>

      </expected\_reply\_list>

      <from\_site\_name>TGCDB</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>1</packet\_id>

      <to\_site\_name>PSC</to\_site\_name>

      <transaction\_id>158194</transaction\_id>

    </header>

    <body>

      <abstract>The development of high-...</abstract>

      <academic\_degree\_list>

        <academic\_degree>

          <degree>PhD</degree>

          <field>Computer Science</field>

        </academic\_degree>

        <academic\_degree>

          <degree>PhD</degree>

          <field>Physics and Computer Science</field>

        </academic\_degree>

      </academic\_degree\_list>

      <alloc\_resource>bridges.psc.xsede</alloc\_resource>

      <alloc\_type>new</alloc\_type>

      <charge\_num>TG-MCB090174</charge\_num>

      <end\_date>2017-12-31</end\_date>

      <grant\_num>MCB090174</grant\_num>

      <nsf\_status\_code>F</nsf\_status\_code>

      <pfos>

        <number>340</number>

      </pfos>

      <pi>

        <dn\_list>

          <dn>/C=US/… Jha</dn>

          <dn>/C=US/…/CN=Shantenu Jha</dn>

        </dn\_list>

        <personal\_info>

          <address>

            <city>Piscataway</city>

            <country>9US</country>

            <state>NJ</state>

            <str\_address>705 CoRE Building</str\_address>

            <str\_address2>Busch Campus</str\_address2>

            <zip>08854-8058</zip>

          </address>

          <business\_phone>

            <number>555-555-5555</number>

          </business\_phone>

          <email>sample@sample.edu</email>

          <first\_name>Shantenu</first\_name>

          <global\_id>600</global\_id>

          <last\_name>Jha</last\_name>

          <org\_code>0087718</org\_code>

          <organization>Rutgers</organization>

          <person\_id>SJHA1234P</person\_id>

        </personal\_info>

        <req\_login\_list>

          <req\_login>sjha</req\_login>

          <req\_login>jha</req\_login>

          <req\_login>ux452534</req\_login>

          <req\_login>shantenu</req\_login>

          <req\_login></req\_login>

        </req\_login\_list>

      </pi>

      <project\_id>mc3bggp</project\_id>

      <project\_title>A RADICAL Use of XSEDE … Resources</project\_title>

      <proposal\_num>MCB090174</proposal\_num>

      <record\_id>XRAS-48241-bridges.psc.xsede</record\_id>

      <resource\_list>

        <resource>bridges.psc.xsede</resource>

      </resource\_list>

      <role\_list>

        <role>allocation\_manager</role>

      </role\_list>

      <sfos\_list>

        <sfos>

          <number>0</number>

        </sfos>

      </sfos\_list>

      <site\_person\_id\_list>

        <site\_person\_id>

          <person\_id>1000522</person\_id>

          <site>Stanford</site>

        </site\_person\_id>

        <site\_person\_id>

          <person\_id>jha</person\_id>

          <site>X-PORTAL</site>

        </site\_person\_id>

      </site\_person\_id\_list>

      <start\_date>2017-01-01</start\_date>

      <su\_alloc>-250000</su\_alloc>

    </body>

  </request\_project\_create>

</amie>

#### Notify Project Create

* After you receive a Request Project Create packet you should respond with a Notify Project Create. This is the packet where you tell XSEDE what the username and person id for the user are on resources at your site.
* The header section must be included with an updated date, the transaction id from the request project create packet and the from\_site\_name updated with your site’s AMIE site name. The other fields should be duplicated as shown below.
* You must provide the project\_id as a local id for the project and a person\_id as a local identifier for the person. These IDs can be whatever you would like to use for a distinct ID from your site.
* You must also provide a remote\_site\_login which is the username on your local resource. This will allow XSEDE to map the XSEDE Portal username to the user’s local username.
* Other required fields can be copied from the request packet.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <notify\_project\_create>

    <header>

      <date>2017-08-01T17:00:51Z</date>

      <expected\_reply\_list>

        <expected\_reply>

          <timeout>1440</timeout>

          <type>data\_project\_create</type>

        </expected\_reply>

      </expected\_reply\_list>

      <from\_site\_name>PSC</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>1</packet\_id>

      <to\_site\_name>TGCDB</to\_site\_name>

      <transaction\_id>158194</transaction\_id>

    </header>

    <body>

      <end\_date>2017-12-31</end\_date>

      <grant\_num>MCB090174</grant\_num>

      <pfos>

        <number>340</number>

      </pfos>

      <pi>

        <dn\_list>

          <dn>/C=US/O=Pittsburgh … /CN=Shantenu Jha</dn>

          <dn>/C=US/O=National … Jha</dn>

        </dn\_list>

        <personal\_info>

          <email>sample@sample.edu</email>

          <first\_name>Shantenu</first\_name>

          <last\_name>Jha</last\_name>

          <org\_code>0087718</org\_code>

          <organization>Rutgers</organization>

          <person\_id>SJHA1234P</person\_id>

          <uid>18900</uid>

        </personal\_info>

        <remote\_site\_login>sjha</remote\_site\_login>

      </pi>

      <project\_gid>14231</project\_gid>

      <project\_id>mc3bggp</project\_id>

      <project\_title>A RADICAL … XSEDE Resources</project\_title>

      <resource\_list>

        <resource>bridges.psc.xsede</resource>

      </resource\_list>

      <start\_date>2017-01-01</start\_date>

      <su\_alloc>-250000</su\_alloc>

    </body>

  </notify\_project\_create>

</amie>

#### Data Project Create

* This packet confirms what you send in the notify packet and provides the current list of DNs for the Project Lead user from the XSEDE Database.
* If the DN list hasn’t changed since the Request Project Create Packet there isn’t really anything you need to do here except send the Inform Transaction Complete Packet.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <data\_project\_create>

    <header>

      <date>2017-08-01T17:13:15Z</date>

      <expected\_reply\_list>

        <expected\_reply>

          <timeout>30240</timeout>

          <type>inform\_transaction\_complete</type>

        </expected\_reply>

      </expected\_reply\_list>

      <from\_site\_name>TGCDB</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>2</packet\_id>

      <to\_site\_name>PSC</to\_site\_name>

      <transaction\_id>158194</transaction\_id>

    </header>

    <body>

      <dn\_list>

        <dn>/C=US/O=National Center … /CN=Shantenu Jha</dn>

        <dn>/C=US/O=Pittsburgh Supercomputing Center/CN=Shantenu Jha</dn>

      </dn\_list>

      <global\_id>600</global\_id>

      <person\_id>SJHA1234P</person\_id>

      <project\_id>mc3bggp</project\_id>

    </body>

  </data\_project\_create>

</amie>

#### Inform Transaction Complete

1. The header section must be included with an updated date, the transaction id from the prior packets in the transaction, your AMIE site name as to\_site\_name and the other fields copied as below. The body can be copied exactly.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <inform\_transaction\_complete>

    <header>

      <date>2017-08-01T17:40:03Z</date>

      <from\_site\_name>PSC</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>2</packet\_id>

      <to\_site\_name>TGCDB</to\_site\_name>

      <transaction\_id>158194</transaction\_id>

    </header>

    <body>

      <detail\_code>1</detail\_code>

      <message>Success</message>

      <status\_code>Success</status\_code>

    </body>

  </inform\_transaction\_complete>

</amie>

## Account Create Transaction

The account create transaction is very similar to the project create transaction except without the project detail. It is used to add additional user accounts to a project.

The Account Create transaction has a 4 packet sequence:

1. Request Account Create
   1. Sent from XSEDE to your site.
   2. Provides data about the user account.
   3. [Request Account Create Definition Appendix](#Request Account Create Definition)
2. Notify Account Create
   1. Sent from your site back to XSEDE
   2. Contains local user account info so that the XSEDE Portal user account can be linked to your local user account for the person.
   3. [Notify Account Create Definition Appendix](#Notify Account Create Definition)
3. Data Account Create
   1. Sent from XSEDE to your site
   2. Verifies that the Notify Account Create packet was accepted and processed and provides the current list of DN’s for the user account
   3. [Data Account Create Definition Appendix](#Data Account Create Definition)
4. Inform Transaction Complete
   1. Sent from your site to XSEDE
   2. Finalizes the Account Create transaction
   3. [Inform Transaction Complete Definition Appendix](#Inform Transaction Complete Definition)

### Example Account Create Transaction

#### Request Account Create

1. Provides information for a user account to be added to a project
2. If the person\_id is included in this packet then the user has previously been mapped to XSEDE and this person\_id must be used in the Notify Account Create response.
3. The Site Person ID for site X-PORTAL is the XSEDE Portal username.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <request\_account\_create>

    <header>

      <date>2017-08-01T21:13:07Z</date>

      <expected\_reply\_list>

        <expected\_reply>

          <timeout>30240</timeout>

          <type>notify\_account\_create</type>

        </expected\_reply>

      </expected\_reply\_list>

      <from\_site\_name>TGCDB</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>1</packet\_id>

      <to\_site\_name>PSC</to\_site\_name>

      <transaction\_id>158212</transaction\_id>

    </header>

    <body>

      <academic\_degree\_list>

        <academic\_degree>

          <degree>MS</degree>

          <field>Chemical Engineering</field>

        </academic\_degree>

      </academic\_degree\_list>

      <alloc\_resource>bridges-gpu.psc.xsede</alloc\_resource>

      <grant\_num>MCB130173</grant\_num>

      <nsf\_status\_code>GS</nsf\_status\_code>

      <project\_id>mc4s9op</project\_id>

      <resource\_list>

        <resource>bridges-gpu.psc.xsede</resource>

      </resource\_list>

      <site\_person\_id\_list>

        <site\_person\_id>

          <person\_id>94270</person\_id>

          <site>SDSC</site>

        </site\_person\_id>

        <site\_person\_id>

          <person\_id>HWANGH1AP</person\_id>

          <site>PSC</site>

        </site\_person\_id>

        <site\_person\_id>

          <person\_id>hhwang8</person\_id>

          <site>X-PORTAL</site>

        </site\_person\_id>

      </site\_person\_id\_list>

      <user>

        <dn\_list>

          <dn>/C=US/O=National … hwang</dn>

          <dn>/C=US/O=Pittsburgh Supercomputing Center/CN=hyea hwang</dn>

          <dn>/DC=EDU/DC=UTEXAS/DC=TACC/O=MYPROXY/CN=tg829080</dn>

        </dn\_list>

        <password\_access\_enable>1</password\_access\_enable>

        <personal\_info>

          <address>

            <city>atlanta</city>

            <country>9US</country>

            <state>GA</state>

            <str\_address>2450 Test Data St</str\_address>

            <str\_address2>Apt 1</str\_address2>

            <zip>30324</zip>

          </address>

          <business\_phone>

            <comment></comment>

            <extension></extension>

            <number>555-555-5555</number>

          </business\_phone>

          <dept></dept>

          <email>sample@sample.com</email>

          <first\_name>hyea</first\_name>

          <global\_id>33991</global\_id>

          <last\_name>hwang</last\_name>

          <middle\_name></middle\_name>

          <org\_code>0087239</org\_code>

          <organization>Georgia Institute of Technology</organization>

          <person\_id>HWANGH1AP</person\_id>

          <title></title>

        </personal\_info>

        <req\_login\_list>

          <req\_login></req\_login>

        </req\_login\_list>

      </user>

    </body>

  </request\_account\_create>

</amie>

#### Notify Account Create

* After you receive a Request Account Create packet you should respond with a Notify Account Create. This is the packet where you tell XSEDE what the username and person id for the user are on resources at your site.
* The header section must be included with an updated date, the transaction id from the request account create packet and the from\_site\_name updated with your site’s AMIE site name. The other fields should be duplicated as shown below.
* You must provide the person\_id as a local identifier for the person. This ID can be whatever you would like to use for a distinct ID for the user from your site.
* You must also provide a remote\_site\_login which is the username on your local resource. This will allow XSEDE to map the XSEDE Portal username to the user’s local username.
* Other required fields can be copied from the request packet.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <notify\_account\_create>

    <header>

      <date>2017-08-01T21:40:03Z</date>

      <expected\_reply\_list>

        <expected\_reply>

          <timeout>1440</timeout>

          <type>data\_account\_create</type>

        </expected\_reply>

      </expected\_reply\_list>

      <from\_site\_name>PSC</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>1</packet\_id>

      <to\_site\_name>TGCDB</to\_site\_name>

      <transaction\_id>158212</transaction\_id>

    </header>

    <body>

      <project\_id>mc4s9op</project\_id>

      <resource\_list>

        <resource>bridges-gpu.psc.xsede</resource>

      </resource\_list>

      <user>

        <dn\_list>

          <dn>/C=US/O=National Center for Supercomputing Applications/CN=hyea hwang</dn>

          <dn>/C=US/O=Pittsburgh Supercomputing … =tg829080</dn>

        </dn\_list>

        <personal\_info>

          <email>sample@sample.com</email>

          <first\_name>hyea</first\_name>

          <last\_name>hwang</last\_name>

          <org\_code>0087239</org\_code>

          <organization>Georgia Institute of Technology</organization>

          <person\_id>HWANGH1AP</person\_id>

          <uid>56059</uid>

        </personal\_info>

        <remote\_site\_login>hwangh</remote\_site\_login>

      </user>

    </body>

  </notify\_account\_create>

</amie>

#### Data Account Create

* This packet confirms what you send in the notify packet and provides the current list of DNs for the Project Lead user from the XSEDE Database.
* If the DN list hasn’t changed since the Request Account Create Packet there isn’t really anything you need to do here except send the Inform Transaction Complete Packet.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <data\_account\_create>

    <header>

      <date>2017-08-01T21:43:29Z</date>

      <expected\_reply\_list>

        <expected\_reply>

          <timeout>30240</timeout>

          <type>inform\_transaction\_complete</type>

        </expected\_reply>

      </expected\_reply\_list>

      <from\_site\_name>TGCDB</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>2</packet\_id>

      <to\_site\_name>PSC</to\_site\_name>

      <transaction\_id>158212</transaction\_id>

    </header>

    <body>

      <dn\_list>

        <dn>/C=US/O=National … hwang</dn>

        <dn>/C=US/O=Pittsburgh Supercomputing Center/CN=hyea hwang</dn>

        <dn>/DC=EDU/DC=UTEXAS/DC=TACC/O=MYPROXY/CN=tg829080</dn>

      </dn\_list>

      <global\_id>33991</global\_id>

      <person\_id>HWANGH1AP</person\_id>

      <project\_id>mc4s9op</project\_id>

    </body>

  </data\_account\_create>

</amie>

#### Inform Transaction Complete

1. The header section must be included with an updated date, the transaction id from the prior packets in the transaction, your AMIE site name as to\_site\_name and the other fields copied as below. The body can be copied exactly.

<?xml version="1.0" encoding="UTF-8"?>

<amie version="1.0">

  <inform\_transaction\_complete>

    <header>

      <date>2017-08-01T17:40:03Z</date>

      <from\_site\_name>PSC</from\_site\_name>

      <originating\_site\_name>TGCDB</originating\_site\_name>

      <packet\_id>2</packet\_id>

      <to\_site\_name>TGCDB</to\_site\_name>

      <transaction\_id>158194</transaction\_id>

    </header>

    <body>

      <detail\_code>1</detail\_code>

      <message>Success</message>

      <status\_code>Success</status\_code>

    </body>

  </inform\_transaction\_complete>

</amie>

# Error Handling

Data Problems

In the early going it is likely you’ll have some issues in your response packets. Usually the problem is missing required fields. As time goes on sometimes maybe you’ll try to use the same person\_id for a user who has created a duplicate XSEDE account. When you send a packet that gets rejected by the server you will get a failure message back.

When you get a failure message you can correct the problematic data in your packet and just resend it and if everything is valid now the transaction will proceed.

Timeouts

Packet headers have a timeout value for the expected reply. This only serves to send an email to XSEDE AMIE Admins if a response hasn’t been received yet so that we can check that there isn’t a system issue. You can still respond to the packet after the timeout period has elapsed.

# RabbitMQ

Two RabbitMQ queues will be used for the data transfer. The queue for data from XSEDE to your site is amie.to.site\_name.xsede.org. The queue for data from your site to XSEDE is amie.from.site\_name.xsede.org.

### RabbitMQ Message Format

The content of the messages is given by the example data for the packets provided above. You also need to provide header data with the message length. A sample publish in Ruby:

write\_exchange.publish(content,headers: {message\_length: content.length}, :priority *=>* priority, :routing\_key *=>* config['amie']['write\_exchange'] + '.queue')  
success = ch.wait\_for\_confirms

Priority can be included as a number from 1 to 10, but is not yet implemented on the central AMIE server. Your write exchange is amie.from.site.xsede.org and the queue is amie.from.site.xsede.org.queue.

# Appendix 1: Packet Definitions

Note that only the bold tags are required. Tags with multiple fields where tags other than the first tag are bold are only required if the first tag is included.

This document only shows the definitions for packet types you are likely to need to implement for Campus SSO integration. If you find a need to implement more packet types or want to know more about AMIE you can download the full AMIE documentation set at <https://software.xsede.org/production/amie/amie-docs.tar>

## Request Project Create definition

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |
| Abstract | abstract |

|  |  |
| --- | --- |
| AcademicDegree,**Degree**,0  AcademicDegree,**Field**,0  AcademicDegree,**Degree**,1  AcademicDegree,**Field**,1  … | academic\_degree\_list/  academic\_degree/degree  academic\_degree/field |

|  |  |
| --- | --- |
| AllocatedResource | alloc\_resource |
| AllocationType | alloc\_type |
| Applications | applications |
| Background | background |
| ChargeNumber | charge\_num |
| CitizenshipList,0  CitizenshipList,1 | citizenship/country |
| Comment | comment |
| Deliverables | deliverables |
| DiskSpace | diskspace |
| EndDate[[1]](#footnote-1) | end\_date |
| Facilities | facilities |
| GrantType | grant\_type |
| GrantNumber | grant\_num |
| Justification | justification |
| Languages | languages |
| Memory | memory |
| Methodologies | methodologies |
| Milestones | milestones |
| NsfStatusCode | nsf\_status\_code |
| OtherResources | other\_resources |
| PfosAbbreviation | pfos/abbr |
| PfosDescription | pfos/description |
| **PfosNumber** | pfos/number |
| PiBusinessPhoneComment | pi/personal\_info/business\_phone/comment |
| PiBusinessPhoneExtension | pi/personal\_info/business\_phone/extension |
| PiBusinessPhoneNumber[[2]](#footnote-2) | pi/personal\_info/business\_phone/number |
| PiCitizenship | pi/personal\_info/citizenship |
| PiCity | pi/personal\_info/city |
| PiCountry | pi/personal\_info/country |
| PiCountryOfAccess | pi/personal\_info/country\_access |
| PiDepartment | pi/personal\_info/dept |
| PiDnList, 0  PiDnList, 1  … | pi/dn\_list/dn |
| PiEmail | pi/personal\_info/email |
| PiEmpCode | pi/personal\_info/emp\_code |
| PiFax | pi/personal\_info/fax |
| PiFirstName | pi/personal\_info/first\_name |
| PiGlobalID | pi/personal\_info/global\_id |
| PiHomePhoneComment | pi/personal\_info/home\_phone/comment |
| PiHomePhoneExtension | pi/personal\_info/home\_phone/extension |
| PiHomePhoneNumber[[3]](#footnote-3) | pi/personal\_info/home\_phone/number |
| PiLastName | pi/personal\_info/last\_name |
| PiMiddleName | pi/personal\_info/middle\_name |
| PiOfficeAddress | pi/personal\_info/address/off\_address |
| PiOrganization | pi/personal\_info/organization |
| PiOrgCode | pi/personal\_info/org\_code |
| PiPersonID | pi/personal\_info/person\_id |
| PiPosition | pi/personal\_info/position |
| PiRemoteSiteID | pi/remote\_site\_id |
| PiRemoteSiteLogin | pi/remote\_site\_login |
| PiRequestedLoginList, 0  PiRequestedLoginList, 1  … | pi/req\_login\_list/req\_login |
| PiRequesterLogin | pi/requester\_login |
| PiState | pi/personal\_info/address/state |
| PiStreetAddress | pi/personal\_info/address/str\_address |
| PiStreetAddress2 | pi/personal\_info/address/str\_address2 |
| PiTitle | pi/personal\_info/title |
| PiZip | pi/personal\_info/address/zip |
| Processors | processors |
| Progress | progress |
| ProjectID | project\_id |
| **ProjectTitle** | project\_title |
| ProposalNumber | proposal\_num |
| Qualifications | qualifications |
| **ResourceList**, 0  ResourceList, 1  … | resource\_list/resource |
| RoleList, 0  RoleList, 1  … | role\_list/role |
| Sector | sector |
| **ServiceUnitsAllocated** | su\_alloc |
| Sfos, Abbreviation, 0  Sfos, Description, 0  Sfos[[4]](#footnote-4), **Number**, 0  Sfos, Abbreviation, 1  Sfos, Description, 1  Sfos, **Number**, 1  … | sfos\_list/  sfos/abbr  sfos/description  sfos/number |
| SitePersonId, **Site**, 0  SitePersonId, **PersonID**, 0  SitePersonId, **Site**, 1  SitePersonId, **PersonID**, 1  … | site\_person\_id\_list/  site\_person\_id/site  site\_person\_id/person\_id |
| **StartDate**[[5]](#footnote-5) | start\_date |
| StatementOfWork | statement\_work |
| Support | support |

## Notify Project Create Definition

ResourceLogin is required for every resource. All required fields except ProjectID, PiPersonID and ResourceLogin can simply be copied from the Request Project Create packet. The Resource Login should be the username for the user locally on your resource. The PersonID is a unique identifier for the user at your site and can be whatever you would like. PersonID could potentially be the same as the Login if users are limited to one login username.

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |
| Abstract | abstract |

|  |  |
| --- | --- |
| AcademicDegree,**Degree**,0  AcademicDegree,**Field**,0  AcademicDegree,**Degree**,1  AcademicDegree,**Field**,1  … | academic\_degree\_list/  academic\_degree/degree  academic\_degree/field |

|  |  |
| --- | --- |
| AccountActivityTime[[6]](#footnote-6) | account\_activity\_time |
| AllocationType | alloc\_type |
| Applications | applications |
| Background | background |
| BoardType | board\_type |
| Comment | comment |
| Deliverables | deliverables |
| DiskSpace | diskspace |
| EndDate[[7]](#footnote-7) | end\_date |
| Facilities | facilities |
| GrantType | grant\_type |
| **GrantNumber** | grant\_num |
| Justification | justification |
| Languages | languages |
| Memory | memory |
| Methodologies | methodologies |
| Milestones | milestones |
| NsfStatusCode | nsf\_status\_code |
| OtherResources | other\_resources |
| PfosAbbreviation | pfos/abbr |
| PfosDescription | pfos/description |
| **PfosNumber** | pfos/number |
| PiBusinessPhoneComment | pi/personal\_info/business\_phone/comment |
| PiBusinessPhoneExtension | pi/personal\_info/business\_phone/extension |
| PiBusinessPhoneNumber[[8]](#footnote-8) | pi/personal\_info/business\_phone/number |
| PiCitizenship | pi/personal\_info/citizenship |
| PiCity | pi/personal\_info/city |
| PiCountry | pi/personal\_info/country |
| PiCountryOfAccess | pi/personal\_info/country\_access |
| PiDepartment | pi/personal\_info/dept |
| PiDnList, 0  PiDnList, 1  … | pi/dn\_list/dn |
| PiEmail | pi/personal\_info/email |
| PiEmpCode | pi/personal\_info/emp\_code |
| PiFax | pi/personal\_info/fax |
| PiFirstName | pi/personal\_info/first\_name |
| PiGlobalID | pi/personal\_info/global\_id |
| PiHomePhoneComment | pi/personal\_info/home\_phone/comment |
| PiHomePhoneExtension | pi/personal\_info/home\_phone/extension |
| PiHomePhoneNumber[[9]](#footnote-9) | pi/personal\_info/home\_phone/number |
| PiLastName | pi/personal\_info/last\_name |
| PiMiddleName | pi/personal\_info/middle\_name |
| PiNotifierLogin | pi/notifier\_login |
| PiOfficeAddress | pi/personal\_info/address/off\_address |
| **PiOrganization** | pi/personal\_info/organization |
| **PiOrgCode** | pi/personal\_info/org\_code |
| **PiPersonID** | pi/personal\_info/person\_id |
| PiPosition | pi/personal\_info/position |
| PiRemoteSiteLogin | pi/remote\_site\_login |
| PiRequestedLoginList, 0  PiRequestedLoginList, 1  … | pi/req\_login\_list/req\_login |
| PiState | pi/personal\_info/address/state |
| PiStreetAddress | pi/personal\_info/address/str\_address |
| PiStreetAddress2 | pi/personal\_info/address/str\_address2 |
| PiTitle | pi/personal\_info/title |
| PiZip | pi/personal\_info/address/zip |
| Processors | processors |
| Progress | progress |
| **ProjectID** | project\_id |
| **ProjectTitle** | project\_title |
| ProposalNumber | proposal\_num |
| Qualifications | qualifications |
| RecordID | record\_id |
| **ResourceList**, 0  ResourceList, 1  … | resource\_list/resource |
| ResourceLogin, **Resource**, 0  ResourceLogin, **Login**, 0  ResourceLogin, **Resource**, 1  ResourceLogin, **Login**, 1  … | resource\_login \_list/  resource\_login/resource  resource\_login/login |
| RoleList, 0  RoleList, 1  … | role\_list/role |
| Sector | sector |
| **ServiceUnitsAllocated** | su\_alloc |
| Sfos, Abbreviation, 0  Sfos, Description, 0  Sfos[[10]](#footnote-10), **Number**, 0  Sfos, Abbreviation, 1  Sfos, Description, 1  Sfos, **Number**, 1  … | sfos\_list/  sfos/abbr  sfos/description  sfos/number |
| **StartDate**[[11]](#footnote-11) | start\_date |
| StatementOfWork | statement\_work |
| Support | support |

## Data Project Create Definition

This is a confirmation that the project has been created and the project and user mapping is confirmed. The DN list may be updated from the Request Project Create so check for updates if your site is concerned with the DNs

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |
| Comment | comment |
| DnList, 0  DnList, 1  … | dn\_list/dn |
| PersonID | person\_id |
| ProjectID | project\_id |

## Inform Transaction Complete Definition

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |
| **DetailCode** | detail\_code |
| **Message** | message |
| **StatusCode** | status\_code |

## Request Account Create Definition

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |

|  |  |
| --- | --- |
| AcademicDegree,**Degree**,0  AcademicDegree,**Field**,0  AcademicDegree,**Degree**,1  AcademicDegree,**Field**,1  … | academic\_degree\_list/  academic\_degree/degree  academic\_degree/field |

|  |  |
| --- | --- |
| CitizenshipList,0  CitizenshipList,1 | citizenship/country |
| Comment | comment |
| GrantNumber | grant\_num |
| NsfStatusCode | nsf\_status\_code |
| **ProjectID** | project\_id |
| **ResourceList**, 0  ResourceList, 1  … | resource\_list/resource |
| RoleList, 0  RoleList, 1  … | role\_list/role |
| SitePersonId, **Site**, 0  SitePersonId, **PersonID**, 0  SitePersonId, **Site**, 1  SitePersonId, **PersonID**, 1  … | site\_person\_id\_list/  site\_person\_id/site  site\_person\_id/person\_id |
| UserBusinessPhoneComment | user/personal\_info/business\_phone/comment |
| UserBusinessPhoneExtension | user/personal\_info/business\_phone/extension |
| UserBusinessPhoneNumber[[12]](#footnote-12) | user/personal\_info/business\_phone/number |
| UserCitizenship | user/personal\_info/citizenship |
| UserCity | user/personal\_info/city |
| UserCountry | user/personal\_info/country |
| UserCountryOfAccess | user/personal\_info/country\_access |
| UserDepartment | user/personal\_info/dept |
| UserDnList, 0  UserDnList, 1  … | user/dn\_list/dn |
| UserEmpCode | user/personal\_info/emp\_code |
| UserEmail | user/personal\_info/email |
| UserFax | user/personal\_info/fax |
| UserFirstName | user/personal\_info/first\_name |
| UserGlobalID | user/personal\_info/global\_id |
| UserHomePhoneComment | user/personal\_info/home\_phone/comment |
| UserHomePhoneExtension | user/personal\_info/home\_phone/extension |
| UserHomePhoneNumber[[13]](#footnote-13) | user/personal\_info/home\_phone/nu |
| UserLastName | user/personal\_info/last\_name |
| UserMiddleName | user/personal\_info/middle\_name |
| UserOfficeAddress | user/personal\_info/address/off\_address |
| UserOrganization | user/personal\_info/organization |
| UserOrgCode | user/personal\_info/org\_code |
| UserPasswordAccessEnable[[14]](#footnote-14) | user/password\_access\_enable |
| UserPersonID | user/personal\_info/person\_id |
| UserPosition | user/personal\_info/position |
| UserRemoteSiteID | user/remote\_site\_id |
| UserRemoteSiteLogin | user/remote\_site\_login |
| UserRequestedLoginList, 0  UserRequestedLoginList, 1  … | user/req\_login\_list/req\_login |
| UserRequesterLogin | user/requester\_login |
| UserRole | user/role |
| UserState | user/personal\_info/address/state |
| UserStreetAddress | user/personal\_info/address/str\_address |
| UserStreetAddress2 | user/personal\_info/address/str\_address2 |
| UserTitle | user/personal\_info/title |
| UserZip | user/personal\_info/address/zip |

## Notify Account Create Definition

ResourceLogin is required for each ResourceList, UserPersonId is just like PiPersonId from the Project Create transaction

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |
| AcademicDegree,**Degree**,0  AcademicDegree,**Field**,0  AcademicDegree,**Degree**,1  AcademicDegree,**Field**,1  … | academic\_degree\_list/  academic\_degree/degree  academic\_degree/field |
| AccountActivityTime[[15]](#footnote-15) | account\_activity\_time |
| Comment | comment |
| NsfStatusCode | nsf\_status\_code |
| **ProjectID** | project\_id |
| **ResourceList**, 0  ResourceList, 1  … | resource\_list/resource |
| RoleList, 0  RoleList, 1  … | role\_list/role |
| ResourceLogin, **Resource**, 0  ResourceLogin, **Login**, 0  ResourceLogin, **Resource**, 1  ResourceLogin, **Login**, 1  … | resource\_login \_list/  resource\_login/resource  resource\_login/login |
| StartDate | start\_date |
| UserBusinessPhoneComment | user/personal\_info/business\_phone/comment |
| UserBusinessPhoneExtension | user/personal\_info/business\_phone/extension |
| UserBusinessPhoneNumber[[16]](#footnote-16) | user/personal\_info/business\_phone/number |
| UserCitizenship | user/personal\_info/citizenship |
| UserCity | user/personal\_info/city |
| UserCountry | user/personal\_info/country |
| UserCountryOfAccess | user/personal\_info/country\_access |
| UserDepartment | user/personal\_info/dept |
| UserDnList, 0  UserDnList, 1  … | user/dn\_list/dn |
| UserEmpCode | user/personal\_info/emp\_code |
| UserEmail | user/personal\_info/email |
| UserFax | user/personal\_info/fax |
| UserFirstName | user/personal\_info/first\_name |
| UserGlobalID | user/personal\_info/global\_id |
| UserHomePhoneComment | user/personal\_info/home\_phone/comment |
| UserHomePhoneExtension | user/personal\_info/home\_phone/extension |
| UserHomePhoneNumber[[17]](#footnote-17) | user/personal\_info/home\_phone/number |
| UserLastName | user/personal\_info/last\_name |
| UserMiddleName | user/personal\_info/middle\_name |
| UserNotifierLogin | user/notifier\_login |
| UserOfficeAddress | user/personal\_info/address/off\_address |
| UserOrganization | user/personal\_info/organization |
| UserOrgCode | user/personal\_info/org\_code |
| UserPasswordAccessEnable[[18]](#footnote-18) | user/password\_access\_enable |
| **UserPersonID** | user/personal\_info/person\_id |
| UserPosition | user/personal\_info/position |
| UserRemoteSiteLogin | user/remote\_site\_login |
| UserRequestedLoginList, 0  UserRequestedLoginList, 1  … | user/req\_login\_list/req\_login |
| UserRole | user/role |
| UserState | user/personal\_info/address/state |
| UserStreetAddress | user/personal\_info/address/str\_address |
| UserStreetAddress2 | user/personal\_info/address/str\_address2 |
| UserTitle | user/personal\_info/title |
| UserZip | user/personal\_info/address/zip |

## Data Account Create Definition

|  |  |
| --- | --- |
| ***tag …*** | ***path*** |
| Comment | comment |
| DnList, 0  DnList, 1  … | dn\_list/dn |
| PersonID | person\_id |
| ProjectID | project\_id |

1. XML date format yyyy-mm-dd [↑](#footnote-ref-1)
2. PiBusinessPhoneNumber is required if PiBusinessPhoneComment or PiBusinessPhoneExtension is present. [↑](#footnote-ref-2)
3. PiHomePhoneNumber is required if PiHomePhoneComment or PiHomePhoneExtension is present. [↑](#footnote-ref-3)
4. Sfos is not required. However, if an Sfos is specified, then the Number is required. [↑](#footnote-ref-4)
5. XML date format yyyy-mm-dd [↑](#footnote-ref-5)
6. XML datetime format yyyy-mm-dd**T**hh:mm:ss**-**hh:mm or yyyy-mm-dd**T**hh:mm:ss**Z** [↑](#footnote-ref-6)
7. XML date format yyyy-mm-dd [↑](#footnote-ref-7)
8. PiBusinessPhoneNumber is required if PiBusinessPhoneComment or PiBusinessPhoneExtension is present. [↑](#footnote-ref-8)
9. PiHomePhoneNumber is required if PiHomePhoneComment or PiHomePhoneExtension is present. [↑](#footnote-ref-9)
10. Sfos is not required. However, if an Sfos is specified, then the Number is required. [↑](#footnote-ref-10)
11. XML date format yyyy-mm-dd [↑](#footnote-ref-11)
12. UserBusinessPhoneNumber is required if UserBusinessPhoneComment or UserBusinessPhoneExtension is present. [↑](#footnote-ref-12)
13. UserHomePhoneNumber is required if UserHomePhoneComment or UserHomePhoneExtension is present. [↑](#footnote-ref-13)
14. XML Boolean – value should be one(1) for true, zero(0) for false. [↑](#footnote-ref-14)
15. XML datetime format yyyy-mm-dd**T**hh:mm:ss**-**hh:mm or yyyy-mm-dd**T**hh:mm:ss**Z** [↑](#footnote-ref-15)
16. UserBusinessPhoneNumber is required if UserBusinessPhoneComment or UserBusinessPhoneExtension is present. [↑](#footnote-ref-16)
17. UserHomePhoneNumber is required if UserHomePhoneComment or UserHomePhoneExtension is present. [↑](#footnote-ref-17)
18. XML Boolean – value should be one(1) for true, zero(0) for false. [↑](#footnote-ref-18)