Advance Web Integration Reference

March 2010

Release 9.5.0.0

SUNGARD HIGHER EDUCATION

What can we help you achieve?

Trademark, Publishing Statement and Copyright Notice

In preparing and providing this publication, SunGard Higher Education is not rendering legal, accounting, or other similar professional services. SunGard Higher Education makes no claims that an institution's use of this publication or the software for which it is provided will insure compliance with applicable federal or state laws, rules, or regulations. Each organization should seek legal, accounting and other similar professional services from competent providers of the organization’s own choosing.

Prepared by: SunGard Higher Education

4 Country View Road
Malvern, Pennsylvania 19355
United States of America
(800) 522 - 4827

Customer Support Center Website

http://connect.sungardhe.com

Documentation Feedback

http://education.sungardhe.com/survey/documentation.html

Distribution Services E-mail Address

distserv@sungardhe.com

Revision History Log

Publication Date	Summary
March 2010	Updated version that supports Advance 9.5.0.0.

Think Before You Print.

Contents

Table of Figures

Introduction

This document presents information for the integration of products with Advance Web that provide additional functionality for institutions and organizations. In each of the product descriptions, the version of initial introduction is identified, along with any additional licensing requirements. Integration of some of these products may require additional licensing from a vendor other than SunGard Higher Education.

Intended Audience

The intended audience of this document is system administrators and anyone responsible for the installation, maintenance and use of these products with Advance Web.

Assumptions about this Guide

This guide is not all-inclusive. This guide covers all the main topics you may want to perform, but it doesn't tell you all of the various ways you can perform a particular task, nor does it tell you all of the possible uses for each utility. This guide chooses one way – usually the most commonly used and most effective way – and steps you through how to perform the task quickly and effectively.

Security Assumptions in this Guide

This guide presents information about applications and forms that can be accessed by a user with viewing and maintaining rights. Because a wide variety of specific security settings can be assigned to users, this guide does not cover each of those settings in detail. Generally speaking, when a user accesses a form and they have viewing-only rights, the New and/or Save menu items will not display on the form header. If a user does not have viewing rights for an application or form, they will not be able to navigate.

For information on security settings, please refer to the *Advance Configuration Utility User's Guide*.

Icons and Terms Used in this Document

Throughout this document we will use terms familiar to experienced users of Advance Web. However, you do not need to have experience with any version or platform of Advance in order to understand this document.

In addition, there are occasions where we wish to bring additional information to your attention. To identify the type of information presented, the text will be preceded by one of the following icons:

Key

🕸 / ⊞ Advance is available in two different interfaces: Web and Windows. Regardless of which interface you are using, you will be accessing the same Advance data. Unless specifically noted otherwise in this document, when we refer to “Advance” we are referring to Advance Web.

ⓘ This “information” icon symbolizes a more detailed explanation of a feature or new term is available.

🛠 This “technical” icon symbolizes the beginning of information, which may be of particular interest to those who are technically inclined, but may have no impact on a typical end-user. If you are simply looking for information on how to perform a task, you may wish to skip this information.

👌 This “OK” icon symbolizes the beginning of information describing expected behavior, which in some way differs from Advance Windows.

👁 This “Watch Out” icon symbolizes the potential for doing something which may adversely affect your system.

1 iModules® Encompass

Concepts and Overview

iModules Encompass is a Web service that manages online communities for institutions and organizations. Integration of iModules Encompass functionality with Advance Web was first introduced in version 9.3.0.1 to share data between the two products. Typical information exchanged includes name, address, email, phone, etc.

The iModules integration consists of three features:

1. Transfer of Biographic data changes only from Advance Web to iModules (Push) using table mapping (see Appendix A) that is consistent with both Advance Web and iModules.
2. Advance Web receives Biographic data changes from iModules (Pull) using table mapping that is consistent with both Advance Web and iModules. Data changes received from iModules is processed through DataLoader to maintain data integrity. Several tables in the database support iModules Biographic Data Interchange.
3. A list of giving history records provided by Advance Web is displayed in the iModules Giving History tab. Transactions to allocations will only display in iModules if the 'Use for AWC Ind' check box is set on the Allocation Stewardship Profile application. Transactions to allocations without this check box checked will not appear in iModules.

   ⓘ *To enable this check box for an Allocation, use Go To to access the Allocation Stewardship Profile application for the allocation you wish to display in iModules and check the 'Use for AWC Ind' check box.*

iModules is included in the Advance Web Core license. In order to use this functionality, an iModules account must be established. For example, if you wish to send data to, or receive data from iModules, a login and password supplied by iModules are required.

ⓘ *For additional compatibility and support information, please refer to the Support section of the Advance Web 9.5.0.0 Release Notes.*

iModules Service Configuration Utility

The iModules Service Configuration Utility is used to set up the schedules for pushing changed data to and pulling changed data from iModules. Using this utility allows for the automation of the data transfer vs. manually transferring the data via the iModules Mediator. Use of this utility is optional and, when configured, works in conjunction with System Option 340, which identifies whether the push/pull process is automated or manual. By default, when your system is installed, the iModules process is set to manual.

An installation of the iModules Service Configuration Utility on your Web server is required to use the iModules Service Configuration Utility. Please refer to the section beginning on page 16 for more information.

iModules Mediator

Both biographic data transfer operations (Push and Pull) can be manually accomplished via the ‘iModules Mediator’ branch on the Home page tree. This is the default functionality for iModules and is not available if the iModules Service is active.

Installation of iModules for using iModules Mediator is automatically included as a part of the Advance Web 9.5.0.0 installation procedure and modification of the web.config file is the only additional requirement. Please refer to the section beginning on page 25 for more information.

How iModules integration works

In a nutshell, you have information about your constituents on Advance and you want to provide them with access to some of that information so they can browse and make their own updates on the Web. By coordinating the content of both Advance and iModules, you can leverage this ability and keep up-to-date records at both locations. Procedurally, you can either manually or automatically send selected data back-and-forth to accomplish this data coordination.

Where to begin

When your institution or organization determines this functionality will be used, the first step is to establish a relationship with iModules. You will be provided with a username and password which provides access to the iModules Web service.

An ‘initial load’ of the data to be stored in iModules is required to identify which records will reside on the iModules site. These records are determined by criteria established by your institution or organization to identify who the iModules users are in your Advance database. Once these records are transferred, only updates are necessary to keep the two sets of records current. How current you want the records to be will determine how frequently your institution or organization transfers records. Discussion on the process for establishing the initial load and identifying subsequent records to transfer manually is in the “Data transfers using iModules Mediator” section of this document, which begins on page 28. Information on modifying the stored procedure used by the iModules Service is in the “Stored Procedure ” section of this document, which begins on page 25.

What happens when data changes occur

The following scenarios will take you through the sequence of events that occur when data changes, either in Advance or iModules. As mentioned earlier, transfer of data can be accomplished either via manual activation or through scheduling automatic transfer times. The scenarios will not differentiate between these two transfer methods.

Terminology to note in these scenarios is data being ‘pushed’, meaning modified information is sent to iModules from Advance to update the constituent’s online records, or ‘pulled’, meaning data that was updated online by constituents in iModules is being brought back into Advance to update their records.

These scenarios make the assumption that Advance is the ‘master repository’ for your data, i.e., the starting point and final destination for all record changes. By identifying Advance as the source, the recommended sequence of data exchange between iModules and Advance will be a push, then a pull, meaning changes in Advance update iModules records first, then any changes on iModules will be brought back to Advance.

Scenario 1 – Data changes in Advance

In this scenario, the address for the entity Mary Phillips was changed by Anthony Barker, an Advance Web user, to reflect her move to a new home.

Now what?

When Anthony made the modification to indicate there was a new address for Mary, Advance made note that address information in the database changed for Mary Phillips. Advance keeps track of all changes like this, so when it’s time for data to be transferred to iModules, there’s an indication of what information needs to be sent to also update Mary’s record there.

At a pre-determined time, the iModules users’ records identified as having changes (including Mary) are gathered in one location (either manually or by a script) and prepared for ‘pushing’. The collection of records is pushed, either manually or automatically, and all of the records in iModules will contain current information from what was changed in Advance.

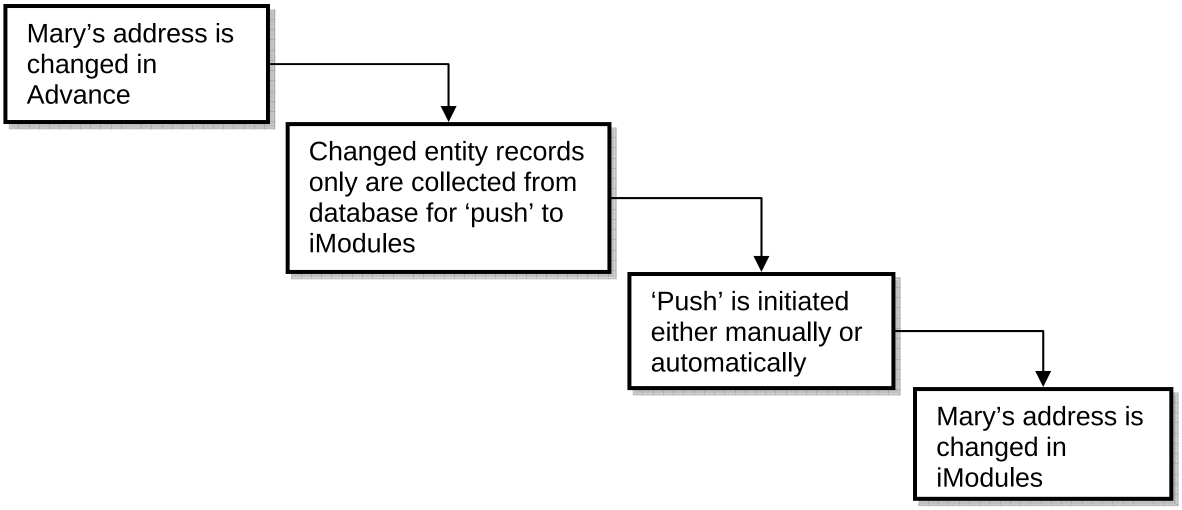

Mary’s address is changed in Advance
Changed entity records only are collected from database for ‘push’ to iModules
‘Push’ is initiated either manually or automatically
Mary’s address is changed in iModules

Figure 1: Advance changes pushed to iModules

Now, the information in the two locations is almost the same. Almost? To be completely current, records require both a push and a pull – the push to get the changes from Advance into iModules and a pull to get any changes in iModules into Advance. Again, this assumes the 'master' records will reside in Advance.

The next scenario provides the solution to keeping the info in both locations as current as possible. It should be noted that the information in iModules and Advance may never be exactly the same, as it is not possible to establish whether changes in Advance records by Advance users and iModules records by your constituents will occur prior to the occurrence of a push or pull. For example, you may just finish pulling the records from iModules and have someone log on to change their address. Because of this, full synchronization of the two data sources may never be possible.

Scenario 2 –Data changes in iModules

In this scenario, iModules data was changed and doesn't reflect what's currently in Advance. Well, since the last time information was transferred between iModules and Advance (a pull), Mary Phillips was busy on the Internet. She logged on and changed her record in iModules to indicate she has a middle initial of 'M.' and that change isn't reflected in Advance.

ⓘ *The system also recognizes information that was removed and blank data is considered a legitimate change.*

What happens now?

iModules keeps track of what changes are made to constituent records, just like Advance does, so when a pull occurs, and Advance brings back records, only those with changes will be available for adding to the database. The pull, like the push, can be initiated either manually or automatically. When the data pull is complete, the records reside in the DataLoader tables and are ready for review. The data review and posting process for iModules information will be established by your institution or organization.

On the next data pull, Mary's updated record is brought over to Advance, so her entity record will match what's in iModules.

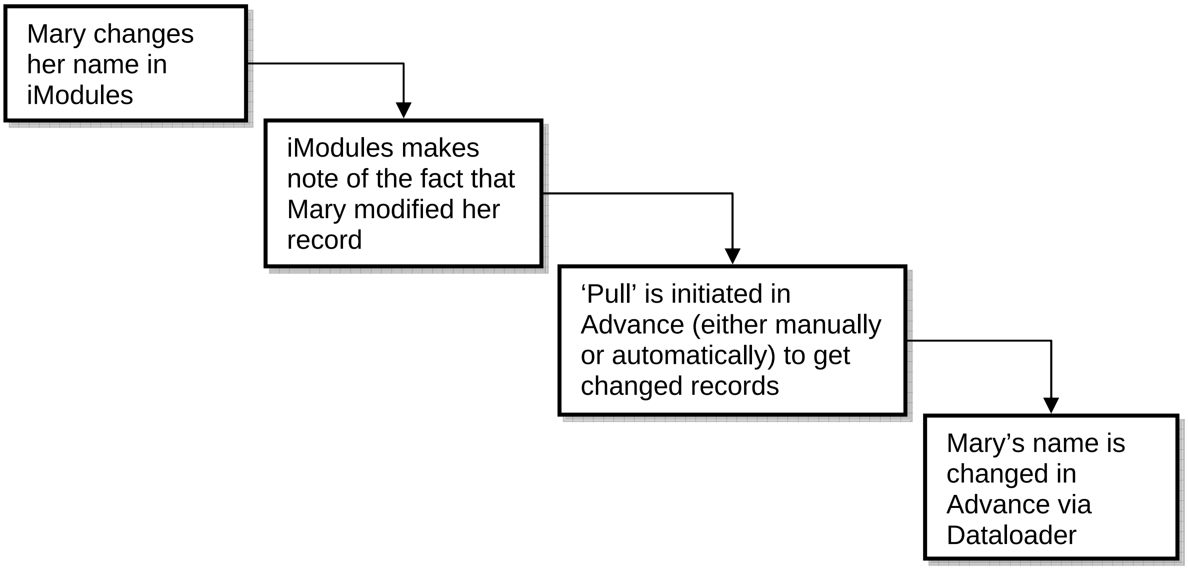

Mary changes her name in iModules
iModules makes note of the fact that Mary modified her record
'Pull' is initiated in Advance (either manually or automatically) to get changed records
Mary's name is changed in Advance via Dataloader

Figure 2: iModules changes pulled to Advance

What’s the takeaway?

The most important thing to note with these two scenarios is that the push/pull processes have a sequential dependency in order to keep the data accurate in both Advance and iModules. This means that, depending on which occurs first, the data may be slightly different in iModules and Advance. The frequency of the push/pull also determines the level of accuracy at each location. Whether you perform these procedures manually, via iModules Mediator, or automatically, via a schedule set in the iModules Service Configuration Utility, care must be taken to make sure the sequencing is correct for the needs of your institution or organization. Also, only those records that change are sent to iModules and brought back into Advance for review in DataLoader, which keeps the data transferred to a minimum.

ⓘ *For data where multiple records may exist for an individual, e.g., degrees, activities, etc., a ‘xsequence value’ is established and is used to differentiate them. For specific information on which fields have xsequence values, please refer to Appendix A, starting on page 131.*

Descriptions of both manual and automatic push/pull are discussed later, so you can determine which is the best for use at your institution or organization.

System option settings

The system options described below are used with iModules functionality.

iModules Service

The following system option requires no modification – it is set by the system.

System Option 340

iModules Service Running

This system option identifies the current status of the iModules Service. When the service is running the iModules Mediator application is disabled in Advance Web. This status is automatically updated by the system and cannot be modified in the Configuration Utility; it is display only.

- N The values ‘Y’ or ‘N’ are set depending on the state of the service. If it is running, the value is ‘Y’; if it is not running, the value is ‘N’. When the value is ‘Y’, the iModules Mediator in Advance Web is disabled to prohibit manual file transfer. This system option is delivered defaulted as ‘N’.

Email

When a push occurs, email records that were modified are checked to see if the iModules Indicator is ‘Y’. The system checks to see if the business indicator in tms_email_type is selected and, if ‘Y’, the modified email is sent to iModules. Also, any additional modified email address whose iModules indicator is ‘Y’ is sent. A maximum of two email addresses can have the iModules indicator set to ‘Y’, one ‘business’ and the other ‘non-business’.

When a pull occurs, modified email records in iModules are brought back for review in DataLoader (as ‘Adds’) prior to being posted to Advance and their email types are set to the defaults defined in System Options 341 and 342:

System Option 341

iModules Default Personal Email Type

This option identifies which email type is used to identify the personal email addresses coming from iModules.

- ♦ Blank The value specified here indicates the value you wish to set for incoming personal addresses.

System Option 342

iModules Default Business Email Type

This option identifies which email type is used to identify the business email addresses coming from iModules.

- ♦ Blank The value specified here indicates the value you wish to set for incoming business addresses.

iModules Service Configuration Utility

The iModules Service Configuration Utility is used to set up the schedules for pushing data to and pulling data from iModules. Using this utility allows for the automation of the data transfer vs. manually transferring the data via the iModules Mediator. Use of this utility is optional and, when configured, works in conjunction with System Option 340, which identifies whether the push/pull process is automated or manual. By default, when your system is installed, the iModules process is set to manual. System Options 341 and 342 are used to identify the default business and personal email types used in iModules.

Utility installation

Similar to the Advance Configuration Utility installation, the iModules Service Configuration Utility has a .exe file that, when run, installs the program. Administrator logon is required, same as the Configuration Utility.

To install this utility:

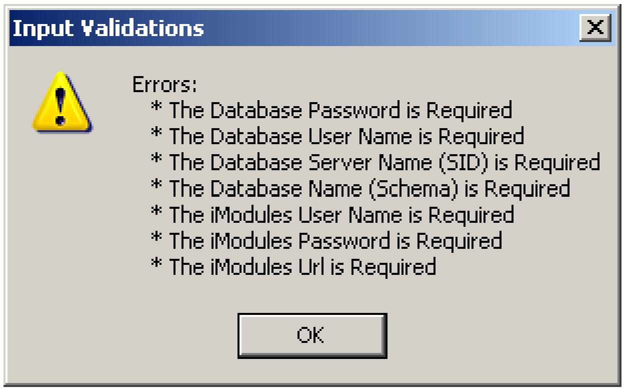
1. Using Windows Explorer on the server where you downloaded the Advance Web installation files, navigate to the Executables directory and double-click SunGardHE.iModulesMediatorService.msi to launch the iModules Service Configuration Utility installer.
2. Click Next > on the introduction screen to continue with the installation.
3. Read the license agreement, then select the top radio button to accept the license agreement and click Next > to continue.
4. Personalize the installation with User Information, then click Next > to continue.
5. Define the Destination Folder for the installation, then click Next > to continue.
6. Click Next > to complete the installation.
7. When the system is updated, click Finish to exit the installation. The iModules Service Configuration Utility automatically launches and is ready for configuration.

Entering utility settings

The functionality included with this configuration utility requires initial evaluation of the types of data, volume and frequency of transfer (push and pull). Once this information is established, launch the utility and configure the tabs in the following sections.

When the utility is configured, the Start and Stop buttons on the upper left of the window are used to initiate the service to transfer data between Advance Web and iModules. Starting the service inactivates iModules Mediator via System Option 340, as described in the previous section.

When entering information in the configuration utility, you can hover your mouse over the fields and display tool tips for assistance. Also, you can click Refresh to update the screen and confirm your settings. When you finish entering information, clicking Save saves entered information on all tabs and an information dialog box displays indicating your Save was successful. If there are errors in your settings, an error message displays.

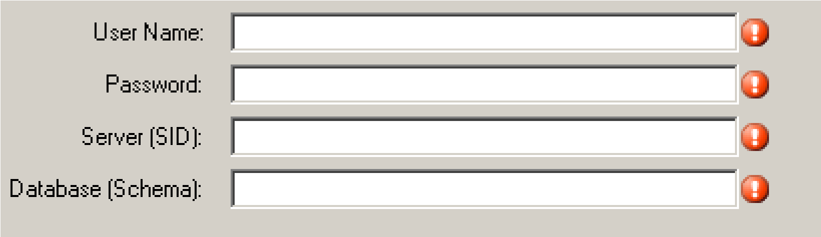


Input Validations

Errors:
* The Database Password is Required
* The Database User Name is Required
* The Database Server Name (SID) is Required
* The Database Name (Schema) is Required
* The iModules User Name is Required
* The iModules Password is Required
* The iModules Url is Required

Figure 3: iModules error message on Save

Fields with errors are identified with ❗.



User Name:

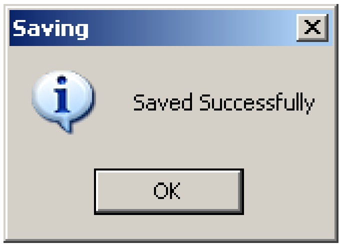
Password:

Server (SID):

Database (Schema):

Figure 4: Identification of fields with errors

A successful Save displays the following:



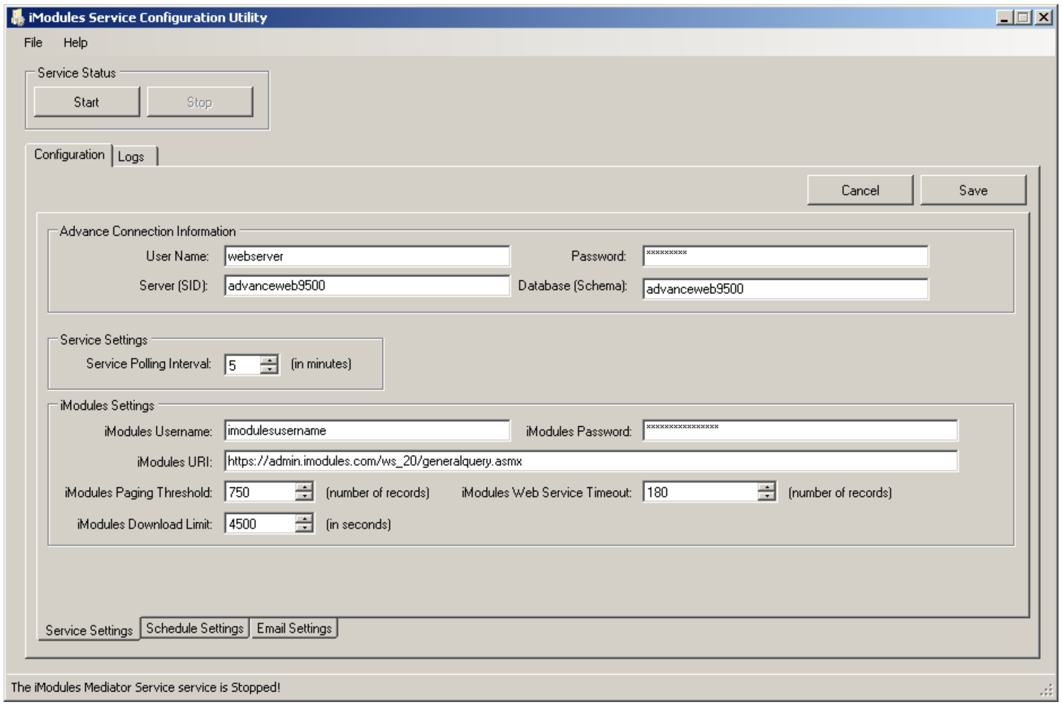
Saving

Saved Successfully

Figure 5: 'Successful Save' message

Service settings tab

Service Settings identifies your database connection settings and iModules access settings.



Advance Connection Information

User Name: webserver

Password: *********

Server (SID): advanceweb9500

Database (Schema): advanceweb9500

Service Settings

Service Polling Interval: 5 (in minutes)

iModules Settings

iModules Username: imodulesusername

iModules Password: ****************

iModules URI: https://admin.imodules.com/ws_20/generalquery.asmx

iModules Paging Threshold: 750 (number of records)

iModules Web Service Timeout: 180 (number of records)

iModules Download Limit: 4500 (in seconds)

The iModules Mediator Service service is Stopped!

Figure 6: Service Settings Tab

<u>**Advance Connection Information**</u>

User Name Enter the Advance schema owner name.

Password Enter the Advance schema owner password.

Server (SID) Enter the name of the server where the Advance database resides.

Database (Schema) Enter the name of the Advance schema.

<u>**Service Settings**</u>

Service Polling Interval The frequency in minutes (1-59) that the polling service checks to see if it needs to run.

<u>**iModules Settings**</u>

iModules Username The name identified with the logon for your iModules account for transferring data.

iModules Password The password for your iModules account for transferring data. For security, this text field is obfuscated as soon as you enter the value.

ⓘ *The **iModules Username** and **iModules Password** supplied to your institution or organization by iModules will provide secure access when data is sent between iModules and Advance.*

You must encrypt the Username and Password using the SunGardHE.Installs.Adved100.exe utility (please refer to theAdvance Web installation instructions).

Both the username and password are case sensitive.

iModules URI This identifies the secure web address where the connection between Advance Web and iModules occurs. If iModules indicates the address for your institution or organization is not the same as the value indicated here, enter the correct value.

iModules Paging Threshold This value identifies the maximum number of rows of data allowed to be sent (pushed) or received (pulled) in a single set. For example, if there were 5,000 rows of data being sent and the paging threshold was set to 1,000, then there would be 5 sets of data sent, each set containing 1,000 rows.

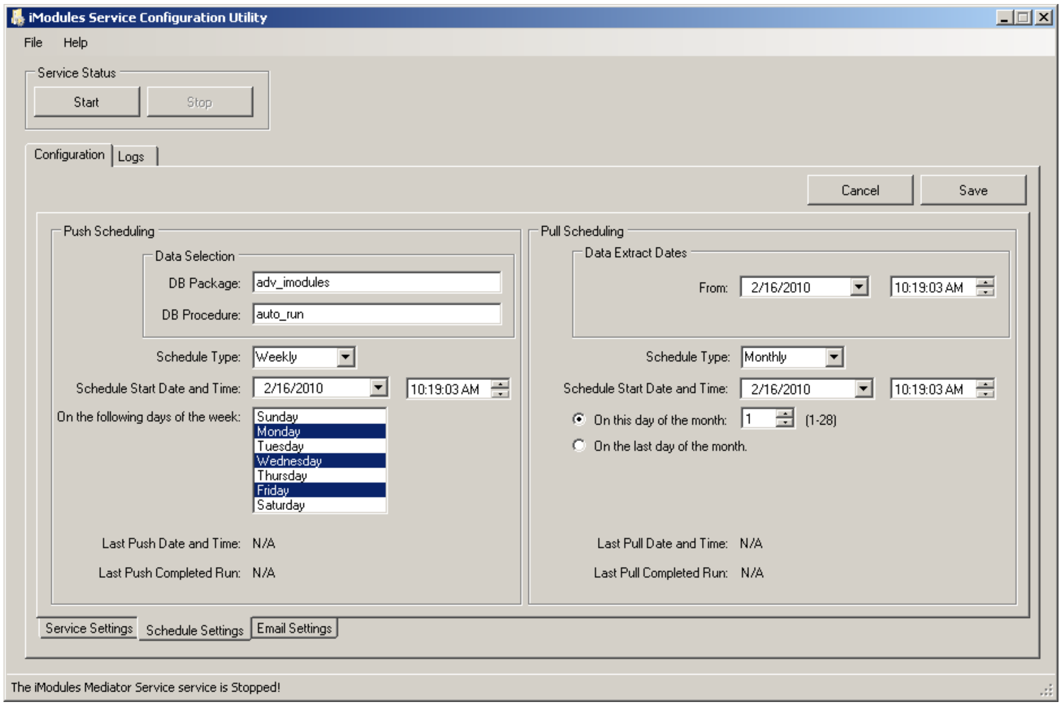
iModules Web Service Timeout The maximum allowable time, in seconds, between a request being sent and a response confirming the request was fulfilled.

iModules Download Limit This value indicates the maximum allowable number of rows for a download (pull) operation.

Schedule settings tab

Schedule Settings is used to identify the scheduling of the data to be pushed to iModules and to be pulled from iModules. These schedules work independently of each other and can be set to different frequencies, depending on the volume of records and needs of your institution or organization. The schedules you define should not be the same time. Also, if one of these scheduled operations (push or pull) is still active when it is time for the next scheduled operation to initiate, the second scheduled operation will not start until the first is complete. All dates are validated.

ⓘ *SunGard Higher Education recommends the sequence of push, then pull.*



DB Package: adv_imodules
DB Procedure: auto_run
Schedule Type: Weekly
Schedule Start Date and Time: 2/16/2010 10:19:03 AM
On the following days of the week: Sunday, Monday, Tuesday, Wednesday, Thursday, Friday, Saturday
Last Push Date and Time: N/A
Last Push Completed Run: N/A

From: 2/16/2010 10:19:03 AM
Schedule Type: Monthly
Schedule Start Date and Time: 2/16/2010 10:19:03 AM
● On this day of the month: 1 (1-28)
○ On the last day of the month.
Last Pull Date and Time: N/A
Last Pull Completed Run: N/A

The iModules Mediator Service service is Stopped!

Figure 7: Schedule Settings Tab

Push Scheduling

<u>Data Selection</u>

DB Package The name of the package that prepares the data to be pushed.

DB Procedure The controlling stored procedure within the package to start the automatic process of data preparation. The script that prepares the data can be modified to suit the needs of your institution or organization. For more information on how to modify the script, please see the Stored Procedure section, which starts on page 25

Schedule Type The frequency of the data push. Values in this drop down list include Hourly, Daily, Weekly and Monthly.

Schedule Start Date and Time The start date and time for the data pull schedule for the frequency type defined above. The format for the date field is MM/dd/yyyy. The format for the time field is hh:mm:ss tt. Used by stored procedures to identify data for push; this is also the value used to determine which data is gathered for the initial push operation, subsequent push operations will use the Last Completed Date Time.

ⓘ *If you want to change the start date for a push or pull, you must stop the service, update the start date and restart the service.*

Depending on the Schedule Type, the following will display:

Hourly

Hour This box increments from 1-23 and identifies how frequently to run the push, e.g., a setting of 4 will push data 4 hours after the initial Schedule Start Date and Time and every 4 hours thereafter.

Daily

Day This time box allows you to identify the hh:mm:ss AM/PM that you want the push to run each day.

Weekly

On the following days of the week This box contains a list of the days of the week. You can select a single or multiple days to identify which day(s) of the week you want the push to occur. The time of each push will coincide with the start time defined.

Monthly

On this day of the month When this radio button is selected, you identify that the push should occur on the day of the schedule type defined. , e.g., if this field is 5, the data push will occur on the 5th day of each month. Values available are 1-28. You are not able to schedule 29, 30 or 31 using this radio button; if you want the push to occur after the 28th day of the month, you must use the option below.

On the last day of the month When this radio button is selected, you identify that the data push should occur on the last day of each month. Depending on the month, your data push would occur on the 28th (February), 29th (February in a leap year), 30th or 31st.

Last Push Date and Time This display only field is incremented by the system and indicates when the last push was initiated.

Last Push Completed Run This display only field is incremented by the system and indicates when the last push was completed.

Pull Scheduling

Data Extract Dates

From The start date and time for the data you want to pull.

ⓘ *Data changes recorded in iModules are Central Standard Time, so the time for data pulls in the 'From' time in the 'Data Extract Dates' section must take the difference between your local time and CST into account.*

Schedule Type The frequency of the data pull. Values in this drop down list include Hourly, Daily, Weekly and Monthly.

Schedule Start Date and Time The start date and time for the data pull schedule for the frequency type defined above. The format for the date field is MM/dd/yyyy. The format for the time field is hh:mm:ss tt.

If you want to change the start date for a push or pull, you must stop the service, update the start date and restart the service.

<u>Depending on the Schedule Type, the following will display:</u>

Hourly

Hour This box increments from 1-23 and identifies how frequently to run the push, e.g., a setting of 4 will push data 4 hours after the initial Schedule Start Date and Time and every 4 hours thereafter.

Daily

Day This time box allows you to identify the hh:mm:ss AM/PM that you want the pull to run each day.

Weekly

On the following days of the week This box contains a list of the days of the week. You can select a single or multiple days to identify which day(s) of the week you want the pull to occur. The time of each pull will coincide with the start time defined.

Monthly

On this day of the month When this radio button is selected, you identify that the pull should occur on the day of the schedule type defined. , e.g., if this field is 5, the data pull will occur on the 5th day of each month. Values available are 1-28. You are not able to schedule 29, 30 or 31 using this radio button; if you want the pull to occur after the 28th day of the month, you must use the option below.

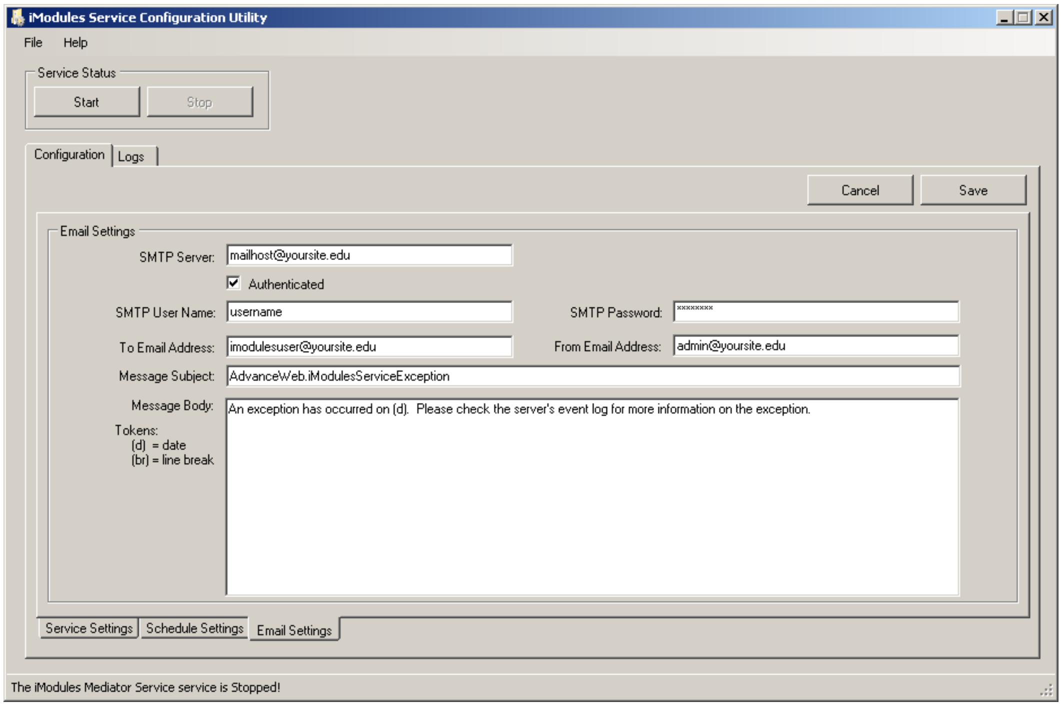
On the last day of the month When this radio button is selected, you identify that the data pull should occur on the last day of each month. Depending on the month, your data pull would occur on the 28th (February), 29th (February in a leap year), 30th or 31st.

Last Pull Date and Time This display only field is incremented by the system and indicates when the last pull was initiated.

Last Pull Completed Run This display only field is incremented by the system and indicates when the last pull was completed.

Email settings tab

Email Settings is used for notification that an exception occurred during the push or pull of data. Once notification is received, the log file can be reviewed to identify the cause of the exception.



SMTP Server: mailhost@yoursite.edu

☑ Authenticated

SMTP User Name: username

SMTP Password: ********

To Email Address: imodulesuser@yoursite.edu

From Email Address: admin@yoursite.edu

Message Subject: AdvanceWeb.iModulesServiceException

Message Body: An exception has occurred on (d). Please check the server's event log for more information on the exception.

Tokens:
(d) = date
(br) = line break

The iModules Mediator Service service is Stopped!

Figure 8: Email Settings Tab

Email Settings

SMTP Server The name of the SMTP server used for sending messages.

Authenticated When this check box is checked, it indicates the SMTP mail server you are using for notification requires authentication. When this checkbox is unchecked, any values in the following two fields are deleted.

SMTP User Name The logon name for the account on the SMTP mail server used for sending notifications.

SMTP Password The password for the account on the SMTP mail server used for sending notifications. For security, this text field is obfuscated as soon as you enter the value.

To Email Address The email address, using a valid email address format, of the individual to notify when an exception occurs during a scheduled data transfer.

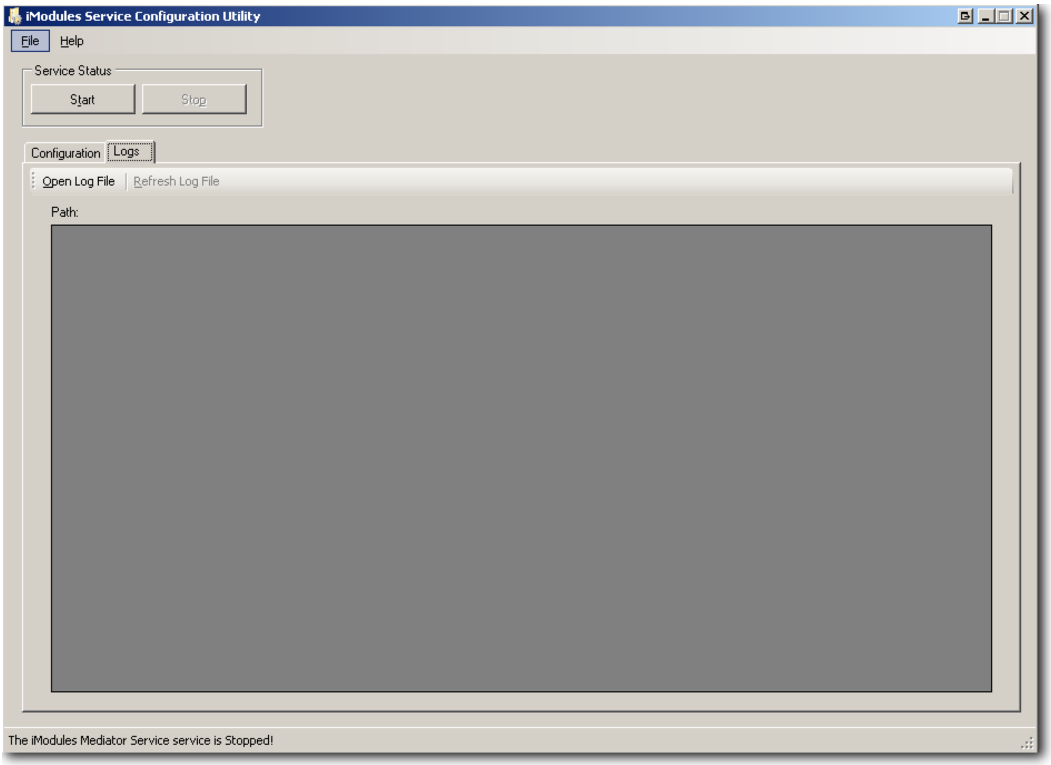
From Email Address The address to identify the source of the email.

Message Subject The text to include as the subject for the email sent indicating an exception occurred during the scheduled data transfer.

Message Body The text to include in the body of the email indicating an exception occurred during the scheduled data transfer. The ‘(d)’ token can be passed to identify the date/time of the exception. By default, the text in the message body indicates the log file should be checked for additional information regarding the exception.

Logs

Log entries are created when push or pull operations are initiated, cancelled, or completed including how many records have been transferred. When the email recipient defined on the Email Settings tab is notified an exception occurred, the log is referenced to view the details of the exception for identification of the problem and determination of corrective actions to apply.



The iModules Mediator Service service is Stopped!

Figure 9: Logs tab

The Logs tab provides access to log files generated by iModules. By default, the system log window is empty when you access it. Accessing and reading the details of these files is similar to viewing logs in the Windows Event Viewer.

To open a log file:

1. While current on the Logs tab, click ‘Open Log File’ beneath the tab, or use the keyboard combination ALT-O.
2. Navigate to c:\temp, the directory where your log files are stored, select the file and click ‘Open’.
3. You can view the individual row details of the log by double-clicking on ❌ or ❓ to the left of the row.

There are two types of log files:

Mediator logs These logs are generated when there is an exception and provide specific information on what occurred.

Transaction logs These logs provide information on individual transactions processed by the service.

Stored Procedure for data collection

A stored procedure (auto_run) is provided in the adv_imodules database package for the automatic collection of records for pushing to iModules when you are using the iModules Service. This stored procedure runs get_changed_id_numbers to identify which records in Advance have modifications to send to iModules and records the IDs in the temp_imodules_staging table. The autorun_extract_report procedure then collects the changed data, based on the IDs, and puts it in the imodules_upload table. It is the data from this table that is pushed to iModules.

Because the guidelines at institutions and organizations vary, please review the stored procedure and modify it if necessary to meet your specific needs.

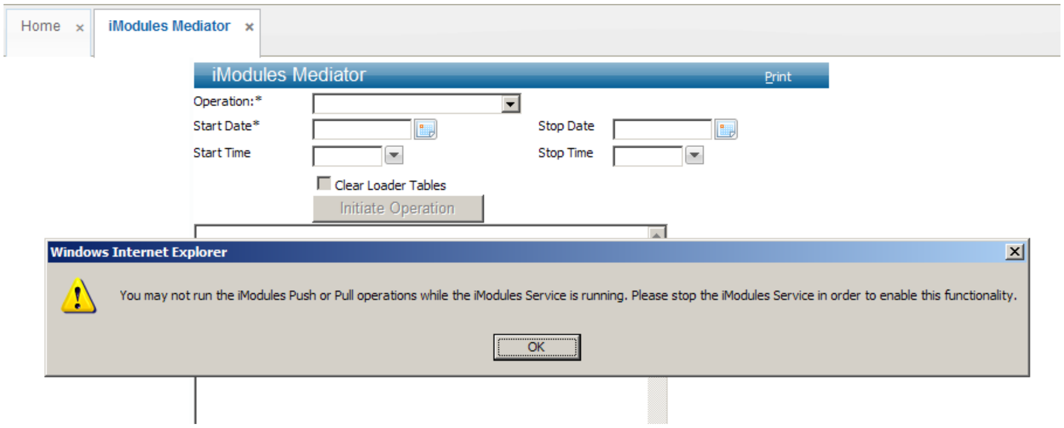
Starting the service

Once you complete the setup in the iModules Service Configuration Utility and make any modifications to the stored procedure, if necessary, you are ready to start the service.

1. While current on any tab of the iModules Service Configuration Utility, click Start to start the service.

iModules Mediator configuration

Both biographic data transfer operations (Push and Pull) can be manually accomplished via the ‘iModules Mediator’ branch on the Home page tree. Effective with this release, this is the default functionality for iModules and is disabled if the iModules Service is active. An error message displays if an attempt is made to use iModules Mediator when it is disabled.



You may not run the iModules Push or Pull operations while the iModules Service is running. Please stop the iModules Service in order to enable this functionality.

Figure 10: iModules Mediator error message when disabled

Installation of iModules for using iModules Mediator is automatically included as a part of the Advance Web 9.5.0.0 installation procedure and modification of the web.config file is the only additional requirement.

For additional compatibility and support information, please refer to the Support section of the Advance Web 9.5.0.0 Release Notes.

To modify the web.config file:

A. The Web.Config file is stored in the root directory where Advance Web is installed. It contains information regarding the authentication and authorization methods, security parameters, etc. Some values in the Web.Config file may be edited using a text editor or via IIS Manager, while others may not. During the Web Server portion of the installation, certain values, such as usernames and passwords are encrypted in this file.

B. Installation includes modification of the web.config file with the addition of the following lines above the </appSettings> line to control the Advance Web integration of iModules:

```
<!-- iModules integration -->
 <add key="iModulesUsername" value="Enter your user name here" />
 <add key="iModulesPassword" value="Enter your password here" />
 <add key="iModulesPagingThreshold" value="1000" />
 <add key="iModulesWebSvcTimeout" value="90" />  <!-- in seconds -->
 <add key="iModulesUri"
value="https://admin.imodules.com/ws_20/generalquery.asmx" />
 <add key="iModulesDownloadLimit" value="5000" />
 </appSettings>
```

<u>Descriptions of the web.config keys:</u>

The **iModulesUsername** and **iModulesPassword** supplied to your institution or organization by iModules provides secure access when data is sent to and received from iModules from Advance Web. The Username/Password is also used when an iModules user indicates they want to view their giving history and the system retrieves the data.

👁 *Both the Username and Password are case sensitive, and must be encrypted using the SunGardHE.Installs.Adved100.exe utility.*

iModulesPagingThreshold identifies the maximum number of rows of data allowed to be sent(pushed) or received(pulled) in a single set. For example, if there are 5,000 rows of data being sent and the paging threshold is set to 1,000, then 5 sets of data would be sent, each set containing 1,000 rows.

iModulesWebSvcTimeout identifies the maximum allowable time between a request being sent and a response confirming the request was fulfilled.

iModulesUri identifies the secure web address where the connection between Advance Web and iModules occurs. If iModules indicates the address for your institution or organization is not the same as the value indicated here, enter the correct value.

iModulesDownLoadLimit indicates the maximum allowable number of rows for a download (pull) operation.

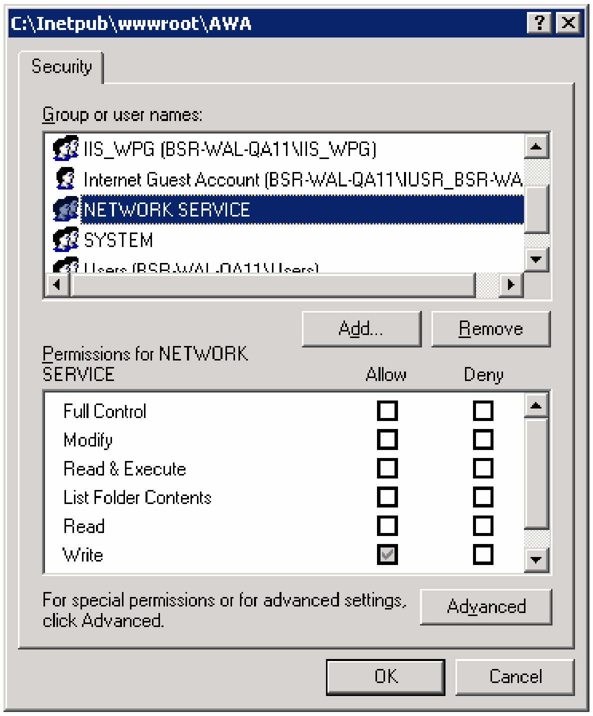
C. Make the following change beneath the <!-- Site Settings --> line in the web.config file to represent your version of Advance Web, for example:

```
<add key="PageTitle" value="Advance 9.4.0.0" />
```

Log Directory Security

Ensure that the Log subdirectory under the Advance Web directory has write permissions for the NETWORK SERVICE user. To confirm this, open IIS Manager on the Web Server where Advance Web is installed and view the security settings for the directory.

ⓘ *Failure to perform this step will result in a system error when log files are generated.*



Group or user names:

- IIS_WPG (BSR-WAL-QA11\IIS_WPG)
- Internet Guest Account (BSR-WAL-QA11\IUSR_BSR-WA
- NETWORK SERVICE
- SYSTEM
- Users (BSR-WAL-QA11\Users)

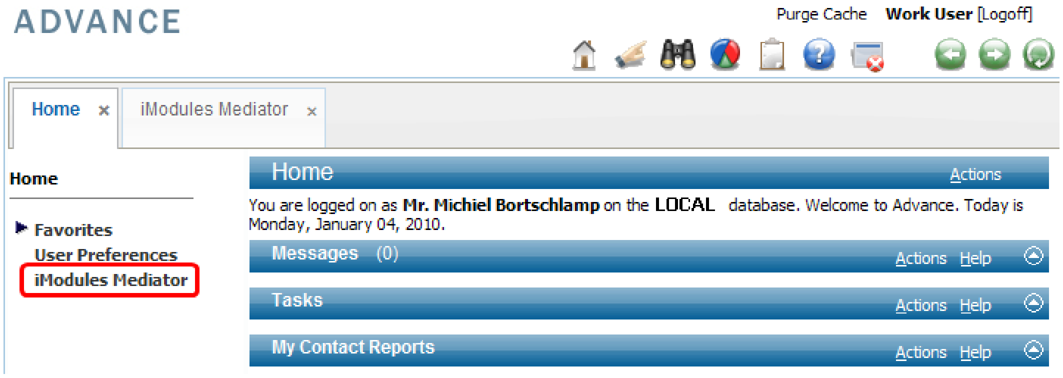
Permissions for NETWORK SERVICE	Allow	Deny
Full Control	□	□
Modify	□	□
Read & Execute	□	□
List Folder Contents	□	□
Read	□	□
Write	☑	□

For special permissions or for advanced settings, click Advanced.

Figure 11: Network Service Security

For additional information on how to perform any of these steps, please refer to your database analyst or contact support at cs-support@sungardhe.com.

Once you complete the initial configuration, the security established for the user(s) responsible for iModules data transfers will provide them with access to the iModules Mediator branch of the page tree on the Home Page. Users without the proper security will not have access to iModules Mediator.



You are logged on as **Mr. Michiel Bortschlamp** on the **LOCAL** database. Welcome to Advance. Today is Monday, January 04, 2010.

Figure 12: iModules Mediator Branch on Page Tree

Initial data transfer

The initial transfer of Advance records to iModules will be accomplished by a method arranged between your institution or organization and iModules. Typically, this data is transferred via a large text file, not an upload through Advance Web iModules Mediator.

Maintenance upload (Push)

Regular updates that reflect changes to the data in the Advance database are performed on a schedule that best suits the needs defined by your institution or organization. The process involves Selecting the Operation, date and time, then, initiating the push. Modifications to the interface in this release include a drop-down list to identify the push operation and start date and time for the push.

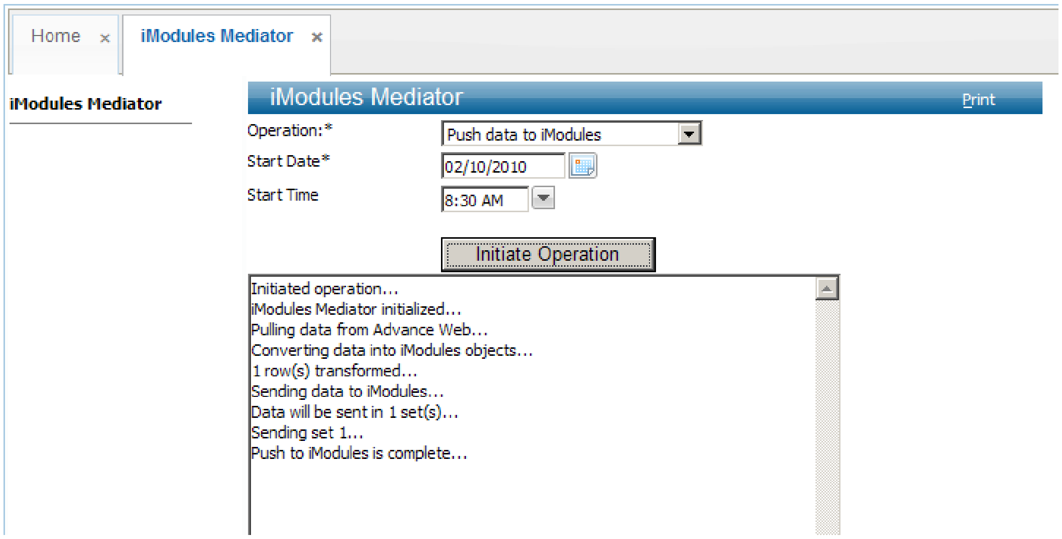
To Push data to iModules via iModules Mediator:

1. While current on the Home Page, click iModules Mediator on the page tree. The iModules Mediator application displays.
2. From the Operation drop-down list, select Push data to iModules. Select a Start Date and Start Time for the data
3. Click Initiate Operation to commence the upload of data.

A stored procedure (auto_run) is provided in the adv_imodules database package for the automatic collection of records for pushing to iModules when you are using the iModules Mediator. When you click Initiate Operation, this stored procedure runs get_changed_id_numbers to identify which records in Advance have modifications to send to iModules and records the IDs in the temp_imodules_staging table. The autorun_extract_report procedure then collects the changed data, based on the IDs, and schedule start date and time, and puts it in the imodules_upload table. It is the data from this table that is pushed to iModules.

Because the guidelines at institutions and organizations vary, please review the stored procedure and modify it if necessary to meet your specific needs.

4. During the ‘Push’ process, messages display indicating progress. When the transfer of data is complete, a message indicates ‘Push to iModules is complete’.



Operation:* Push data to iModules

Start Date* 02/10/2010

Start Time 8:30 AM

```
Initiated operation...
iModules Mediator initialized...
Pulling data from Advance Web...
Converting data into iModules objects...
1 row(s) transformed...
Sending data to iModules...
Data will be sent in 1 set(s)...
Sending set 1...
Push to iModules is complete...
```

Figure 13: iModules Mediator Push is complete

iModules transaction logging occurs in the Log folder of the root Advance Web installation directory (a separate file for each ‘Push’), and contains iModules Exception type, iModules Exception Code, Constituent Id and a textual description.

Maintenance Download (Pull)

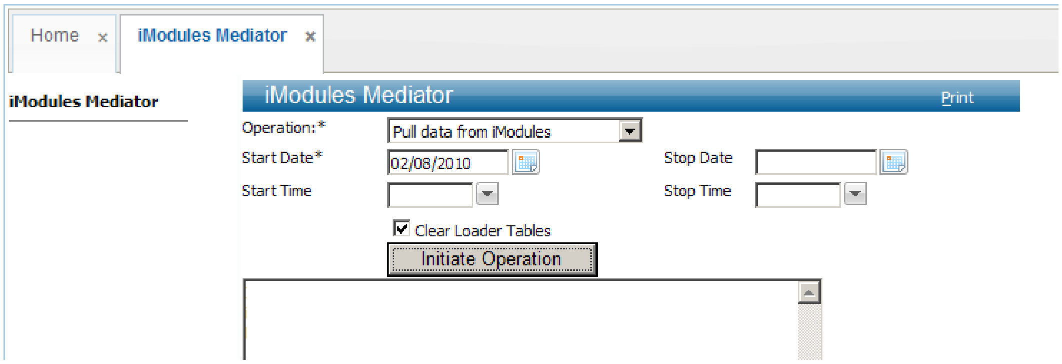
Regular updates that reflect changes to the data in iModules are performed on a schedule that best suits the needs defined by your institution or organization. Downloads of data will update the Advance database with changes made by constituents via iModules. Prior to being loaded into the Advance database via DataLoader, the data can be reviewed.

Data Pull

On a schedule established at your institution or organization, data from iModules can be pulled into the Advance database to record updates made by your constituents via iModules. The ‘Pull’ is accomplished via the iModules Mediator.

To download the data to iModules:

1. While current on the Home Page, click iModules Mediator on the page tree. The iModules Mediator application displays.



Operation:* Pull data from iModules
Start Date* 02/08/2010
Stop Date
Start Time
Stop Time
☑ Clear Loader Tables

Figure 14: iModules Mediator - Pull

Identify the date range and times or, at a minimum, the Start Date, for the data to extract from iModules by entering dates in the date fields. You can either enter a complete date, e.g., MMDDYYYY, or click 📅 to select a date from the popup calendar. Time is HH:MM:SS.

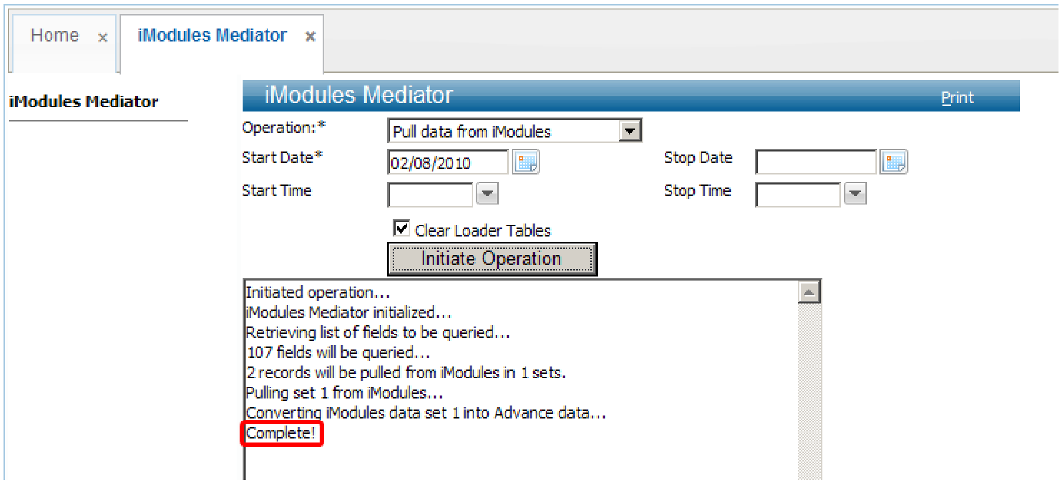
ⓘ *Data changes recorded in iModules are Central Standard Time, so the time for data pulls in the ‘Start Time’ time in the ‘Pull Data from iModules’ section must take the difference between your local time and CST into account.*

2. Identify whether you want to clear the DataLoader tables (☑ Clear Loader Tables) prior to moving the data into them or append the data to the existing data in the DataLoader tables (□ Clear Loader Tables).

3. Once you identify the criteria of the data to pull, click Initiate Operation to commence the download of data.

4. Data is retrieved from iModules and distributed to the Advance Data Loader tables during the download process. When the data transfer is complete, the message ‘Complete!’ displays.

ⓘ *Should there be a problem pulling the data from iModules, an error message will display.*



Operation:* Pull data from iModules

Start Date* 02/08/2010

Stop Date

Start Time

Stop Time

☑ Clear Loader Tables

```
Initiated operation...
iModules Mediator initialized...
Retrieving list of fields to be queried...
107 fields will be queried...
2 records will be pulled from iModules in 1 sets.
Pulling set 1 from iModules...
Converting iModules data set 1 into Advance data...
Complete!
```

Figure 15: iModules Mediator Pull is complete

5. Once the data resides in the Advance database in the DataLoader tables, you can use DataLoader to review and post the data. For information on procedures to review and post data via DataLoader, please refer to the *Advance Web DataLoader User’s Guide*.

ⓘ *There is a Batch Type named ‘iModules’ (C) for creating a batch with this data.*

2 Exchange 2007 Task and Calendar Integration

Overview & Usage

This feature allows the entry of tasks in Advance Web to automatically generate updates for a corresponding Outlook calendar or task entry. This task then continues to be synchronized between Advance Web and Outlook as changes are made in either application. This functionality supports the use of Advance Web as the main data entry point for tasks while providing flexibility for users to view or update the information in Outlook or Advance Web. This integration also allows institutional personnel to work more efficiently and allows institutions to accurately and completely capture the necessary statistics to measure the work being performed by its personnel.

This functionality is included with the core license.

Environment

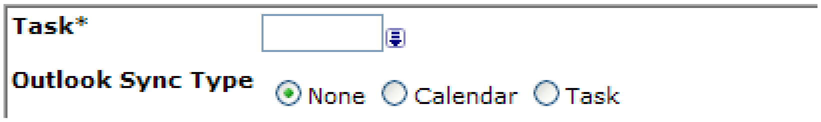
Exchange 2007 is required to take advantage of this feature; Web Services are used to perform the synchronization.

Important Notes

- Advance Web contains logic related to the Exchange task and calendar integration functionality, which is only invoked when you are current on task forms, based on their form IDs. If you have profile-specific changes for the SunGard Higher Education delivered task forms, this logic will be invoked, since the form IDs are identical. However, if you clone one of the task forms, a new form ID is generated and this logic is not available for the new form. If you cloned a task form prior to Advance Web 9.3 and you wish to utilize synchronization functionality, you must re-engineer your cloned changes as profile-specific changes using the current version of the default form.
- This enhancement is supported for Advance Web only. It is a not supported for Advance Windows or Advance Mobile. Therefore, the versions of the task forms remain per the standard for each of these applications. Users will not be able to view any of the Outlook task fields or take advantage of the synchronization function from either of these two products.
- This functionality only applies to new tasks; tasks created prior to 9.3 will have their ‘Outlook Sync Type’ identified as ‘None’ and cannot be synchronized.
- The synchronization of document attachments for Tasks is not supported for this functionality.
- AdvLoader and DataLoader are not supported for this functionality.

General Information

If Outlook synchronization is turned on for a task context (via System Option 295), when ‘New’ is selected on the task form header, you will see the Outlook Sync Type field with three radio buttons:



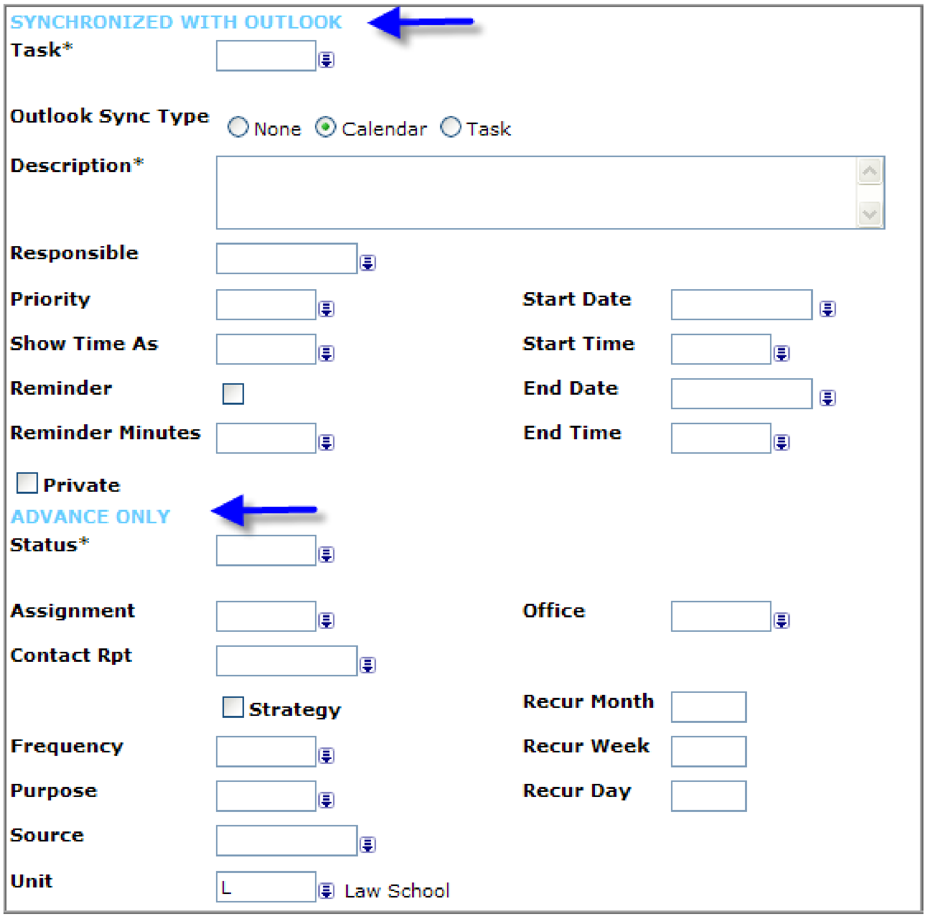
Task*

Outlook Sync Type ● None ○ Calendar ○ Task

Figure 16: Outlook Sync Type

The default is *None*, which indicates this task will not synchronize with Outlook. If you wish to synchronize the task as an Outlook Calendar entry, select the Calendar radio button. To synchronize the task as an Outlook Task, select the Task radio button.

When either the ‘Calendar’ or ‘Task’ radio button is selected, the display of the task form will immediately change, because different data elements are available based on the selected radio button. The task form has two distinct sections. The example below illustrates the Prospect Task form as it displays if the *Calendar* synchronization option is selected:



SYNCHRONIZED WITH OUTLOOK

Task*

Outlook Sync Type ○ None ● Calendar ○ Task

Description*

Responsible			
Priority		Start Date	
Show Time As		Start Time	
Reminder	□	End Date	
Reminder Minutes		End Time	

□ Private

ADVANCE ONLY

Status*			
Assignment		Office	
Contact Rpt			
	□ Strategy	Recur Month	
Frequency		Recur Week	
Purpose		Recur Day	
Source			
Unit	L Law School		

Figure 17: Outlook Sync Type form

The SYNCHRONIZED WITH OUTLOOK section displays all of the fields that will synchronize with Outlook. The ADVANCE ONLY section displays the fields that are for Advance Web purposes only and do not synchronize with Outlook.

For synchronization to take place between Advance Web and Outlook, the task **must** be added in Advance Web and either the Calendar or Task radio button must be selected. Advance Web will have no knowledge of tasks or calendar entries initially created in Outlook.

Responsible Assignment

All tasks synchronizing with Outlook require a *Responsible* value to be entered when the task is added. This is true whether you are creating an Outlook task or a calendar entry. This is required so Advance Web can determine which Outlook user should have these updates applied.

Field Validation

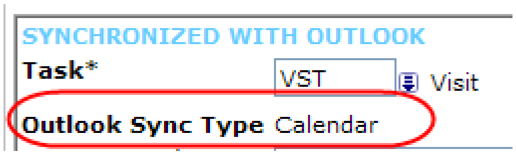
The behavior of the fields in the ***Synchronized with Outlook*** section on a Task form is consistent with Outlook. As an example, the drop-down values match those available in Outlook for the corresponding field. This is critical to ensuring that synchronization between the two applications occurs correctly.

Deleting Synchronized Tasks and Calendar Entries

If you wish to delete a task that was synchronized as an Outlook Task or Calendar entry, this deletion should be performed in Advance Web. This is the only method to permanently remove the task/calendar entry from both Advance and Outlook. If a task or calendar entry is deleted in Outlook, it will not result in the deletion of the associated task in Advance. In fact, if the corresponding Advance Web task is updated in the future, it will result in the re-creation of the Outlook task or calendar entry.

Additional Usage Rules

- Once a task is added and saved to synchronize with Outlook, the Outlook Sync Type value cannot be changed. It will show as display only:



SYNCHRONIZED WITH OUTLOOK

Task* VST Visit

Outlook Sync Type Calendar

Figure 18: Display only Outlook Sync Type after saving

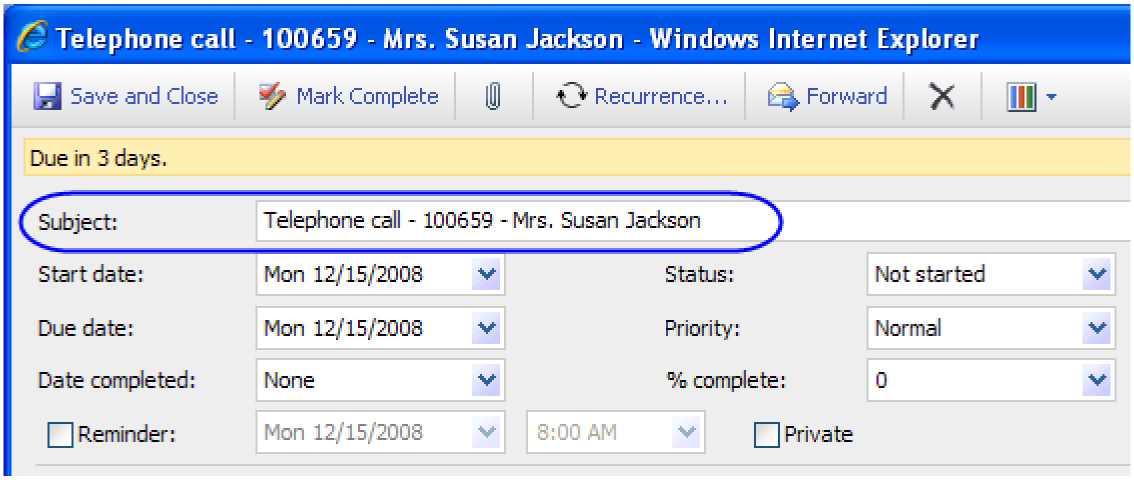
- If you initially add a task and keep the default of *None* for the Outlook Sync Type, you can change this to Calendar or Task in a subsequent update. Again, once it is changed to Calendar or Task and this change is saved, this field can no longer be modified and will be display only.

Task Description Field Length

The length of the Task Description field in the Task table in Advance is 4,000 characters to accommodate the synchronization of larger amounts of text.

Subject Line

The Subject Line value of an Outlook Task or Calendar entry created via synchronization is comprised of three components and separated by dashes, as illustrated below:



Telephone call - 100659 - Mrs. Susan Jackson - Windows Internet Explorer
Save and Close
Mark Complete
Recurrence...
Forward

Due in 3 days.

Subject: Telephone call - 100659 - Mrs. Susan Jackson

Start date: Mon 12/15/2008
Status: Not started

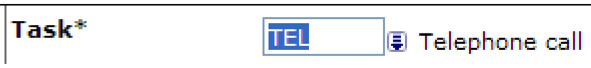
Due date: Mon 12/15/2008
Priority: Normal

Date completed: None
% complete: 0

□ Reminder: Mon 12/15/2008 8:00 AM
□ Private

Figure 19: Outlook task

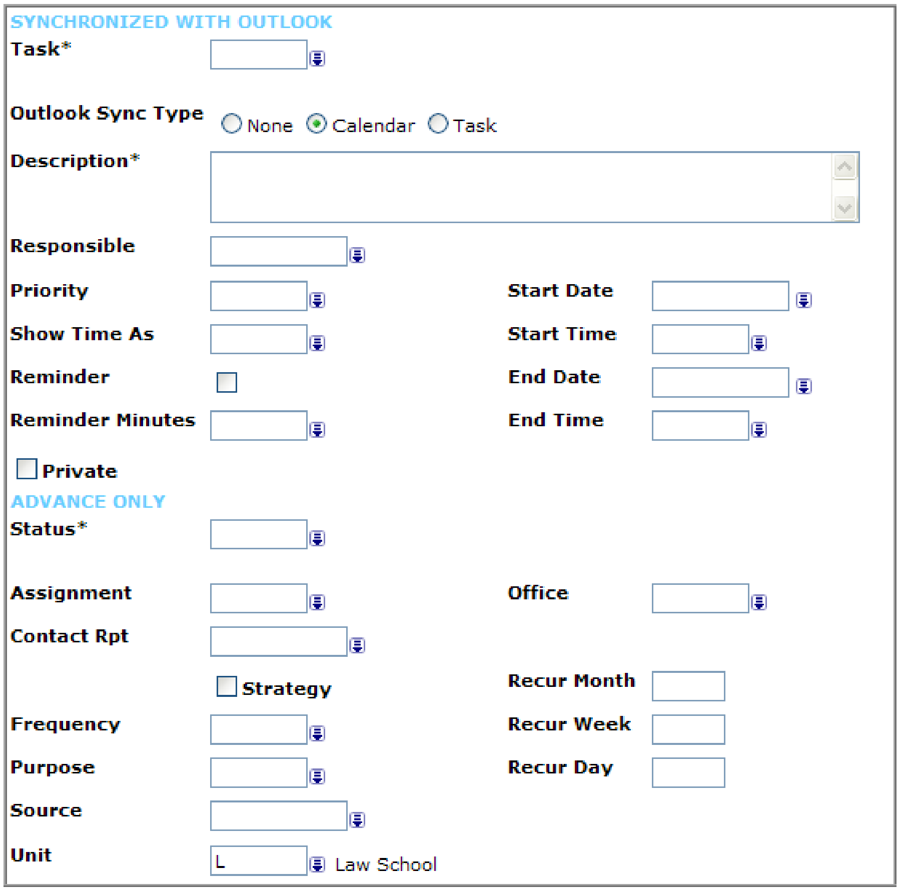
The above example is an entity task being viewed in Outlook. The three components are the following:

1st	The Task Type description	 Task* TEL Telephone call
2nd	Context ID	In this case it is the Entity ID. This will vary by context.
3rd	Context Description	In this case it is the Entity's Preferred Name. This will vary by context.

Given that Advance Web is creating this value by concatenating three separate components, updates to the Subject line in Outlook will be ignored by Advance Web.

Outlook Calendar Creation

When the ‘Calendar’ radio button is selected as the Outlook Sync Type, the Task form will dynamically change to show the calendar version of the task form.



SYNCHRONIZED WITH OUTLOOK

Task*			
Outlook Sync Type	○ None ● Calendar ○ Task		
Description*			
Responsible			
Priority		Start Date	
Show Time As		Start Time	
Reminder	□	End Date	
Reminder Minutes		End Time	
□ Private			

ADVANCE ONLY

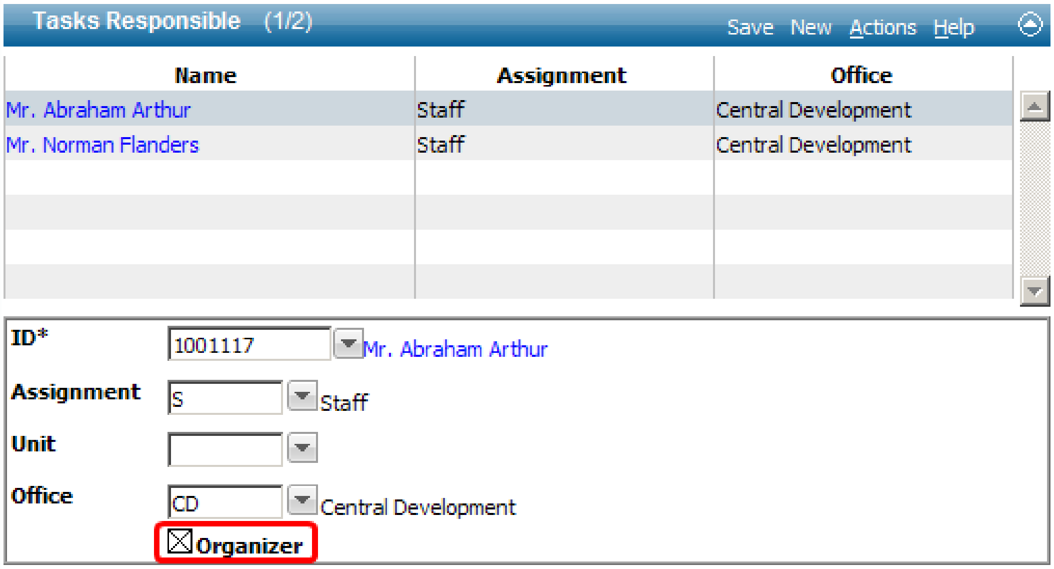
Status*			
Assignment		Office	
Contact Rpt			
	□ Strategy	Recur Month	
Frequency		Recur Week	
Purpose		Recur Day	
Source			
Unit	L Law School		

Figure 20: Outlook Calendar Sync Type

The Calendar synchronization option should be used if you want to add an appointment for yourself to your calendar or if you want to schedule a meeting and invite others to it.

Meeting Organizer

The person creating the Advance Web task is identified as the meeting organizer in both Outlook and Advance Web. This is indicated by the Organizer field that shows on the Task Responsible form for tasks synchronized as Outlook calendar entries. This field is display only and is maintained automatically by Advance Web. You cannot change this value.



Name	Assignment	Office
Mr. Abraham Arthur	Staff	Central Development
Mr. Norman Flanders	Staff	Central Development

ID*	1001117	Mr. Abraham Arthur
Assignment	S	Staff
Unit		
Office	CD	Central Development
	☒ Organizer	

Figure 21: Tasks Responsible - Organizer check box

If the meeting organizer does not assign him/herself as the initial Responsible ID in the task form, Advance Web will automatically create a Task Responsible record for this person, marking them as the organizer. This calendar entry also shows up automatically on the organizer's Outlook calendar.

Anyone 'invited' to this meeting in Advance Web who is not the organizer (on the Task Responsible form for the task), will automatically receive an e-mail invitation in Outlook to the meeting. This is consistent with the notification method used by Outlook when someone is invited to a meeting. The responses made to this invitation in Outlook do not impact the task responsible assignments in Advance Web, these are only for Outlook purposes.

Calendar Modifications

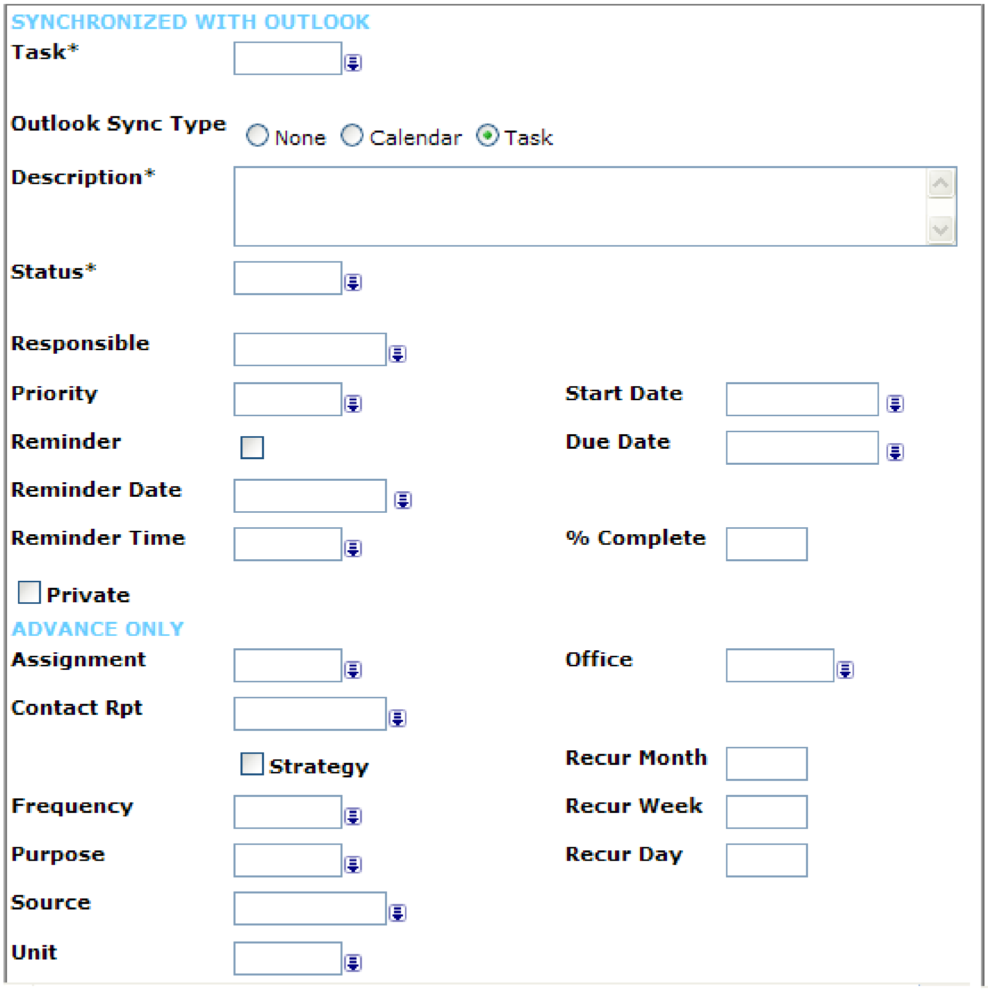
Modifications made to Calendar entries in Outlook that originated from Advance Web can only be made by the 'Organizer'. Advance Web will restrict non-organizers from modifying or deleting the task or any task responsible resources associated with the task.

Private Indicator and Reminder Information

Both the Private and the Reminder information (check box and reminder minutes) are only passed to meeting attendees as part of the initial meeting invitation. However, if the meeting organizer subsequently changes these fields in Advance, these updates will only apply to the organizer's view of this calendar item, not the invited attendees. These are not fields that the organizer can maintain for anyone other than him/herself after the initial invitation is created. This is consistent with standard Outlook behavior.

Outlook Task Creation

When the ‘Task’ radio button is selected as the Outlook Sync Type, the Task form will dynamically change to show the Task version of the task form.



SYNCHRONIZED WITH OUTLOOK

Task*

Outlook Sync Type ○ None ○ Calendar ● Task

Description*

Status*

Responsible

Priority | Start Date

Reminder □ | Due Date

Reminder Date

Reminder Time | % Complete

□ Private

ADVANCE ONLY

Assignment | Office

Contact Rpt

□ Strategy | Recur Month

Frequency | Recur Week

Purpose | Recur Day

Source

Unit

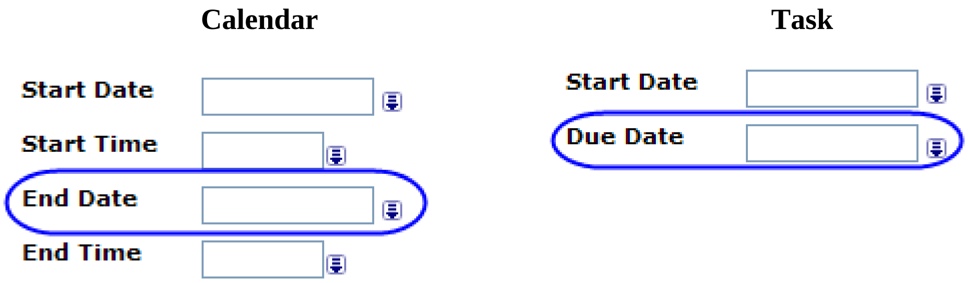
Figure 22: Outlook Task Sync Type

The Task synchronization option should **ONLY** be used if the person entering the task is assigning it to him/herself. This option is intended to support a person creating his/her own To Do or Task list in Outlook. This option will only update Outlook if the *Responsible* value entered is equal to the person actually creating this task. If it is anyone else, it will not synchronize with Outlook.

Once you save your task, a row containing the name of the person selected for the *Responsible* field on the task displays on the ‘Tasks Responsible’ form per standard Advance Web functionality. The task is synchronized with the Exchange server and added to your Outlook Tasks list. Again, this only occurs if the person adding the task is also assigned as the task responsible resource. If additional names are added to the Tasks Responsible form, the task will not be synchronized with Exchange for these other individuals.

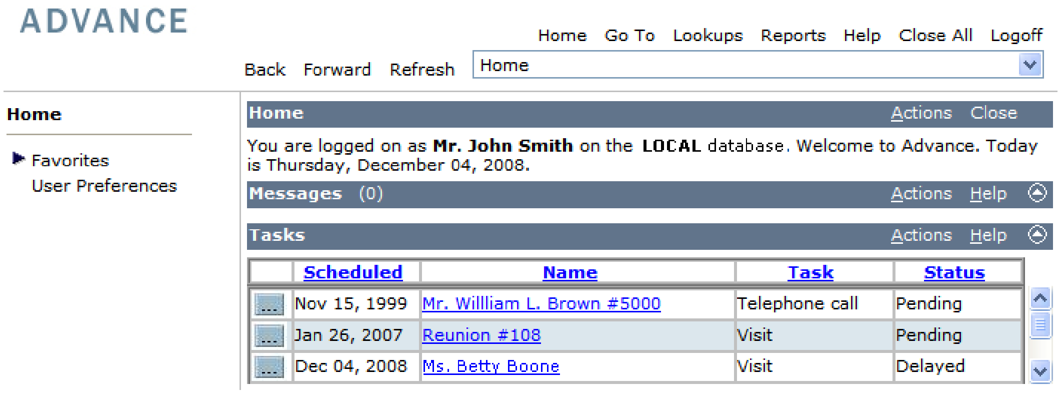
Outlook Tasks on the Home Page

One of the checks performed to determine which tasks should display on the Home Page Tasks form is whether the *Completed* date is populated. If it is, the task does not display on the Home Page Tasks form. However, this is an issue for tasks synchronized with Outlook, as the *Completed* date will typically be populated. This *Completed* date is actually the *End Date* for a task synchronized as an Outlook Calendar and the Due Date for one synchronized as an Outlook Task:



Therefore, the *Completed* date check is not performed for synchronized tasks. Consequently, the only qualifying factor for synchronized tasks is if the task status has the ‘active’ indicator checked in the TMS view. If so, it will display on the Home Page. If it does not, the synchronized task will not display on this form.

Tasks not synchronized with Outlook will display on this form per the existing Advance Web functionality.



You are logged on as **Mr. John Smith** on the **LOCAL** database. Welcome to Advance. Today is Thursday, December 04, 2008.

	Scheduled	Name	Task	Status
...	Nov 15, 1999	Mr. Willliam L. Brown #5000	Telephone call	Pending
...	Jan 26, 2007	Reunion #108	Visit	Pending
...	Dec 04, 2008	Ms. Betty Boone	Visit	Delayed

Figure 23: Tasks displayed on Home Page

Outlook is Unavailable

In the rare instance where Outlook is unavailable, Advance Web will track synchronized task/calendar additions, modifications and deletions in Advance Web, and these will automatically synchronize with Outlook when it does become available. This is handled via the sync_status value in the task_outlook_sync table.

Advance Web is Unavailable

In the rare instance where Advance Web is unavailable, Outlook does not have the ability to indicate to you that Advance Web is unavailable for the web service to perform the update. Therefore, you may wish to implement policies and procedures at your institution or organization to handle this rare scenario. As an example, you may issue an e-mail to all impacted personnel letting them know Advance Web is unavailable. Any changes made in Outlook will need to either be made again in Outlook or made directly to Advance Web, as out-of-sync conditions could be created that cannot be automatically reconciled through Advance Web functionality.

Feature Setup

To properly implement this feature, there are a few setup steps required. These steps are listed in the following sections.

Outlook Impersonation Account

In order to allow Advance Web to communicate properly with the Exchange Server, you must create an Exchange Impersonation Account in Outlook. This Impersonation Account should be given full access to create, edit and delete task and calendar/meeting entries in Outlook. While this will not result in individual Advance Web users having this level of access, it is required to facilitate the correct functioning of this feature and the integration with the Exchange 2007 Web services. Please refer to your Exchange documentation for additional information.

Additionally, the following two system options require setup via the Advance Configuration Utility to indicate the User Name and Password for this Impersonation Account:

System Option 296

Exchange Impersonation Account User Name

The impersonation account in Outlook used to provide Advance Web with the level of access it needs to perform all Outlook synchronization functionality. Account must have full access to create, edit and delete task and calendar/meeting entries in Outlook.

System Option 297

Exchange Impersonation Account Password

This is the password for the impersonation account specified in System Option 296.

Exchange Domain Information

The following two system options provide the necessary information for Advance Web to send the Exchange 2007 updates to the appropriate Exchange server and domain.

System Option 293

Exchange EWS Server Name

This value indicates the name or IP Address of the Exchange Server that synchronizes with Advance, e.g., https://{IP address or name of server}/ews/exchange.asmx.

System Option 294

Exchange Domain Name

This value indicates the name of the Exchange domain controller, e.g., sghe.com, and lets the Exchange web server know to which Exchange domain updates should be communicated.

Task Context Activation

The following system option setting indicates which tasks to synchronize with Outlook.

System Option 295

Synchronized Contexts with Outlook

This system option allows you to specify which Task contexts you wish to synchronize with Outlook. You can select one, multiple, all, or none of these contexts.

- Allocation
- Committee
- Contact Report
- Contract/Grant
- Entity
- Event
- Program Prospect
- Proposal
- Prospect

ⓘ *The default for this system option is all contexts unselected, which indicates Outlook Synchronization is turned off.*

Synchronization Frequency

Updates sent from Advance Web to Outlook occur virtually real-time. However, updates coming back from Outlook to Advance web go through a queuing process. There are two system options used by this process:

System Option 291

Advance Web Synchronization Frequency

The value for this system option indicates the number of minutes Advance Web will wait before checking to see if there are pending updates from Exchange to be synchronized for the logged in user and applied to Advance Web.

Each time this time period passes, the next activity performed by you will result in Advance Web checking to see if there are any pending updates for you from Exchange.

Example: System Option 291 = 5 minutes

- An update notification is received from Exchange for a task for the signed-on user. A record in the task_outlook_sync table has an 'N' in the sync_task row indicating this pending Exchange update exists.
- The Advance Web user navigates through the system, does a refresh, updates a record, i.e. activity occurs for this user within Advance Web.
- The system determines elapsed time since it last checked for Exchange updates for this user.
    - If the elapsed time is 5 minutes or more, this check is performed again. Any pending updates for this user are applied to Advance Web and the sync_status for these updated tasks is changed from N to S.
    - If the elapsed time is less than 5 minutes, this check is not done and any pending updates are not applied to Advance Web.
    - If a user has pending updates and is not logged on to Advance Web, this check will be performed and the updates will be applied upon their next login.

System Option 292

Exchange Notification Frequency

Indicates the number of minutes Exchange will wait before communicating notification of pending updates to Advance. Once updates are queued in Advance, they will remain until user to which changes apply logs in to Advance or value in System Option 291 is reached.

Updates from the Exchange Server are queued on the Exchange Server and only sent to Advance Web at the frequency identified by System Option 292. This system option can be thought of as the timer setting at which notifications of updates from the Exchange Server are sent to Advance Web.

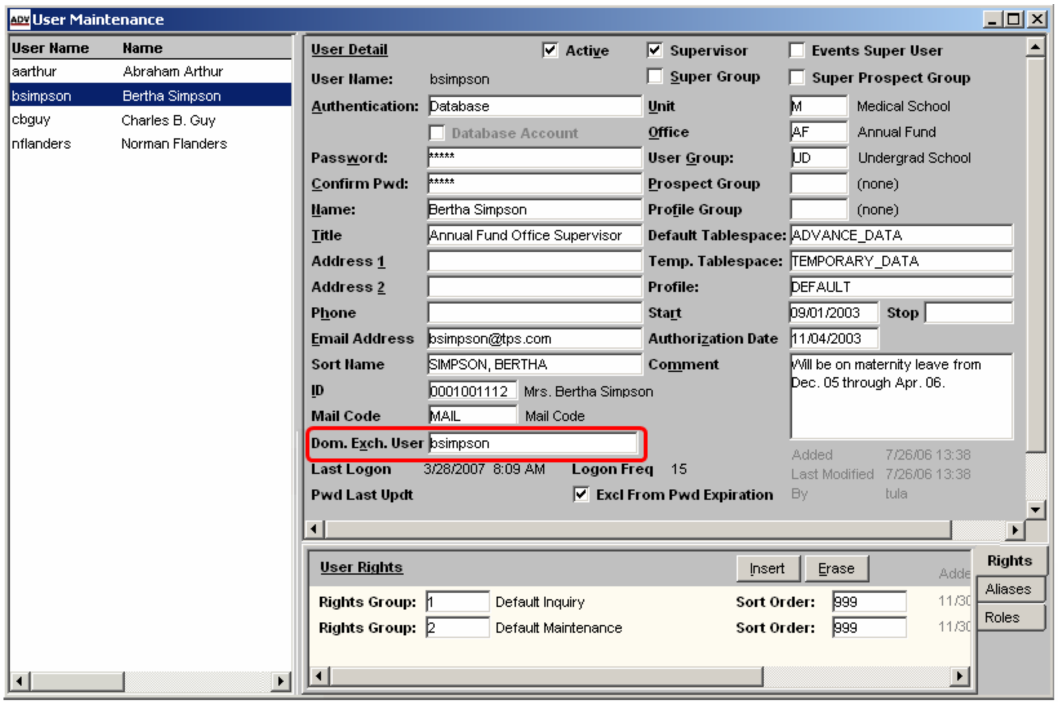
When sent to Advance Web, these are not the actual task and calendar updates but a notification that indicates an update is available on the Exchange Server for a particular task. These pending Exchange updates are identified by records in the task_outlook_sync table having an 'N' in the sync_status column. Advance Web will actually retrieve the updates from Exchange. This process is controlled by System Option 291 described above.

The default value provided for System Option 292 is 2 (minutes). This should be adjusted accordingly based on the needs of your institution or organization.

User Set-Up

If you turn on Outlook Synchronization for one or more Task Contexts (via System Option 295), all Advance Web users who will be adding, modifying or deleting synchronized tasks must have the new 'Domain Exch User' field populated in the Configuration Utility with their Exchange User Name. Otherwise, updates to Exchange will not occur, as the appropriate user account information will not be available.

Values entered in this field are stored in zz_user.domain_exchange_user_name.



User Name	Name
aarthur	Abraham Arthur
bsimpson	Bertha Simpson
cbguy	Charles B. Guy
nflanders	Norman Flanders

User Detail ☑ Active ☑ Supervisor ☐ Events Super User

User Name: bsimpson ☐ Super Group ☐ Super Prospect Group

Authentication: Database

☐ Database Account

Password: *****

Confirm Pwd: *****

Name: Bertha Simpson

Title: Annual Fund Office Supervisor

Address 1

Address 2

Phone

Email Address: bsimpson@tps.com

Sort Name: SIMPSON, BERTHA

ID: 0001001112 Mrs. Bertha Simpson

Mail Code: MAIL Mail Code

Dom. Exch. User: bsimpson

Last Logon: 3/28/2007 8:09 AM Logon Freq 15

Pwd Last Updt ☑ Excl From Pwd Expiration

Unit: M Medical School

Office: AF Annual Fund

User Group: UD Undergrad School

Prospect Group: (none)

Profile Group: (none)

Default Tablespace: ADVANCE_DATA

Temp. Tablespace: TEMPORARY_DATA

Profile: DEFAULT

Start: 09/01/2003 Stop

Authorization Date: 11/04/2003

Comment: Will be on maternity leave from Dec. 05 through Apr. 06.

Added 7/26/06 13:38

Last Modified 7/26/06 13:38

By tula

User Rights

Rights Group: 1 Default Inquiry Sort Order: 999 11/30

Rights Group: 2 Default Maintenance Sort Order: 999 11/30

Figure 24: Configuration Utility - User Maintenance

To specify Domain Exchange User Names:

6. Log in to the Advance Configuration Utility.
7. If prompted, confirm you are authorized to maintain user information by entering your Advance Security Officer username and password, then, click OK.
8. When the Users Security window appears, it displays all existing users in the Users pane on the left side of the window.
9. Scroll through the list on the left and highlight the User Name you wish to modify..
10. Enter the Domain Exchange User value in the User Detail pane on right side of the window.
11. Click Save.
12. Repeat steps 1-6 for each applicable user.

Web Configuration File

The Web Configuration file (web.config) is located on the Web Server where Advance Web is installed. For example: C:\Inetpub\wwwroot\AdvanceWeb\Web.config.

This file contains three keys for this functionality, which require modification using any text editor, such as Notepad.

The three keys are as follows and can be found in the section of the Web.Config file labeled “Task Synchronization with Exchange Server”:

```
        <add key="ClientNotificationURL"
        value="http://advancewebserver/advancevirdir/PushNotificationClie
        nt.asmx" />
```

This is the Advance Web service that Exchange calls after a modification is made in Exchange to a synchronized task or calendar entry. This path represents the directory structure where the Advance web service is located.

where:

- **advancewebserver** should be changed to the name of your Web Server where Advance Web is installed.
- **advancevirdir** should be changed to the directory where Advance Web is installed on the Web Server.

🛠 *The ‘PushNotificationClient.asmx’ file runs dynamically when Advance Web starts and is not a physical file.*

```
        <add key="ExchangeTestCredentials" value="1" />
```

A value of 0 indicates use in a production environment using the domain qualifier and the user’s credentials are used. A value of 1 indicates use in a test environment using the local qualifier and the Exchange test user’s credentials are used.

```
        <add key="ExchangeTimersOn" value="0" />
```

A value of 0 indicates the exchange timer is disabled and the values in the Exchange system options are not examined. A value of 1 indicates the timer is enabled and the values in the Exchange system options are examined.

🛠 *You must change this from 0 to 1 in order to activate this feature.*

TMS Views and Table Setup

The following TMS View and Database Tables can be maintained directly via Advance Web after the installation is complete.

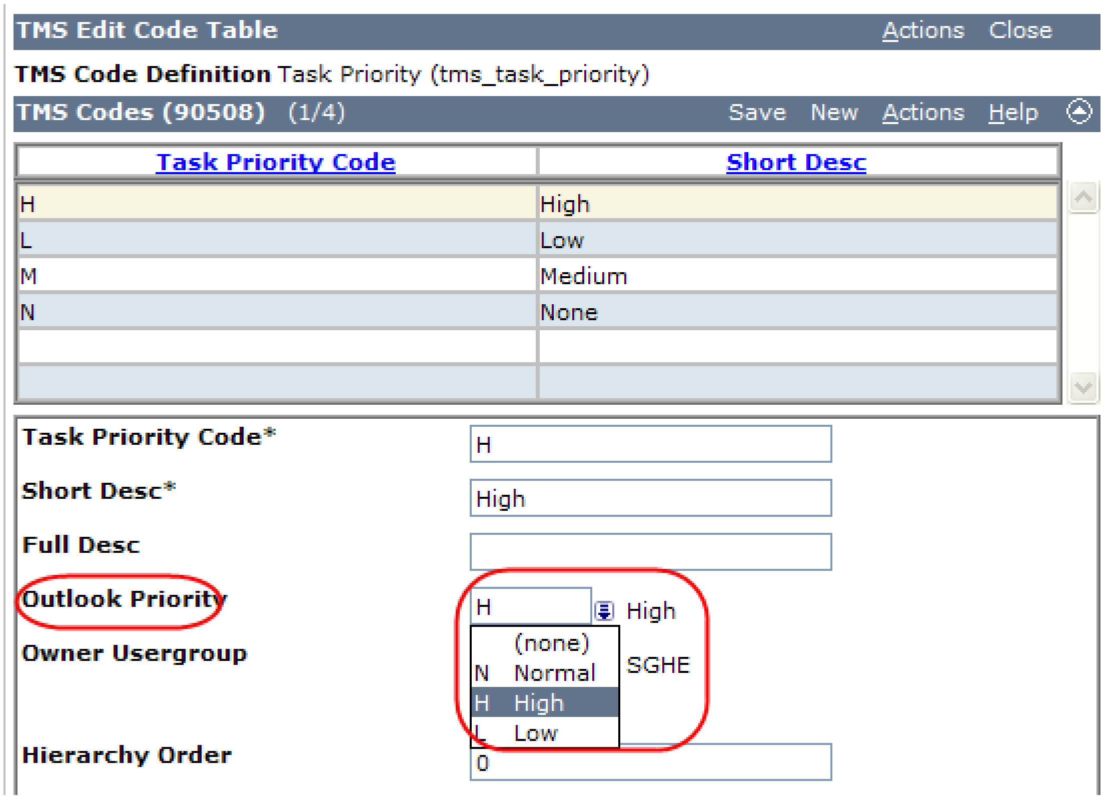
tms_task_priority

Outlook has three valid values for Priority for an Outlook Task or Calendar Entry: Normal, High and Low. Therefore, institutions that turn on the Outlook Integration feature must have three records in the Outlook Priority field of tms_task_priority with the following:

- One (and only one) must have **L** selected to indicate it is equivalent to the Low Priority in Outlook.
- One (and only one) must have **N** selected to indicate it is equivalent to the Normal Priority in Outlook.
- One (and only one) must have **H** selected to indicate it is equivalent to the High Priority in Outlook.

Values in the Outlook Priority drop-down list are system-provided and cannot be modified.

If a task is synchronizing with Outlook, the drop-down for the Priority field on the Task form is restricted to these three Outlook priority values. It is critical that these three values be identified correctly in tms_task_priority for tasks to synchronize correctly with Exchange.



TMS Code Definition Task Priority (tms_task_priority)

Task Priority Code	Short Desc
H	High
L	Low
M	Medium
N	None

Task Priority Code*	H
Short Desc*	High
Full Desc	
Outlook Priority	H High
Owner Usergroup	SGHE
Hierarchy Order	0

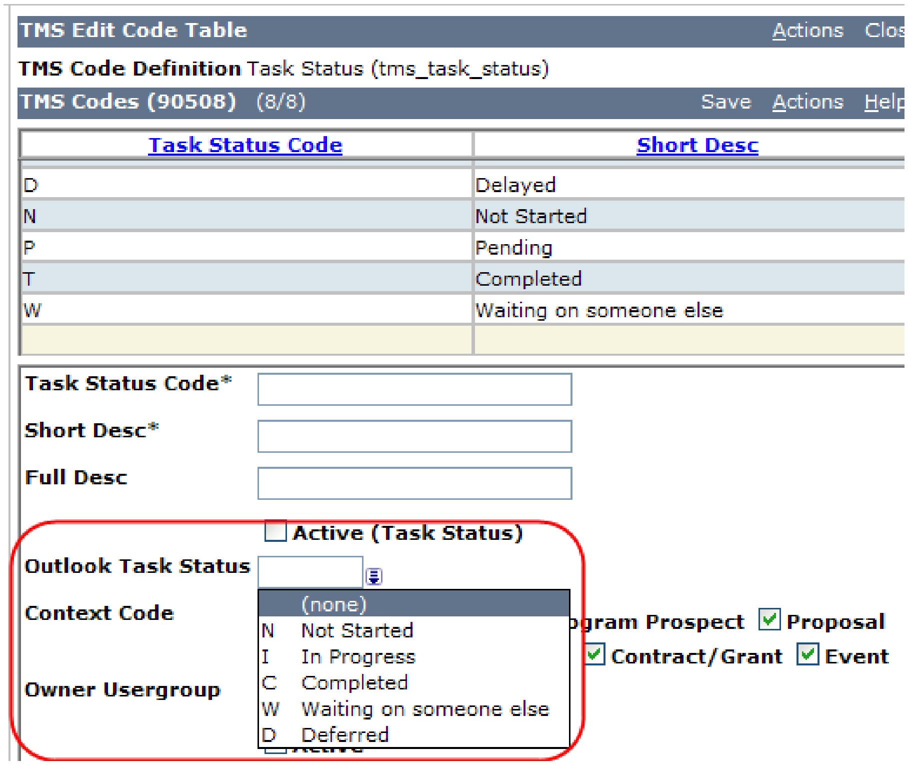
Figure 25: TMS Code Definition - Task Priority

tms_task_status

Outlook has five valid values for Status for an Outlook Task or Calendar Entry. Therefore, institutions that turn on the Outlook Integration feature must have five records in the Outlook Task Status field of tms_task_status with the following:

- One (and only one) must have **N** selected to indicate it is equivalent to the *Not Started* status in Outlook.
- One (and only one) must have **I** selected to indicate it is equivalent to the *In Progress* status in Outlook.
- One (and only one) must have **C** selected to indicate it is equivalent to the *Completed* status in Outlook.
- One (and only one) must have **W** selected to indicate it is equivalent to the *Waiting on someone else* status in Outlook.
- One (and only one) must have **D** selected to indicate it is equivalent to the *Deferred* status in Outlook.

If a task is synchronizing with Outlook, the drop-down for the Status field on the Task form is restricted to these five Outlook values. It is critical that these five values be identified correctly in the tms_task_status for tasks to synchronize correctly with Exchange.



TMS Code Definition Task Status (tms_task_status)

(8/8)

Task Status Code	Short Desc
D	Delayed
N	Not Started
P	Pending
T	Completed
W	Waiting on someone else

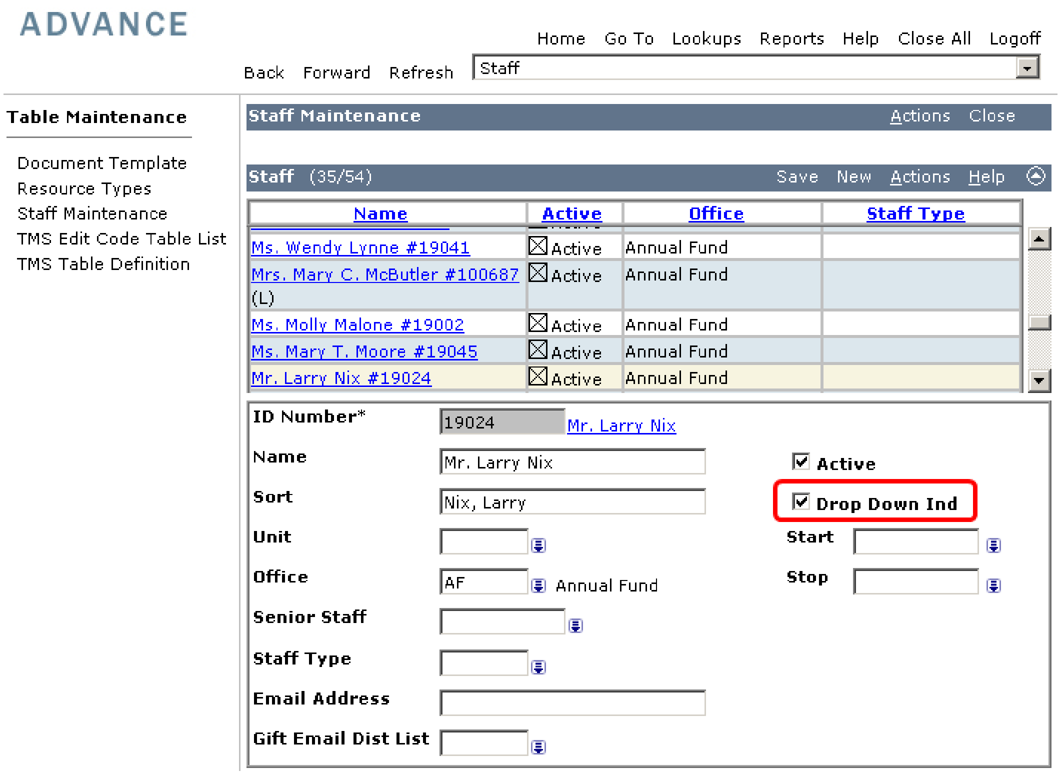
Task Status Code*	
Short Desc*	
Full Desc	
	□ **Active (Task Status)**
Outlook Task Status	
Context Code	**gram Prospect** ☑ **Proposal** ☑ **Contract/Grant** ☑ **Event**
Owner Usergroup	

```
   (none)
N  Not Started
I  In Progress
C  Completed
W  Waiting on someone else
D  Deferred
```

Figure 26: TMS Code Definition - Task Status

staff

In order to ensure the appropriate staff members are available for selection when using this functionality, confirm the 'Drop Down Ind' is selected for each staff member available for task assignment.



Staff

Name	Active	Office	Staff Type
Ms. Wendy Lynne #19041	☒Active	Annual Fund	
Mrs. Mary C. McButler #100687 (L)	☒Active	Annual Fund	
Ms. Molly Malone #19002	☒Active	Annual Fund	
Ms. Mary T. Moore #19045	☒Active	Annual Fund	
Mr. Larry Nix #19024	☒Active	Annual Fund	

ID Number* 19024 Mr. Larry Nix

Name Mr. Larry Nix

Sort Nix, Larry

Unit

Office AF Annual Fund

Senior Staff

Staff Type

Email Address

Gift Email Dist List

☑ Active

☑ Drop Down Ind

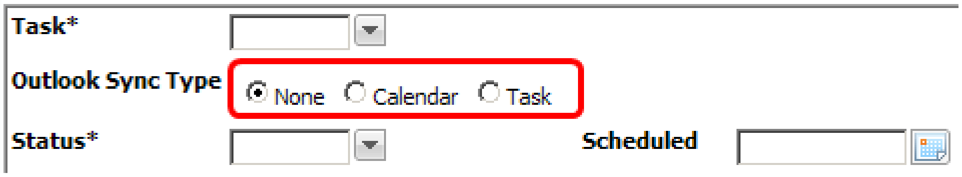
Start

Stop

Figure 27: Staff Maintenance

tms_outlook_sync

This view contains three system-required values that act as radio buttons and allow you to select the type of Outlook Synchronization.



Task*

Outlook Sync Type ● None ○ Calendar ○ Task

Status*

Scheduled

Figure 28: tms_outlook_sync

Although you can modify attributes of these codes, e.g., Description, Active, Owner, etc., Advance does not allow you to delete or add new codes. The New button does not appear on the Form Header when displaying this TMS view and, if you select Delete from the Actions menu, a message indicating system required values may not be deleted will display.

3 Experian™ QAS Pro Web 5.62 Address Verification

This functionality integrates QAS Web Pro from Experian for easy address verification when adding or maintaining addresses. Ensuring constituent's addresses are correct and formatted consistently is not only important, but required in order to prevent mail from being returned as undeliverable. Using the latest data file, QAS Pro Web online address validation software accurately verifies and formats addresses in real time in Advance Web. This mechanism not only prevents errors on the front end while increasing the usefulness of your data, it also enhances your ability to communicate effectively with your constituents.

The correct version of QAS Pro Web (plus QAS for Advance integration files) for Advance Web at your institution or organization must be obtained separately from Experian. Bio Maintenance licensing is required to use this feature.

ⓘ *Advance Web contains logic related to this feature, which is only invoked when you are current on address forms, based on their form IDs. If you have profile-specific changes for the SunGard Higher Education delivered address forms, this logic will be invoked, since the form IDs are identical. However, if you clone one of the address forms, a new form ID is generated and this logic is not available for the new form. If you cloned an address form prior to Advance Web 9.3 and you wish to utilize this functionality, you must re-engineer your cloned changes as profile-specific changes using the current version of the default form.*

Address Verification Types

Two address verification types are available when this functionality is used:

Rapid Addressing

Rapid Addressing, also known as Type Down, is available on forms used by persons whose main responsibility is address maintenance and the address being entered is new. It is available on the Add Person, Add Organization and Addresses forms and is triggered by the following actions.

- Tabbing into Address Line 1 and the address type is address, not phone, email, etc.
- Using the Alt + Y keyboard shortcut while current on an appropriate form.
- Selecting Verify Address from the Actions list on the form header.

When any of these user actions occur, QAS software is triggered and you are presented with a number of screens that allow you to quickly type down and select a valid address. The QAS prompts and workflow are not part of Advance Web and will be decided between your institution or organization and QAS.

Rapid Address is accessed via the following forms:

- Add Organization
- Add Person
- Addresses (New Address)

ⓘ *If there is existing data on the Addresses form in the address fields, you will be allowed to verify that address by modifying it or by selecting an address verification control. These actions will trigger Web Verification and not Rapid Addressing. If no address data exists, the Rapid Addressing will be used.*

<u>Web Verification:</u>

Web Verification is on the Addresses form to be used for verification of existing addresses, but it is also on forms that are used by persons whose main responsibility is <u>not</u> entering address data. This is available on the Entity Update (Home Address & Business Address) forms, which are accessible when System Option 260 is enabled. Web Verification is triggered by the following actions.

- Using the Alt + Y keyboard shortcut while current on an appropriate form.
- Selecting Verify Address from the Actions list on the form header.

When any of these actions occur, the data is passed over to QAS software for verification. If the address entered is an exact match to QAS data, there will be no user interaction with QAS software. Otherwise, you will be prompted by the QAS software to make a selection from a list of possible matches, be allowed to refine the address or have the address accepted as-is. As with Rapid Addressing, the prompts and workflow will be decided between your institution or organization and QAS.

Web Verification is accessed via the following forms:

- Addresses (Existing Addresses)
- Entity Update – Home Address
- Entity Update – Business Address

Initial Setup

Prior to using address verification you must set up your system with TMS values and modify system option settings. Additional server configuration is necessary and your representative at the address verification software manufacturer should be contacted for this part of the setup of your system.

Supported Datasets

QAS Web Pro Version 5.62 supports the following country datasets:

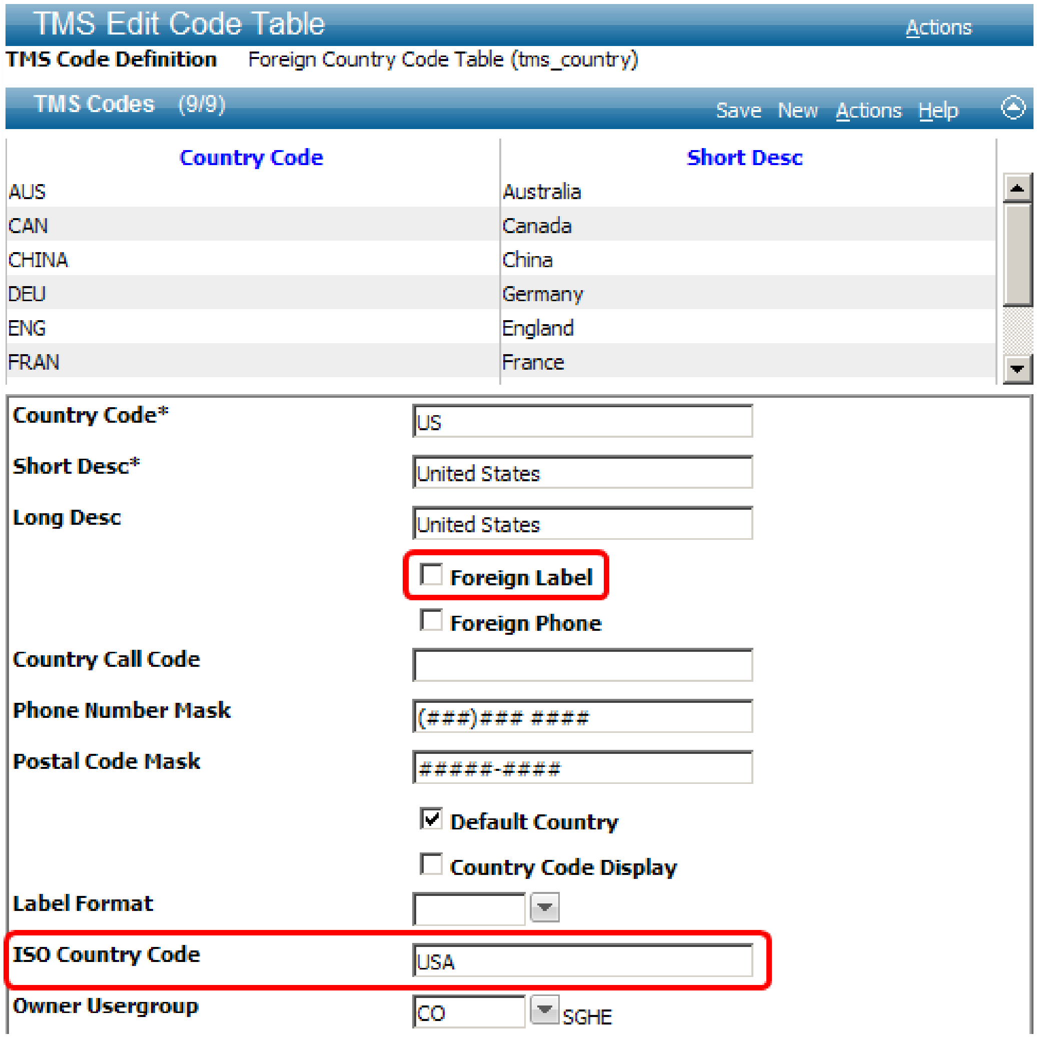
- AUS – Australia
- BEL – Belgium*
- CAN – Canada
- CHE – Switzerland
- DEU – Germany
- DNK – Denmark
- ESP – Spain
- FIN – Finland*
- FRA – France
- GBR – United Kingdom
- IRL – Ireland
- LUX – Luxembourg
- NLD – The Netherlands
- NOR – Norway*
- NZL – New Zealand
- SGP - Singapore
- SWE – Sweden*
- USA – United States of America

** These countries are not supported by Web Verification. You should not enter these codes in the Country Dataset ISO code field in tms_country. This will ensure that the QAS Country Dataset is not called with Web Verification.*

TMS Country Settings

The ISO Country Code field on the Foreign Country Code (tms_country, AE) TMS view must be populated with the appropriate code for each country, e.g., ‘USA’ for the United States of America, as shown in the previous list of datasets. You should only populate the ISO Country Code field for the Country datasets you have purchased.

The Foreign Label check box should be unchecked for countries with an ISO Country Code. Selecting a value from the Label Format drop-down list is optional.



TMS Code Definition Foreign Country Code Table (tms_country)

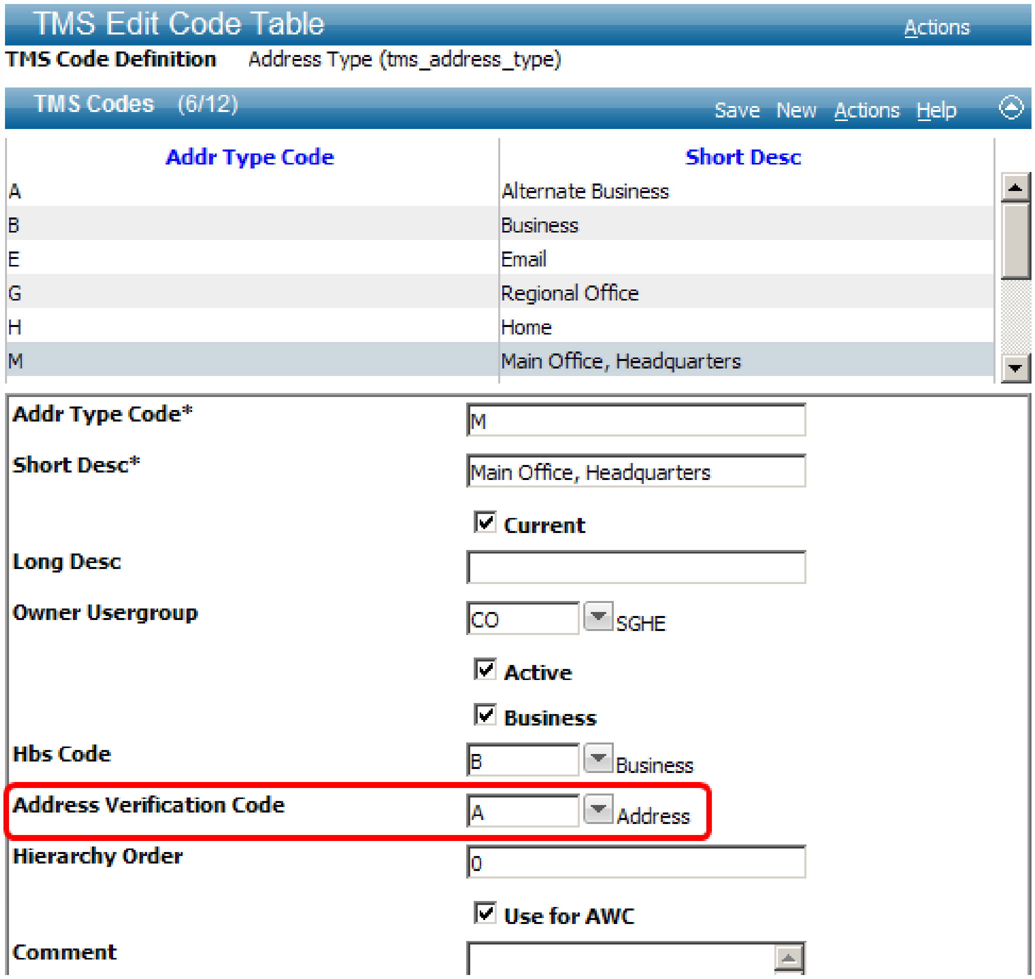
Country Code	Short Desc
AUS	Australia
CAN	Canada
CHINA	China
DEU	Germany
ENG	England
FRAN	France

Country Code*	US
Short Desc*	United States
Long Desc	United States
	□ Foreign Label
	□ Foreign Phone
Country Call Code	
Phone Number Mask	(###)### ####
Postal Code Mask	#####-####
	☑ Default Country
	□ Country Code Display
Label Format	
ISO Country Code	USA
Owner Usergroup	CO SGHE

Figure 29: TMS Code Definition - Foreign Country Code

TMS Address Type Settings

The Address Verification Code field on the Address Type (tms_address_type, AH) TMS view is populated with 'A' on the address type on which you wish to have address verification performed. Address types with this value blank will not have address verification functionality.



TMS Code Definition Address Type (tms_address_type)

TMS Codes (6/12)

Addr Type Code	Short Desc
A	Alternate Business
B	Business
E	Email
G	Regional Office
H	Home
M	Main Office, Headquarters

Addr Type Code*	M
Short Desc*	Main Office, Headquarters
	☑ **Current**
Long Desc	
Owner Usergroup	CO SGHE
	☑ **Active**
	☑ **Business**
Hbs Code	B Business
Address Verification Code	A Address
Hierarchy Order	0
	☑ **Use for AWC**
Comment	

Figure 30: TMS Code Definition - Address Type

System Option Settings

Two system options affect use of address verification:

System Option 310

Address Verification Software

This system option indicates whether you wish to enable Address Verification Software.

Y — Enabling this system option will activate functionality built to allow Advance Web to interface with 3rd party address verification software. If this system option is enabled, Advance Web's Zip and Vanity City functionality is disabled.

♦ N — When this system option is not enabled ('No'), the address verification functionality is not available, accelerator key (ALT + Y) is not active and the Verify Address selection in the Actions drop-down list is not displayed on the appropriate forms.

System Option 311

Address Verification Software – Accept County Codes

Indicates that Advance Web's interface to address verification software will return and populate County Code. The county codes returned are Federal Information Processing Standard (FIPS) codes for the US only.

Y — When a value of 'Y' is specified, this system option is enabled and the address verification software will return and populate County Codes. This system option should only be enabled when System Option 310 is enabled

♦ N — When a value of 'N' is specified, this system option is disabled and address verification software will not return and populate County Codes.

If System Option 311 is enabled, you must set the State FIPS Code for states and territories, e.g., DC, PR, in tms_states and have the 5 digit County Code populated in TMS_County. A script is not provided to populate the five digit County Code in tms_county.

ⓘ *The FIPS County Code is a five-digit Federal Information Processing Standard (FIPS) code, which uniquely identifies US counties (http://en.wikipedia.org/wiki/FIPS_state_code has a list of FIPS State Codes). The five-digit FIPS code is comprised of a three-digit code for a county in a state, prefixed by that state's two-digit state code. When System Option 311 is enabled, the address verification software returns a three-digit FIPS county code. This code is not unique, because it does not include the leading two-digit FIPS state code. For example, the three-digit county code 001 could represent Barnstable County in Massachusetts or any other county in the US designated by a state as 001. To make this code unique, Advance Web takes the verified address's state abbreviation, e.g., MA, matches it to the state's two-digit FIPS code stored in tms_state, e.g., MA = 25, and appends it to the three-digit county code, e.g., 25001. The County Code is now unique and is validated against the five digit FIPS code that is stored in tms_county.*

Additional Setup Steps

The key ‘com.qas.proweb.serverURL’ in your web.config file needs to contain the address of the QAS server. Also, the file properties.aspx in the Advance Web directory needs to be updated with the address of the QAS server. If neither of these contains the address for the QAS server, QAS will not work in Advance Web. Both of these need to be updated after each Advance Web upgrade since the settings do not get copied over from the previous version.

Dataset configuration

1. After installing QAS, all datasets must be configured using the QAS Configuration Editor (not the Advance Configuration Utility) to add the fourth line of address and county code.
2. QAS will deliver Rapid Address to you with all countries available for selection in a QAS defined drop down list box. It is your responsibility to work with QAS to ensure that countries for which you do not have licenses are removed from this drop down list. If this is not done, and users select countries for which there is no associated country dataset, the country returned to Advance Web will be “undefined” and could cause unpredictable behavior.

Zip Suffix

There is an option to populate the zip suffix field on an address with the ‘delivery point’ value from the QAS data. This ability is turned on by configuring the QAS server using the following steps:

1. Open the QAS Configuration Editor.
2. In the appropriate address layout (usually Database Format) open the DataPlus branch.
3. Select the line labeled <end> and click Insert Line.
4. Click Add Item and select Postnet barcode->DPC.
5. Save the configuration.
6. Restart the QAS server.

Address Verification Activation Setup Summary

To recap the aforementioned setup steps for activating address verification:

Activate Address Verification:

1. Enable System Option 310
2. TMS_Country – Populate ‘ISO Country Code’ for the purchased country datasets.
3. TMS_Address_Type – Set the Address Verification Code to A on the appropriate address types.

Activate County Code:

1. Enable System Option 311
2. TMS_States – Populate the FIPS State Code by executing the provided script.
3. TMS_County – Populate County code with a five digit FIPS Code (i.e. 25001, 25 = MA, 001 = Barnstable County).

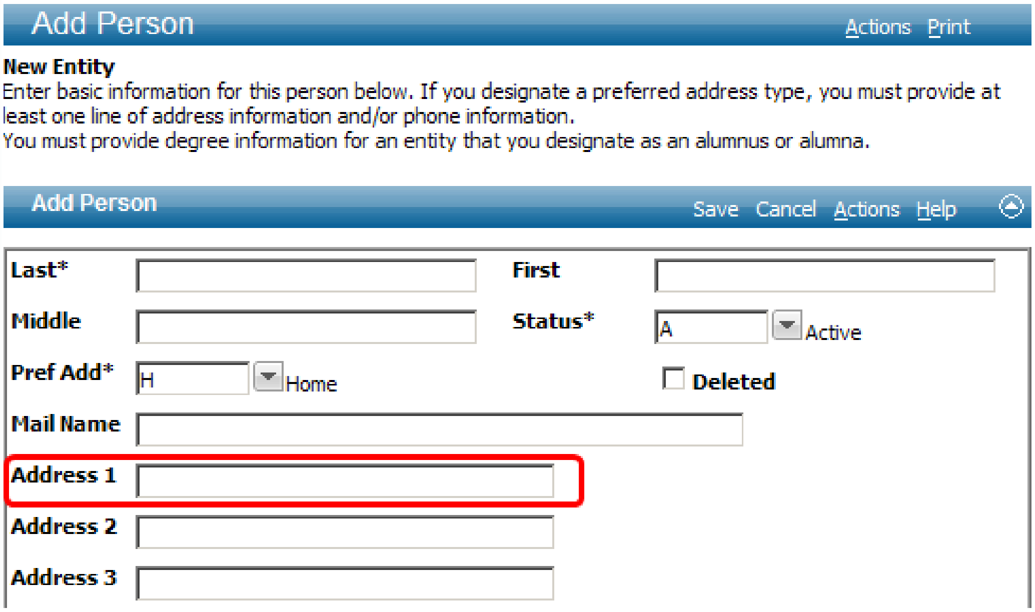
Using Rapid Address

With your system configured to use Rapid Address, you can use this functionality to enter a new person, as the example below demonstrates. This functionality works in the same manner when entering an address for a new organization and a new address for an existing entity.

To enter a New Person’s address using Rapid Address:

1. While current on a form which displays ‘Add Person’ on the page tree, click Add Person.
2. On the Add Person form, enter at least the required information, i.e., Last name, Status and Preferred Address, using the Tab key to navigate to each field. Tab to the ‘Address 1’ field.

   ⓘ *You may also use one of the other methods mentioned earlier, e.g., Alt+Y, to invoke the Rapid Address window.*



Add Person

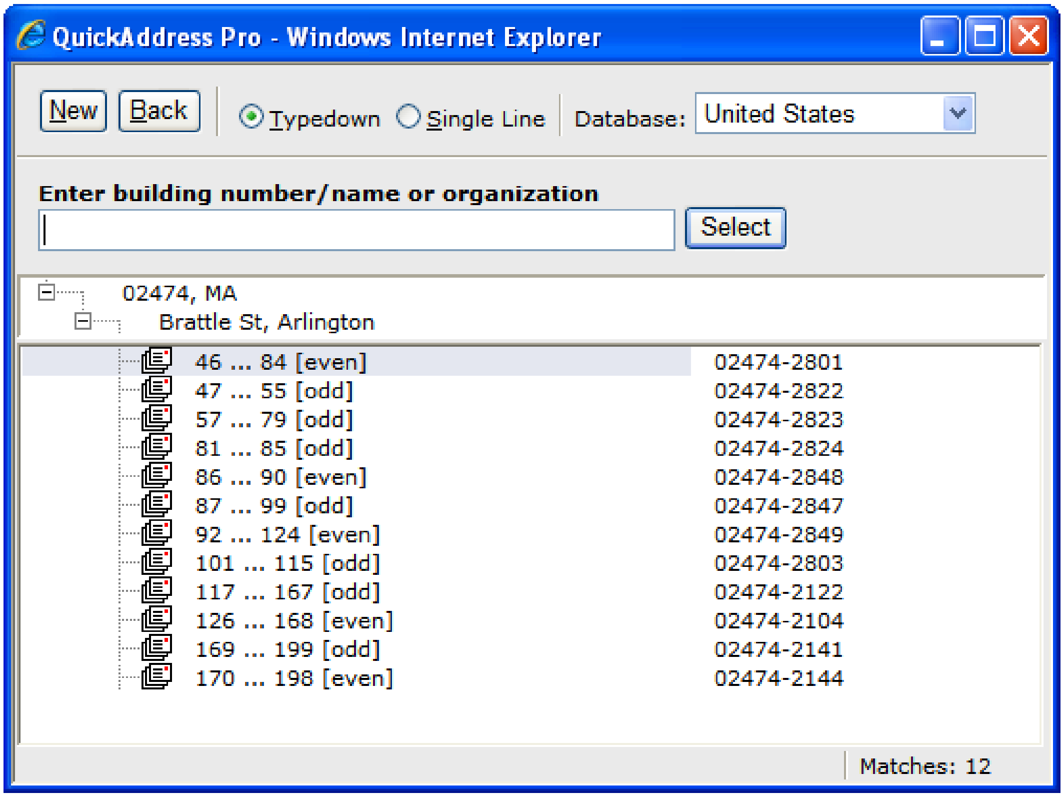
New Entity
Enter basic information for this person below. If you designate a preferred address type, you must provide at least one line of address information and/or phone information.
You must provide degree information for an entity that you designate as an alumnus or alumna.

Add Person

Last*		**First**	
Middle		**Status***	A Active
Pref Add*	H Home		□ **Deleted**
Mail Name			
Address 1			
Address 2			
Address 3			

Figure 31: Adding a person entity

3. The Rapid Address window displays, prompting you to begin entering address information. As you enter and make the selections from the address suggestions presented, the address forms.



QuickAddress Pro - Windows Internet Explorer

New | Back | ● Typedown ○ Single Line | Database: United States

Enter building number/name or organization

Select

02474, MA
Brattle St, Arlington

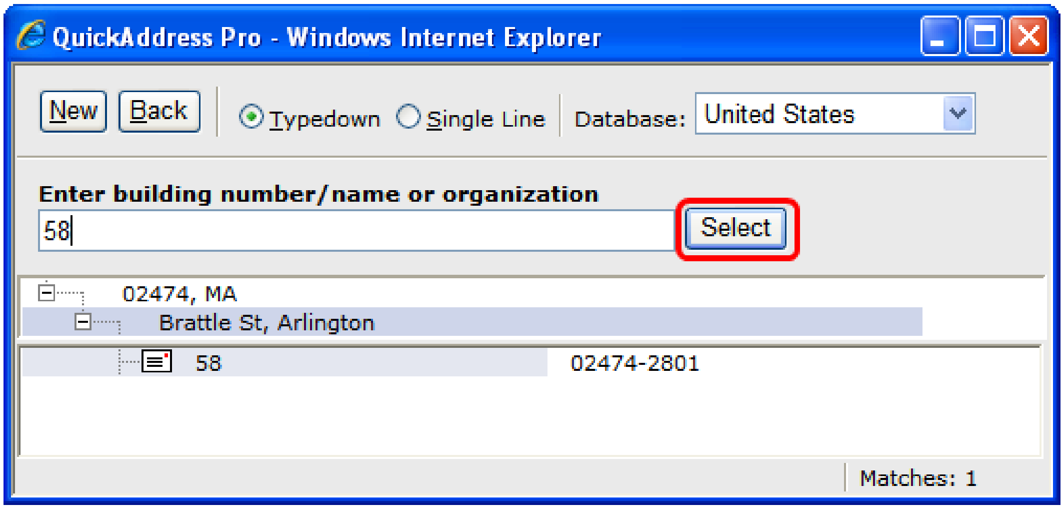
46 ... 84 [even]	02474-2801
47 ... 55 [odd]	02474-2822
57 ... 79 [odd]	02474-2823
81 ... 85 [odd]	02474-2824
86 ... 90 [even]	02474-2848
87 ... 99 [odd]	02474-2847
92 ... 124 [even]	02474-2849
101 ... 115 [odd]	02474-2803
117 ... 167 [odd]	02474-2122
126 ... 168 [even]	02474-2104
169 ... 199 [odd]	02474-2141
170 ... 198 [even]	02474-2144

Matches: 12

Figure 32: Rapid Address window

ⓘ *If the Rapid Address service is not available, you can still add address information and the service will respond when it does become available. For additional information, please contact your vendor representative.*

4. When you finish entering the last part of the address, click Select or Enter to confirm.



QuickAddress Pro - Windows Internet Explorer

New | Back | ● Typedown ○ Single Line | Database: United States

Enter building number/name or organization

58

Select

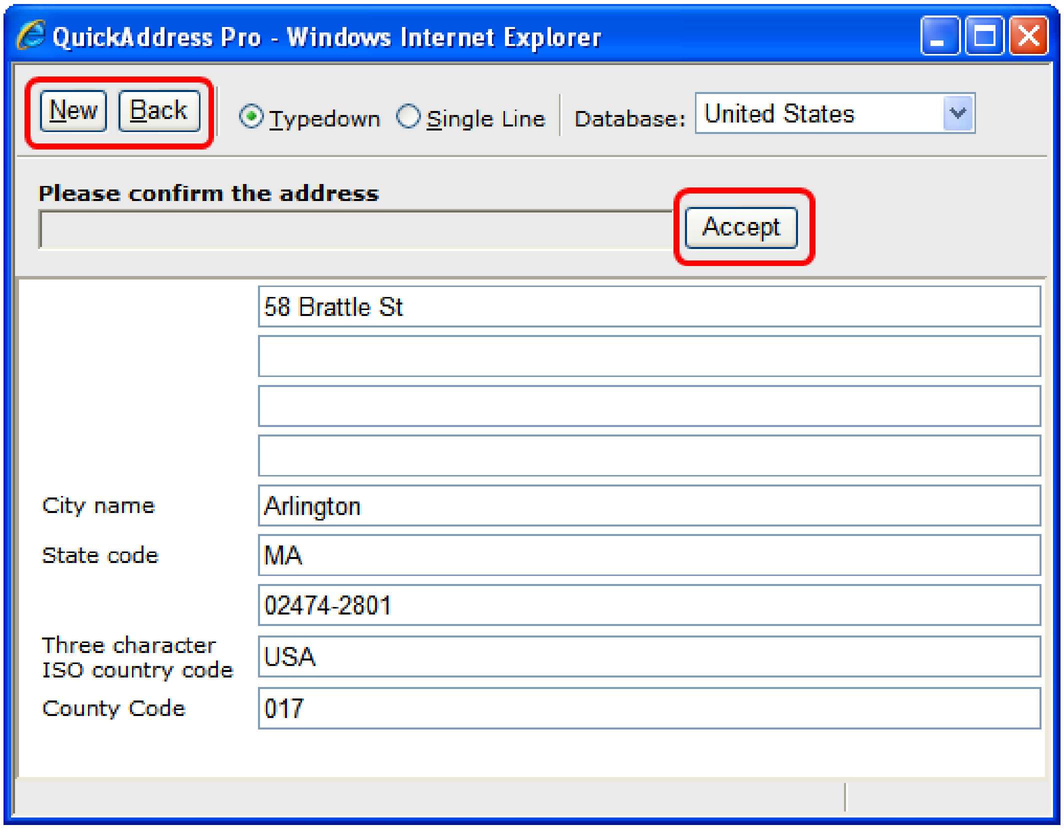
02474, MA
Brattle St, Arlington

58	02474-2801

Matches: 1

Figure 33: Confirm the address

5. If the information is correct, click Accept or Enter. If you wish to return to a previous screen to make a different selection, click Back. If you wish to discard the address and start over, click New.



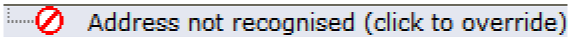
● Typedown ○ Single Line Database: United States

Please confirm the address

	58 Brattle St
City name	Arlington
State code	MA
	02474-2801
Three character ISO country code	USA
County Code	017

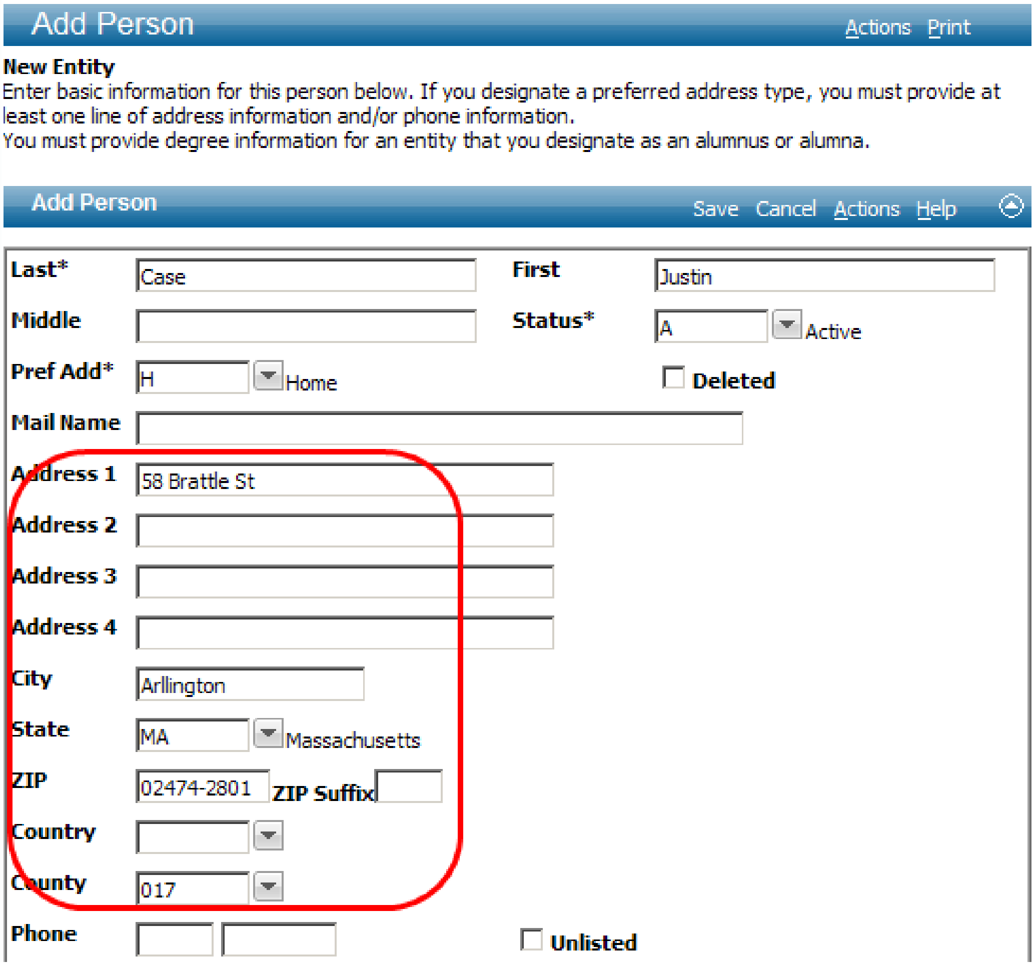
Figure 34: Accept the address

ⓘ *If the address you are trying to enter is not recognized by the system as a valid address, an error message displays indicating that you need to either enter correct information or override the error and accept the unrecognized address:*



Address not recognised (click to override)

6. When you accept the address, you are placed back on the New Person form with the address information completed. Continue entering additional information and Save your changes.



New Entity

Enter basic information for this person below. If you designate a preferred address type, you must provide at least one line of address information and/or phone information.

You must provide degree information for an entity that you designate as an alumnus or alumna.

Last*	Case	**First**	Justin
Middle		**Status***	A Active
Pref Add*	H Home		□ **Deleted**
Mail Name			
Address 1	58 Brattle St		
Address 2			
Address 3			
Address 4			
City	Arllington		
State	MA Massachusetts		
ZIP	02474-2801 **ZIP Suffix**		
Country			
County	017		
Phone		□ **Unlisted**	

Figure 35: Review and Save the address

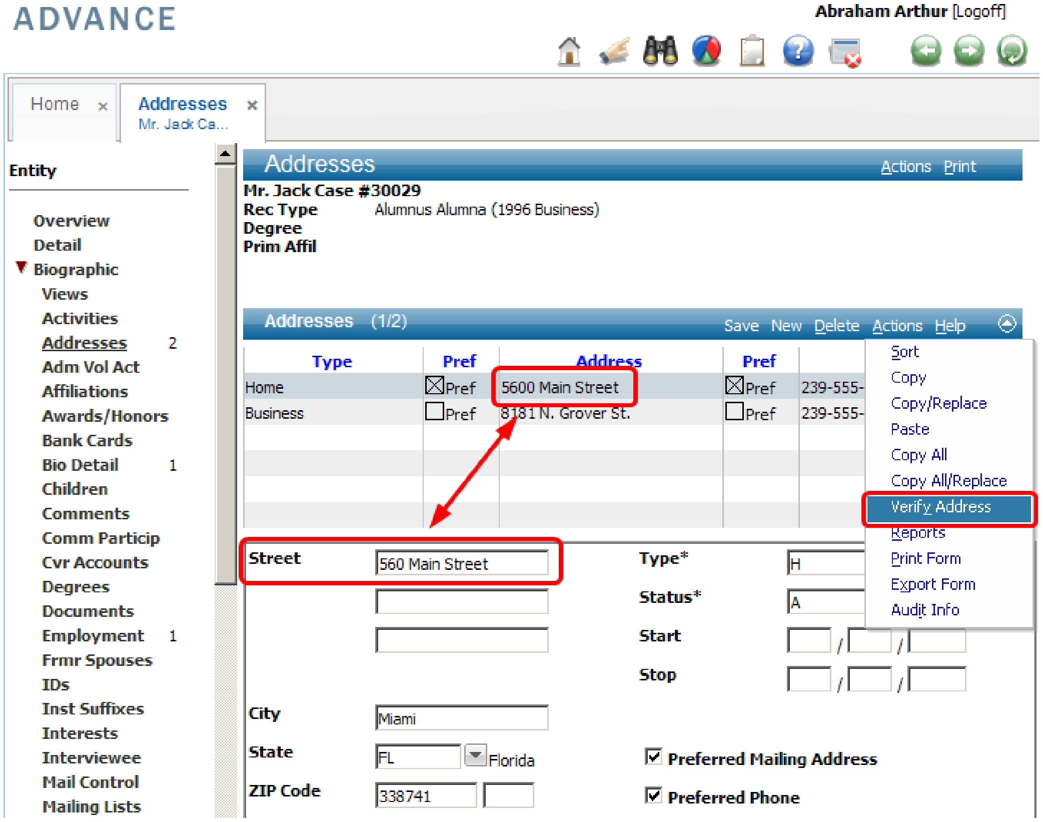
Using Address Verification

With your system configured to use Address Verification, you can use this functionality to update an entity's address, as the example below demonstrates. This functionality works in the same manner when using Entity Update for changes to a home or business address.

To enter an updated address for an entity:

1. While current on an entity's Addresses form, make the appropriate changes to the address, e.g., in the figure below, 5600 is being changed to 560. Tab to another address field and select 'Verify Address' from the Actions list.

   ⓘ *You must also tab to another field if you wish to use the Alt+Y shortcut or you are saving your change.*



Mr. Jack Case #30029
Rec Type Alumnus Alumna (1996 Business)
Degree
Prim Affil

Type	Pref	Address	Pref	
Home	☒Pref	5600 Main Street	☒Pref	239-555-
Business	□Pref	8181 N. Grover St.	□Pref	239-555-

Street: 560 Main Street

Type*: H

Status*: A

Start: / /

Stop: / /

City: Miami

State: FL Florida

ZIP Code: 338741

☑ Preferred Mailing Address

☑ Preferred Phone

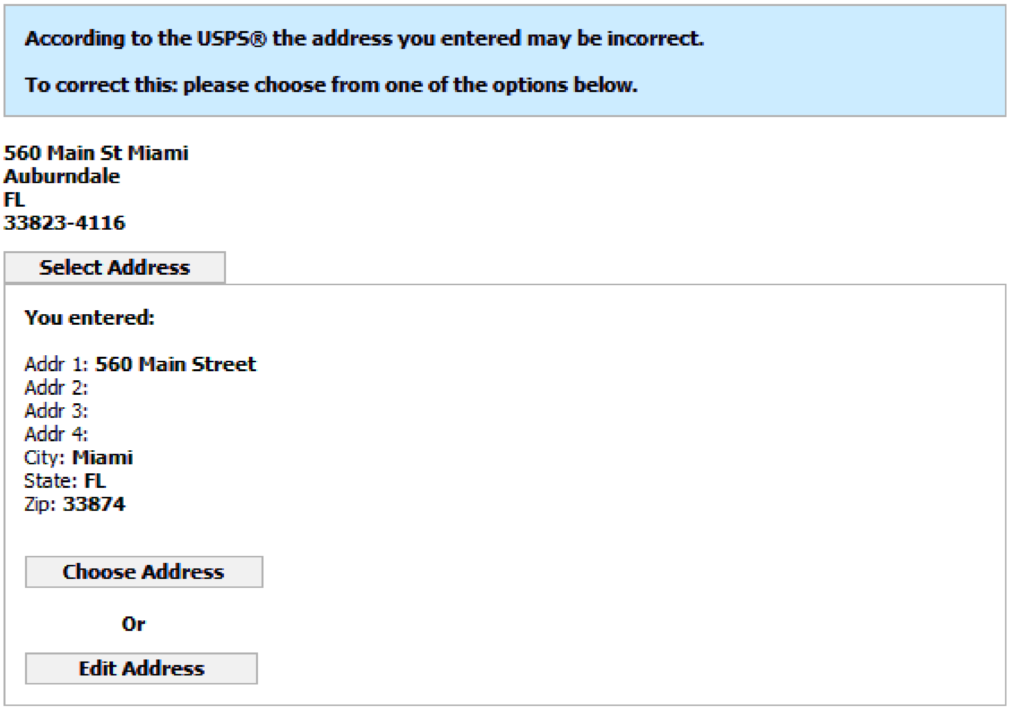
Figure 36: Verify Address command

ⓘ *Address verification is automatically performed when you Save any address changes. If the information entered is valid, no further action is necessary. Otherwise, address verification forms will be presented as shown below.*

If data other than the address data (i.e. phone) is changed, where the address has not been verified, and the record is saved, a warning message displays at the top of the form:

- Address Verification - The address has not been verified. Address Verification can be initiated by selecting ALT + Y or selecting Verify Address from the Actions Menu.

2. The application will verify the information you entered is correct. If the address is valid, the Addresses form will remain displayed and you can Save your changes. If the information is not correct, you will be presented with a valid address option, as shown below.



According to the USPS® the address you entered may be incorrect.

To correct this: please choose from one of the options below.

560 Main St Miami
Auburndale
FL
33823-4116

You entered:

Addr 1: **560 Main Street**
Addr 2:
Addr 3:
Addr 4:
City: **Miami**
State: **FL**
Zip: **33874**

Figure 37: Incorrect address options

ⓘ *If the Address Verification service is not available, you can still modify address information and the service will respond when it does become available. For additional information, please contact your vendor representative.*

3. If there is a recommended address for your entry, you may choose the address by clicking ‘Select Address’. If there is more than one address that may be similar to the address you entered, you may select a link labeled ‘Display all potential matches’.

   You may also accept your entry and return to the Addresses form by clicking **Choose Address** and choose a different address. If you wish to return to the Addresses form and edit the information you entered previously, click **Edit Address**.

4. Save your changes.

4 Banner Student Download

Functionality is provided for institutions and organizations licensed for Banner Student and Advance Web Bio Maintenance to perform a one-way, Banner-to-Advance transfer of data for students, alumni and parents. The data transfer can be in the form of 'initial loads' of information, as well as incremental updates on a continuing basis. In the process of the transfer, a set of tables has data loaded into them. These tables are then used to create DataLoader batches.

During the installation process of Advance Web 9.1.2 and higher, you will be asked whether you wish to install the new adv_banner package. If you answer 'N'o, the adv_banner package will not be installed; if you answer 'Y'es, the adv_banner package will be installed. For additional information, please refer to the *Advance Web Installation Instructions.*

Initial setup of your system

The rest of this section describes modifications, procedures and general information involved with the setup and running of Banner data transfer functionality.

Step 1: Populate the thirdpty_tms_xref table

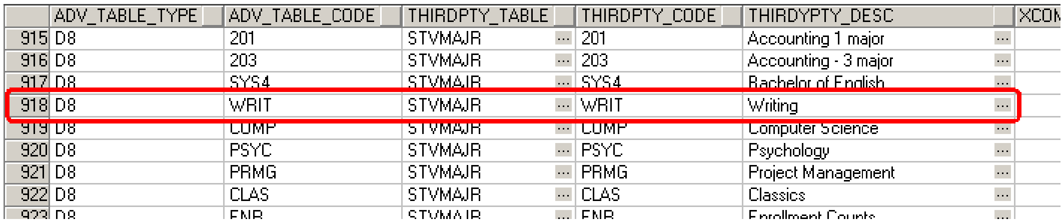
The thirdpty_tms_xref table is used to map the Advance TMS values to your Banner equivalents. This table is empty when it is installed on the database. Use the procedure adv_banner.generate_tms_xref_data to dynamically populate the thirdpty_tms_xref table, then, make the appropriate modifications (as described below).

ⓘ *The procedure makes the assumption Advance codes are the same as their Banner equivalents.*

ⓘ *Once the table is populated, be sure to confirm the accuracy of the table's values before transferring any data.*

It is important to maintain the values in thirdpty_tms_xref as a part of your initial setup by identifying values which differ between Advance and Banner, and updating the table, using your database maintenance tool of choice, with the appropriate Advance values. In some cases, you may need to create new TMS codes in Advance for values that only exist in Banner.

For example, in the figure below showing sample data in thirdpty_tms_xref, the Advance TMS view 'Concentration' has the code 'WRIT', which maps to the code 'WRIT' in Banner. If the Banner code WRIT corresponds to a different value in Advance, you would change the value of adv_table_code to the correct value.



	ADV_TABLE_TYPE	ADV_TABLE_CODE	THIRDPTY_TABLE	THIRDPTY_CODE	THIRDYPTY_DESC	XCO
915	D8	201	STVMAJR	201	Accounting 1 major	
916	D8	203	STVMAJR	203	Accounting - 3 major	
917	D8	SYS4	STVMAJR	SYS4	Bachelor of English	
918	D8	WRIT	STVMAJR	WRIT	Writing	
919	D8	COMP	STVMAJR	COMP	Computer Science	
920	D8	PSYC	STVMAJR	PSYC	Psychology	
921	D8	PRMG	STVMAJR	PRMG	Project Management	
922	D8	CLAS	STVMAJR	CLAS	Classics	
923	D8	ENR	STVMAJR	ENR	Enrollment Counts	

Figure 38: Sample data in thirdpty_tms_xref

ⓘ *If a code is not mapped in this table, when the data transfer from Banner to Advance occurs, the Banner data value will be transferred and preceded by 'B->'. This value will fail DataLoader validation and the value will have to be changed manually or by using 'search and replace'.*

Step 2: Define values in bnr_integration_options and TMS

Next, you need to identify the options to use for bringing Banner data to Advance. These options are defined in the bnr_integration_options table, and are described in the table below, along with instructions for modification:

Option	Label	Description
1	PIDM ID Type*	This option is used to represent Banner PIDMs as Alternate IDs in Advance. This value needs to be added to tms_ids_type code and tms_advloader_alt_ids_code in Advance.
2	Parent ID Type*	This option is used to represent the parents of Banner IDs as Alternate IDs in Advance. The Parent ID is generated by the integration process. This value needs to be added to tms_ids_type_code and tms_advloader_alt_ids_code.
3	Student ID Type*	This option is used to represent the Students and Graduates as Alternate IDs in Advance. This is the current Spriden_id in Banner. This value needs to be added to tms_ids_type code and tms_advloader_alt_ids_code.
4	Data Source Code*	This option is used to represent the data source, as many of the tables being updated have a data_source_code This value needs to be added to tms_advloader_batch_source, tms_entity_source, tms_data_source, tms_change_source, and tms_ethnic_source.
5	Record Type for de-matriculated student *(optional)*	When a student is de-matriculated, a record type change is generated with this value, e.g., 'AL' for alum. Please see the 'Matriculation Status Changes' section for more information.
6	Record Status for de-matriculated student *(optional)*	When a student is de-matriculated, a record status change is generated with this value, e.g., 'I' for inactive. Please see the 'Matriculation Status Changes' section for more information.
7	Solicit_ctrl_code for de-matriculated student *(optional)*	When a student is de-matriculated, the solicit_ctrl_code is set to this value, e.g., 'D' for do not solicit. Please see the 'Matriculation Status Changes' section for more information.

Option	Label	Description
8	Social Security Number ID Type	This option, if populated, is used to bring over Banner SSNs as an Alternate ID. This option should contain the value in tms_ids_type that represents Social Security Number in Advance.
9	Participation Code for Sport	There is no Banner equivalent to the Participation Code in the Advance sport table. Therefore, this value must be supplied here, since it is required. This must be a valid value in tms_sport_particip.
10	Participation Code for Student Activities	There is no Banner equivalent to the Participation Code in the Advance student_activity table, therefore, this value must be supplied here. Must be a valid value in tms_student_particip.
11	Obit Code for Deceased Individuals	This is the value to put in the death_obit_code field in Entity Birth Death data when someone who is already in Advance becomes deceased in Banner, e.g., 'CS' for Clipping Service. Must be a valid value in tms_obituary.
12	Parent Child Relationship Code	This option is used to represent a parent-child relationship. This code is used when creating Children records for Parent loads. This is the tms_relationship code to represent parent-child relationships in Advance, e.g., 'PC' for parent-child.
13	Source of Parent data	This value should be set to 'E' for Emergency Contact' or 'G' for Guardian. When this value is set to 'E', the emergency contact in Banner is used to create the parent data. When this value is set to 'G', the guardian information in Banner is used to create the parent data.
14	User Group for zz_adv_loader _entity	This value is required and used by zz_adv_loader_entity. It is recommended that this user exist in zz_user_group.
15	Advance Email Address Type	This is the Advance address type to assign to the email address brought over from Banner. Only the preferred email is brought over. If this option blank, no email address will be brought over.
16	Default Record Status	This is the default record status for new entities. This value must exist in tms_record_status, e.g., 'A' for active.

Option	Label	Description
17	Advance User Interface - (W)indows or Web (B)rowser	‘B’ is the only valid value for this option at this time. 🛠 *The process needs to know whether it is dealing with Advance Web DataLoader or Windows DataLoader. The tables associated with DataLoader Batch Type determine whether the particular type of data gets processed.*
18	Matriculation Status Logic: (E)nrolled, (R)egistered or space	This option can have one of three values: ‘E’ means to check if someone is enrolled and matriculated in Banner. ‘R’ means to check if someone is registered and matriculated in Banner. Blank is valid if the user supplies the extract_sql. (This means you have already determined which users are de-matriculating and which ones are re-matriculating).
19	Advance Student Record Type	When performing a matriculation load, this is the value of the record type for a re-matriculated student, e.g., ‘ST’. It is likely this will be the same as the record_type associated with a Student load in bnr_extract_sql.
20	Advance Alumnus Record Type	When a person graduates, this is the value of their new record type, e.g., ‘AL’ for alum. It is likely this will be the same as the record type associated with a graduate load in bnr_extract_sql.
21	Process Non-local Degrees (Y/N)	Bring over non-local or degrees or not. When this is ‘Y’, non-local degrees are brought over from the sordegr table (Prior College Degree Table). ⓘ *Institution codes are not translated; it is assumed you have the same institution codes in Banner and Advance*

** If you are replacing a pre-existing Advance/Banner integration, these values may already exist and do not have to be added.*

Step 3: Populate the bnr_incremental table

As a part of your initial setup, you need to identify the end_dt_time in the bnr_incremental table. This value is the date/time the incremental_load process will use the first time it runs. When the incremental_load procedure runs, it only examines Banner changes made after this date/time value, and brings that data over to Advance. At the end of the data transfer, begin_dt_time, end_dt_time and operator_name from the run are automatically updated in the bnr_incremental table. This date should be set to the date of your first initial load.

ⓘ *This does not need to be done for subsequent incremental loads, but can be changed for any particular load, if you wish.*

Step 4: Modify the bnr_extract_sql table

The bnr_extract_sql table contains the SQL code to define which Banner entities to extract for initial loads. The system-delivered SQL extracts Student and Alumni information for a specific Banner term, as well as parent information for a group of Banner IDs of a specific Banner term. You may need to modify this SQL to define what ‘Student’ and ‘Alumni’ mean at your institution or organization. You may also define additional ‘Other’ types of entities by inserting rows in this table and writing the appropriate SQL statements. The purpose of this task is to insert rows in the Banner adv_ids_in_advance table. The entries in this table populated from the SQL in bnr_extract_sql represent the ‘universe’ of banner pidms that will be processed in that run.

Columns in the bnr_extract_sql table include:

- Batch_app_type_code
- Description
- Extract_sql
- Record_type_code
- Parent_ind
- Process_date
- Operator_code

The record_type_code is the record type of entities created for the particular batch_app_type_code. This is required for initial loads.

The parent_ind is used to indicate that the load is a parent load and that Parent/Child relationships should be created. It is important to note that Parent/Child relationships can only be created for a child that already exists in Advance, i.e., has already been created via an initial load of students or graduates. Additional information on Parent Loads can be found in the “adv_banner.initial_load” section of this document.

The process_date and operator_code are automatically updated after each initial load, matriculation load and student graduate load.

Bnr_extract_sql contains system-provided samples of SQL for use or modification to suit the needs at your institution or organization. The following Batch App Type metadata is provided:

- BS for Student
- BG for Graduate
- BP for Parent (Student)
- BQ for Parent (Graduate)
- BH for Student Graduate Load(to turn students into graduates)
- BM for Matriculation status changes
- BK for Parent incremental loads
- BL for Student and Graduate incremental loads

You should not delete or modify rows in bnr_extract_sql for a batch_app_type codes of ‘BK’ or ‘BL’, as these reference hard-coded values, and are reserved for parent incremental loads and incremental loads of student and graduate information.

Below is an example of the system-delivered extract_sql code for Student (BS):

```
INSERT INTO adv_ids_in_advance@advance.banner
(spriden_pidm)
    SELECT DISTINCT stu.sgbstdn_pidm
    FROM   sgbstdn@advance.banner stu
          ,spbpers@advance.banner pers
    WHERE  pers.spbpers_pidm = stu.sgbstdn_pidm
    AND    stu.sgbstdn_stst_code = 'AS'
    AND    stu.sgbstdn_term_code_admit = '<term_code>'
    AND    pers.spbpers_dead_ind IS NULL
    AND    pers.spbpers_dead_date IS NULL
    AND
adv_integration.f_degr_enrolled_ind@advance.banner(stu.sgbs
tdn_pidm, stu.sgbstdn_term_code_admit) = 'Y'
```

If you wish, you can generate a list of pidms to load into a separate table, then, insert the pidms from that table. For example, let's say you populate a table called mypidms (pidm number(8)). The extract for that table would be:

```
INSERT INTO adv_ids_in_advance@advance.banner
(spriden_pidm)
SELECT pidm from mypidms
```

Step 5: Set up telephone numbers

Bringing telephone data from Banner requires special logic for determining the telephone number in an address.

1. If the address has a link to a specific telephone row in Banner, the phone number in that row is put in the Advance address, regardless of whether the address has its own phone number.
2. If the address does not have a link, but the address has its own phone number, the phone number for the address is put in the Advance address.
3. Telephone rows that are not associated with an address come over as an alternate Advance address with only a phone number. Please see the next step for more information on these numbers.

Mapping additional phone numbers

The bnr_telephone_xref table in Advance is populated from stvtele in Banner as part of generate_tms_xref_data. You are responsible for maintaining this cross-reference information. If there are certain types of phone numbers, e.g., faxes, that you do not want to come over, delete the row that represents a Banner fax number. Then, map each of the Banner telephone codes you want to come over to specific Address types in Advance, for example:

stvtele_code	stvtele_desc	adv_addr_type
SE	Seasonal	S
BI	Billing	G
HQ	Corporate Headquarters	M

Do not map these telephone numbers to 'H'ome or 'B'usiness, as 'H' or 'B' telephone numbers only come over if they are linked or are in the address.

Step 6: Populate the Home/Business Address priority table

Banner allows multiple home and business addresses, so prioritization needs to be identified for the ones that will be the single Home address and single Business address in Advance. The bnr_address_priority table maps the various Home addresses in Banner and the various Business addresses in Banner using the following logic:

- If the highest priority Home address does not exist for a particular Banner ID, the system looks for the next highest priority Home-type address.
- If the highest priority Business address does not exist for a particular Banner ID, the system looks for the next highest priority Business-type address.
- The higher the priority value assigned the higher the priority of the address type.

Use thirdpty_tms_xref:

A great amount of flexibility exists regarding the bringing over of addresses and you do not have to use address prioritization. If you wish, you can translate address types using the standard thirdpty_tms_xref technique. However, in that situation you must be sure that you do not map multiple types of Banner addresses to 'H', or multiple types of Banner addresses to 'B'. Otherwise, if a given Banner ID has multiple types of 'H' addresses, one will overwrite the other. This is also true for Business addresses.

Use address prioritization/thirdpty_tms_xref:

When using prioritization, all of the lower priority 'H' and 'B' addresses should appear in the thirdpty_tms_xref with cross-references to Advance address types that are not 'H' or 'B'. The highest priority 'H/B' address does not have to be in the thirdpty_tms_xref table because it will always become an 'H' or 'B'. The lower priority 'H/B' addresses sometimes become 'H/B' (if the higher priority one does not also exist).

You are responsible for populating the bnr_address_priority table. The table below shows examples of address prioritization:

adv_addr_type	bnr_addr_type	Banner Priority
H	PR	4
H	MA	3
H	PQ	2
H	P3	1
B	W1	4
B	W2	3
B	W3	2
B	HQ	1

ⓘ ***Special Address Considerations:***

The DataLoader reviewer needs to decide what to do with an address. All 'H'ome and 'B'usiness addresses come over as 'A'dd operations, so you could end up with a duplicate address if the 'A'dd operation is performed. Addresses that are not 'H', 'B', or identified by the Advance system option 64 setting will come over as 'M'odifies if the address type and source code are the same. If it is an email address (as defined by Advance system option 64) and the source code matches Banner (Option 4), it will come over as a 'M'odify.

In DataLoader, the reviewer can select any of the existing addresses and change the operation from 'A'dd to 'M'odify, if that is appropriate. It is always a judgment call whether to modify an existing address or add it as a new one.

Step 7: Populate the bnr_parent table

The bnr_parent table in the Advance database identifies which emergency contact relationship codes are parents or which guardian codes are parents. Option 13 in the bnr_integration_options table determines whether to use emergency contacts or guardians.

The gender code in bnr_parent is used when creating the parent entity. Valid values are 'M' for Male, 'F' for Female', and 'U' for Unknown.

When the parent entity is created in thirdpty_entity the Children row will also be formulated in thirdpty_children. The relationship type is the value of Option 12 in the bnr_integration_options table.

The parent_ind in bnr_extract_sql determines whether a parent load is performed, and whether it is the system-delivered parent load or a user-defined parent load. Parent loads only add or modify entity, address, and children relationships.

Step 8: Populate the ids_base table

In order for the incremental load to work for Banner entities loaded into Advance before installing this software, and to prevent duplicate entities, the cross-references between Banner pidms and Advance id_number must exist in zz_adv_loader_entity in Advance. You can populate this directly in zz_adv_loader_entity, but it is not necessary if you already have this cross-reference defined as Advance Alternate IDs in the ids_base table. The process will look in the ids_base table and populate zz_adv_loader_entity with any missing cross-references it finds in ids_base.

Step 9: Mapping the data between Banner and Advance

Please refer to Appendix B of this document for information regarding the mapping of data between the two applications.

Initial Load

The initial_load procedure is typically performed one or more times per year on an ongoing basis. The intent of this procedure is to bring over groups of new data, e.g., a class of students that enters as freshmen or graduates as seniors. This procedure takes two parameters: batch_app_type_code, and term_code.

There are three types of initial loads:

- Student Load (Batch app type: ‘BS’)
- Graduate Load (Batch app type: ‘BG’)
- Parent Load
  - Banner Student Parent Load (Batch app type: ‘BP’)
  - Banner Graduate Load (Batch app type: ‘BQ’)

ⓘ *A Parent load must be executed AFTER the student load has been completely processed in DataLoader because the Advance id_numbers of the students must be known to the Parent Load before that batch is created.*

The following data types are brought over during an initial load for students and graduates:

- Address (thirdpty_address)
- Email (thirdpty_email)
- Awards and Honors (thirdpty_awards_and_honors)
- Degrees (thirdpty_degrees)
- Entity (thirdpty_entity)
- Entity Birth and Death (thirdpty_entity_birth_death)
- IDS: Alternate ID such as Banner Student ID and SSN (thirdpty_ids)
- Student activities (thirdpty_student_act)
- Student Sports (thirdpty_sports)

The following data types are brought over during an initial load for parents:

- Address (thirdpty_address)
- Entity (thirdpty_entity)
- Children (thirdpty_children)

🛠 *Conditional logic is based on Batch App Type. During the processing of data, the adv_banner package will check to see if the particular type of data is represented in ldr_batch_tables for the specific batch_app_type_code selected. If the thirdpty_ table associated with a specific type of data is not in the list, that type of data will not be processed. This gives you a way to easily eliminate certain types of data from being processed. The ldr_batch_tables refers to the DataLoader Batch Type metadata maintained in the Advance Configuration Utility. DataLoader Batch Type data are system-delivered for each of the rows in bnr_extract_sql.*

Initial loads can occur for Students, Graduates, and Parents using system-delivered information (after all necessary configuration has occurred). The following examples are for the purpose of demonstration and illustrate the steps you would use when performing different loads. Specific settings will differ, depending on the configuration at your institution or organization.

Example Steps for a Student Load

A. To load incoming students in the term 200810, perform a student load by running the initial load procedure using the following syntax, using the batch_app_type code and term code as parameters:

```
exec adv_banner.initial_load (‘BS’, ‘200810’);
```

B. Log on to Advance Web and use Advance Web DataLoader to create a batch.

C. Create a batch using the Batch Type of ‘BS’.

D. Use DataLoader to validate the data and post the data to Advance.

Example Steps for a Graduate Load

A. To load graduating students in the term 200810, perform a graduate load by running the initial load procedure using the following syntax, using the batch_app_type code and term code as parameters:

```
exec adv_banner.initial_load (‘BG’, ‘200810’);
```

B. Log on to Advance Web and use Advance Web DataLoader to create a batch.

C. Create a batch using the Batch Type of ‘BG’.

D. Use DataLoader to validate the data and post the data to Advance.

You have the option of bringing over Emergency Contact information or Guardian information as parent entities with a parent/child relationship to a student/grad. Both types of information have a relationship code that can be used to indicate a parent relationship. This will result in the parents being created as new entities with children records.

Example Steps for a Parent Load

A. To load parents of the students loaded in the first example, perform a parent load by running the initial load procedure using the following syntax, using the batch_app_type code and term code as parameters:

```
exec adv_banner.initial_load (‘BP’, ‘200810’);
```

B. Log on to Advance Web and use Advance Web DataLoader to create a batch.

C. Create a batch using the Batch Type of ‘BP’.

D. Use Advance Web DataLoader to validate the data and post the data to Advance.

Incremental Load

The incremental load includes all changes for parents, students and graduates. The batch app type code to run the incremental load is ‘BL’, which is the only parameter passed.

ⓘ *As a part of your initial setup, you identified the end_dt_time in the bnr_incremental table. Be sure that this it set correctly before running the incremental load. This date should be set to the date of your first initial load.*

The following data types are brought over during an incremental load:

- Address (thirdpty_address)
- Awards and Honors (thirdpty_awards_and_honors)
- Children (thirdpty_children)
- Degrees (thirdpty_degrees)
- Entity (thirdpty_entity)
- Entity Birth and Death (thirdpty_entity_birth_death)
- IDS: Alternate ID such as Banner Student ID and SSN (thirdpty_ids)
- Student activities (thirdpty_student_act)
- Student Sports (thirdpty_sports)

Example Steps for Incremental Loads

A. Perform an incremental load by running the incremental load procedure using the following syntax, using the batch_app_type code as a parameter.

A. exec adv_banner.incremental_load (‘BL’);

B. Log on to Advance Web and use Advance Web DataLoader to create a batch.

C. Create a batch using the Batch App Type of ‘BL’ to see student and graduate changes or ‘BK’ to see parent changes.

D. Use Advance Web DataLoader to validate the data and post the data to Advance.

Student-to-Graduate Load

If students are already loaded and they are graduating, you will need to run student_graduate_load. This procedure takes two parameters, batch app type code and graduation date (grad_date). A ‘BH’ batch_app_type_code row in bnr_extract_sql is system-delivered for this purpose. You can modify the SQL in that row or use your own batch_app_type_code, if you wish. The grad_date parameter is a token value that will replace the token ‘grad_date’ in the extract SQL, if it exists.

The third_pty_record_type table is populated during this load

The net result of performing this load is to generate record type changes for graduating students in the following situations:

A. The student has a record type equal to the value in Option 19 (e.g., Advance Student Record Type = ‘ST’), but does not have a record type equal to the value in Option 20 (e.g., Advance Alumnus Record Type = ‘AL’)

B. The student has a record type equal to the value in Option 20 (e.g., Advance Alumnus Record Type = ‘AL’), but does not have a record type equal to the value in Option 19 (e.g., Advance Student Record Type = ‘ST’).

C. The student has records types equal to Option 19 (‘ST’) and Option 20 (‘AL’).

In situation ‘A’, the existing record type, e.g., ‘ST’ will be modified to the value in Option 20 (‘AL’), thereby preserving the other data in the record.

In situation ‘B’, a Modify operation will be created in DataLoader for the Option 20 (‘AL’) record type. All existing data in the record type will be preserved. The DataLoader reviewer can discard the row or post it for audit trail purposes.

In situation ‘C’, a Modify operation will be created in DataLoader for the Option 20 (’AL’) record type and a Delete operation will be created in DataLoader for the Option 19 (‘ST’) record type.

Matriculation Load

When a student’s matriculation status has changed in Banner it needs to be reflected in Advance. The following matriculation status changes are possible:

- Student de-matriculation (student is no longer matriculated)
- Student re-matriculation (a former student becomes matriculated again)

For example, you might use the following settings to run your matriculation load for the term ‘200810’:

1. Set Banner option 5 to the record type value for a de-matriculated student, e.g., ‘AL’.
2. Set Banner option 6 to the record status value for a de-matriculated student, e.g., ‘I’.
3. Set Banner option 7 to the solicit control code for a de- matriculated student e.g., ‘D’.
4. Set Banner option 19 to the Advance student record type value, e.g., ‘ST’.
5. Run the matriculation load procedure to bring entities with matriculation status changes for a given term:
6. exec adv_banner.matriculation_load ('BM', ‘200810’);
7. Create a batch using the batch type of ‘BM’ in Advance Web Dataloader
8. Review the batch and post the data to Advance

The following changes occur for a de-matriculated student in the specified Banner term:

- The record type in Advance is set to ‘AL’ (Option 5).
- The record status is changed to ‘I’ (Option 6).
- The solicit control code is set to ‘D’ (Option 7).

The following changes occur for a re-matriculated student in the specified Banner term

- The record type in Advance is set to ‘ST’ (Value in Option 19 to represent student).
- The record status is changed to ‘A’ (Value in Option 16 to represent student status when a student gets created in Advance).
- The solicit control code is set to a value which is the default value for the solicit control code for a student entity when the student is created in Advance (which is usually blank).

Three tables are populated during the matriculation load:

- thirdpty_entity
- thirdpty_mail_control
- thirdpty_record_type

The logic for determining whether someone is matriculated depends on the term code. To provide as much flexibility as possible for matriculation logic, a procedure called matriculation_load is used. This procedure takes two parameters, batch_app_type_code and term_code. The batch_app_type_code will point to one of the rows in bnr_extract_sql. A system-delivered ‘BM’ row can be used for this purpose or, just like initial loads, you can create your own row. The extract_sql value is blank for the ‘BM’ row.

If you leave the extract_sql value blank, the first thing the matriculation_load procedure will do is populate adv_ids_in_advance with all pidms that it considers to be current students or de-matriculated students. To be considered a current student, it will check record type if Option 5 has a value and/or record status if Option 6 has a value.

- If Option 5 has a value, a pidm is considered to be a current student in Advance, if the record type is the same as Option 19 and Option 6 is blank.
- If Option 6 has a value, a pidm is considered to be a current student in Advance, if the record status is the same as the one in Option 16 and Option 5 is blank
- If both Option 5 and Option 6 have values, a pidm is considered to be a current student in Advance, if the record type is the same as Option 19 and the record status is the same as the one in Option 16.

ⓘ *When a matriculation_load is run, either Option 5 and/or Option 6 must have a value.*

The value of Option 18 establishes how it is determined whether a person is currently matriculated in Banner.

- If you enter a value for Option 18 (‘E’nrolled or ‘R’egistered) and the extract_sql for the ‘BM’ row is not populated, the system-delivered logic determines which pidms will be brought over.
- If you enter a value for Option 18 (‘E’nrolled or ‘R’egistered) when you populate the extract_sql for the ‘BM’ row, you are responsible for supplying the bnr_matric_ind values in adv_ids_in_advance for each pidm.
- If you leave Option 18 blank, you are responsible for supplying the data in adv_ids_in_advance to indicate that a person has de-matriculated or re-matriculated. When you supply this data it will significantly speed up the matriculation logic procedure. For example:
    - Set the adv_matric_ind to ‘Y’ and the bnr_matric_ind to ‘N’ (for de-matriculation)
    - Set the adv_matric_ind to ‘N’ and the bnr_matric_ind to ‘Y’ (for re-matriculation).

ⓘ *If you populate the extract_sql for the ‘BM’ row, or whatever row you are using, it will override the logic for populating adv_ids_in_advance.*

ⓘ *When populating the extract_sql for the ‘BM’ row, you can still use Option 18 to determine the bnr_matric_ind values.*

Special cases

Identifying recently deceased entities

When bringing over new Banner IDs during an initial load, only people that are not deceased are extracted. For incremental loads, people recently deceased in Banner will be updated in Advance. If an individual is deceased in Banner since the last incremental update, a Modify row is created in the thirdpty_entity_birth_death table. The death_obit_code is set to the value in Option 11 in the bnr_integration_options table. When Advance system option 187 (‘Perform Automated Entity Deceasing Logic’) is ‘Y’, automated entity deceasing occurs when the Bio Detail (Birth/Death) date is posted in DataLoader.

Identifying Logically Unique Data

The following types of data will only generate Add operations, i.e., no Modify operations, because all of the relevant data in those records help to uniquely identify the row:

- Awards and Honors
- Student Activities

Email Logic

Option 15 is used to determine the type of email address to create in Advance. If multiple email addresses exist in Banner of this type, only the preferred is brought over. If this is blank, no email will be brought over. When checking Advance to see if an existing email in Advance should be updated, addr_type_code and data_source_code are compared. If the addr_type_code is the same as this value and the data_source_code is the same as option 4, the operation code will be set to ‘M’odify.

Social Security Number logic

Option 8 determines whether or not Social Security Numbers are brought over to Advance as Alternate IDs. This option should contain the value in tms_ids_type that represents Social Security Number in Advance. If the option is populated, the SSN will be brought over if it is recorded in Banner.

Local vs. Non-Local Degrees

Option 21 establishes whether to bring over non-local or degrees or not. When it is ‘Y’, non-local degrees are brought over from the sordegr table (Prior College Degree Table).

ⓘ Institution codes are not translated; it is assumed you have the same institution codes in Banner and Advance.

5 IFAS

Single Sign-on

IFAS (Integrated Financial and Administrative Solution) is a SunGard Public Sector Web-based ERP product designed to help improve efficiencies in your organization. Functionality is provided for configuring Advance Web so users with valid accounts on both Advance Web and IFAS may navigate to IFAS, after logging on to Advance Web, without an additional logon. To accomplish this single sign-on, when the navigation from Advance Web to IFAS is attempted via a method of your choosing, e.g., a hyperlink, Advance Web checks the HTTP Server Variable for the authenticated user's username stored during initial logon. If the username is valid for logon to IFAS, i.e., matches exactly, including proper case, the user is granted access.

To set up this functionality, the web.config file on the web server must be modified as follows:

1. Open the web.config file and search for the section which contains `<!--WebSec Authentication Model-->`. Beneath this line, you will find `<add key="SecurityProviderType" value="advWebSec" />`
2. Change the value `"advWebSec"` in this line to the security provider type `"advWebServerPreauthenticated"`.

   ⓘ *The IIS virtual directory for Advance Web must have Anonymous access turned off in the Directory Security settings for the correct user values to be sent in the HTTP header.*
3. After changing the value, add another key line beneath the line you changed for `"SecurityProviderArgs"`, which will identify how you wish to set the username. The SecurityProviderArgs key supports a character on which the username can be split. The parts of the SecurityProviderArgs value are separated by a semicolon (;):
   - The first part of the value is the HTTP server variable to retrieve, e.g. AUTH_USER.
   - The second part of the value is the character used to split the username.
   - Optionally, a third part of the value is a zero-based index to pull out of the split value. If a value is not specified, this defaults to 0 (the value before the first occurrence of the split character).

     For example:

     `AUTH_USER;@` Accepts an email address and uses the part before the @ sign as the username

     `AUTH_USER;\;1` Accepts a Windows username of the form Domain\User and uses the User part.
4. Save your changes.

IFAS Chart of Accounts Synchronization

Functionality is provided in Advance Web for synchronizing the Chart of Accounts (the system of record for an identified subset of fields in Advance Web) between Advance Web and IFAS. When additions or changes are made, information will be sent from IFAS to Advance Web to provide the updated data and Advance Web performs the appropriate updates to the corresponding tables.

This functionality is flexible, so the policies and procedures at your institution or organization will determine the appropriate steps to be followed when an allocation record is updated. This includes determining the impact on any gift batches that are processed as well as those currently open and in process. This may include running the Mass Gift Mod Utility to update historical transaction data. You should also define similar policies and procedures to ensure appropriate handling occurs when an Allocation record is created or updated via this IFAS Chart of Accounts interface.

Appropriate personnel at your institution or organization are responsible for using the tools in Advance Web and IFAS to perform the necessary steps to configure the interface's functionality. No additional licensing is required to use this functionality in Advance Web.

System Options

As a part of this functionality, the following Advance system options must be modified:

System Option 238

User Group value for IFAS-initiated Allocation table updates

- ♦ blank The setting of this system option allows your institution or organization to choose which User Group value should be in the User Group field in the Allocation table for updates initiated by the IFAS Chart of Accounts Interface. Drop-down list of values validated against the zz_user_group database table.

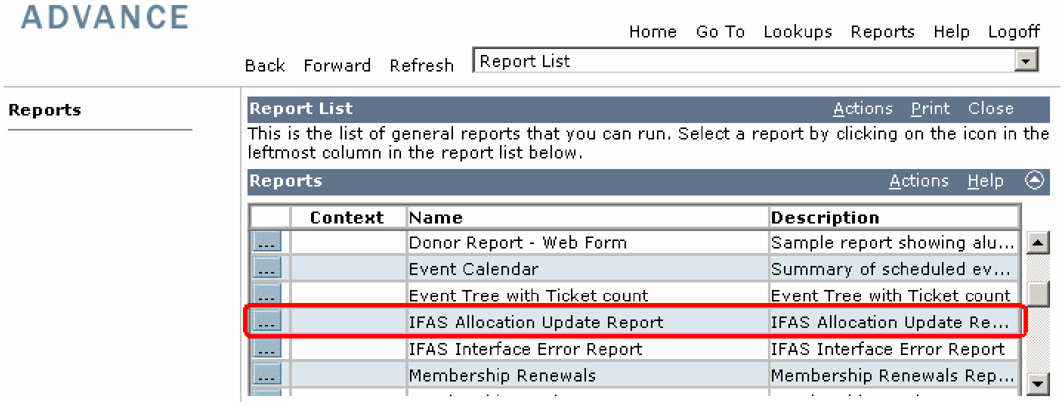
ⓘ *If your institution or organization does not set this value, an error condition results when an Add or Modify operation is attempted. Advance Web returns an error message to IFAS to indicate this field is required and is blank.*

System Option 313

IFAS: Validate IFAS when closing gift batch

- ♦ N When turned on, closing a gift batch in the Gift Batch List automatically validates the data on IFAS. The batch cannot be closed if there are errors.

A report, ‘IFAS Allocation Update Report’ is available from the Reports menu in the Main Menu.



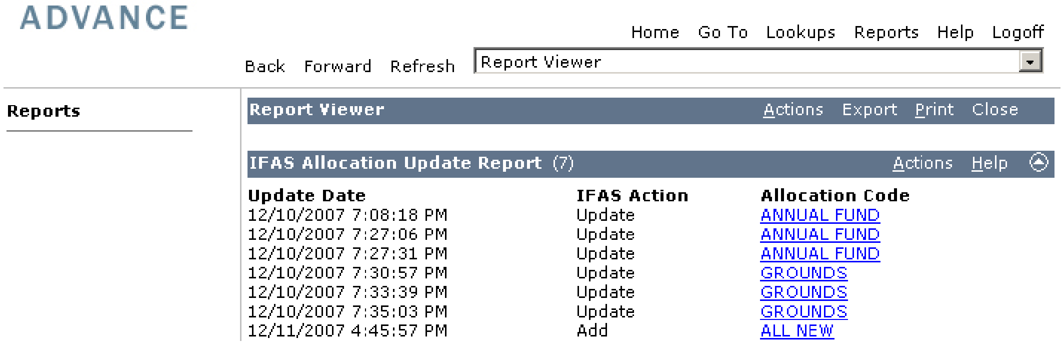
ADVANCE
Home Go To Lookups Reports Help Logoff
Back Forward Refresh
Report List
Reports
Report List Actions Print Close
Reports Actions Help

This is the list of general reports that you can run. Select a report by clicking on the icon in the leftmost column in the report list below.

	Context	Name	Description
...		Donor Report - Web Form	Sample report showing alu...
...		Event Calendar	Summary of scheduled ev...
...		Event Tree with Ticket count	Event Tree with Ticket count
...		IFAS Allocation Update Report	IFAS Allocation Update Re...
...		IFAS Interface Error Report	IFAS Interface Error Report
...		Membership Renewals	Membership Renewals Rep...

Figure 39: IFAS Allocation Update Report

This report provides information on the IFAS Action and Allocation Code for the date range selected.



ADVANCE
Home Go To Lookups Reports Help Logoff
Back Forward Refresh
Report Viewer
Reports
Report Viewer Actions Export Print Close
IFAS Allocation Update Report (7) Actions Help

Update Date	IFAS Action	Allocation Code
12/10/2007 7:08:18 PM	Update	ANNUAL FUND
12/10/2007 7:27:06 PM	Update	ANNUAL FUND
12/10/2007 7:27:31 PM	Update	ANNUAL FUND
12/10/2007 7:30:57 PM	Update	GROUNDS
12/10/2007 7:33:39 PM	Update	GROUNDS
12/10/2007 7:35:03 PM	Update	GROUNDS
12/11/2007 4:45:57 PM	Add	ALL NEW

Figure 40: Report Viewer

How this functionality works:

The steps below provide an overview of how this functionality works:

1. **IFAS Trigger -** An Add, Modify or Delete transaction made in IFAS to a Chart of Accounts record will automatically trigger a call to the Advance Web Chart of Accounts web service and pass the updated dataset to Advance Web. This data is then passed to the Advance Web stored procedure/web service for processing Allocations.

   ⓘ *This is only done for accounts in IFAS that are configured to require an update to Advance Web. It is possible there may be accounts in IFAS that do not correlate to any allocations in Advance Web.*

2. **IFAS Web Service Call -** The stored procedure/web service calls the stored procedure used by the Allocation Detail form to handle Adds and Updates.

3. **Allocation Update Process -** When the Allocation Update process is run for each record, the resulting action is one of the following:

   Successful Update

   - The record is updated in the Allocation table.
   - An Audit Record is created to indicate this update was performed successfully.
   - A reply indicating success is passed back to IFAS.

Unsuccessful Update

- The record is not updated in the Allocation table.
- An Audit Record is created to indicate the update was not successful and the reason why the update was not successful.
- An error message is returned to IFAS indicating the error(s) occurring for that record.

4. **Action Indicator -** The IFAS gift_financial_interface table contains an Action field to indicate the IFAS action that now requires a corresponding update to Advance Web. The valid options are Add (A), Modify (M) or Delete (D).
5. **Special Handling: Delete -** If IFAS sends a Chart of Account update record with Delete as the action, Advance Web performs an Update resulting in the modification of the allocation status field to Inactive. No other data elements are updated based on information sent by IFAS for a Delete. The normal Advance field updating occurs, i.e., date added, date modified, user group.
6. **Fields IFAS does not populate -** The following fields are not populated by IFAS in the interface file. Any values provided in these fields are ignored, as the logic to populate these fields is performed by Advance Web the same as an Allocation record add or update:
    - date_added
    - date_modified
    - user_group
7. **Layout of Errors Reported back to IFAS -** The result set/reply returned by the web service indicates whether the update was successful. If there are errors, the errors are included in this reply.
8. **Layout of Allocation Audit Log** (allocation_load_audit_ifas) – Provides a complete history of all allocations created/updated in the Allocation table as well as an audit trail of those rejected transactions.

    The following are stored in the Allocation Audit Log:

    - All fields in the initial transaction sent by IFAS including the Action.
    - Date Attempted – load_attempted_date - contains the date the update was attempted in the Allocation table.
    - Status – load_status - F = Fail, P = Pass
    - Error Messages – load_error_msg – this CLOB contains the text associated with the one or many error messages that occurred during the update attempt.

IFAS Gift Transactions

An interface between Advance and IFAS facilitates sending financial information from the Advance Gift and Pledge tables to IFAS to create an IFAS Batch, which can be reviewed and posted to the IFAS Financial System. This interface:

- Allows your institution or organization to identify the transactions to be included in the IFAS interface file.
- Provides your institution or organization with a method of identifying which Advance data elements populate the various data elements in the IFAS interface file.
- Automates the transfer of pertinent gift and pledge data from Advance to IFAS.
- Provides an audit mechanism to track the success or failure of the transfer of transactions from Advance to IFAS.
- Provides reporting that can be used for reconciliation purposes.
- Provides a method for clients to identify, correct, and resubmit any transactions that fail to be transferred to IFAS.

Overview of Process

Once you perform all necessary system configurations (described in the following sections), the general process for moving transactions from Advance to IFAS is as follows:

1. Rules are established via the Advance Configuration Utility or using custom SQL.
2. Transactions are entered into Advance batches and processed by the Advance gift processing procedure as usual.
3. A system process is scheduled to follow nightly gift processing. This scheduling needs to be set up by your institution. This process reads the newly posted Advance transaction data and apply your rules for selecting, mapping, and writing the appropriate Advance data to the new gift_financial_interface table, and will call a web service to pass the data to IFAS. Following the web service call, the transactions will be written to the Advance Web audit table, and each transaction passed through the web service will be marked with a “GL Interface Status” flag to indicate whether or not it was successfully fed to IFAS.
4. If any errors occur in passing transactions from the gift_financial_interface table to IFAS, the failed transactions are written to the error log, along with the reason for failure. If a single transaction in a given Advance batch fails to pass to IFAS, the entire Advance batch fails. All transactions in that Advance batch are written to the error log, with specific error messages for the failed transactions, and general “batch failed” error messages for transactions not passed due solely to failure of other transactions in the batch. You should review the error log and address any errors. Any Advance batches having an error will fail and not pass to IFAS.
5. The “GL Interface Status” flag in the Advance batch control table indicates whether or not the transactions in that batch were successfully passed to IFAS. A failed batch continues to be selected each time the Advance Gift Interface File creation process is run, until its errors are corrected, it processes successfully and is accepted by the IFAS web service.
6. Once transactions successfully pass from Advance to IFAS, the IFAS Batch Proof and Distribution processes are run to finalize the posting to IFAS and report is generated. Manual intervention and communication between the IFAS and Advance users responsible at your institution or organization is required if any errors occur at this point.

Mapping the data

A gift interface table, gift_financial_interface, stores the transactions to be passed through the Web Service to IFAS. This table contains the specific columns to be used by IFAS to create batches. Once the IFAS batches are created, they are run through the standard IFAS batch proof and distribution processes for posting into IFAS.

All gift_financial_interface columns shown in the table below are available for mapping via the Advance Configuration Utility (please see the instructions, starting on page 88), with the exception of record_number and rule_id, which are system-generated.

A 'Y' in the 'Mapping Required' column indicates required data and the column must be mapped. Mapping is optional for all other columns.

† These columns are system-generated

** These columns are Advance-specific columns for auditing and other purposes.*

***'je' (Journal Entries)prefixed columns are GIK (Gifts-In-Kind)*

ⓘ *For more information on the structure of this table, please refer to the "Database Changes" section at the end of this document.*

Column Name	Mapping Required	Comments
record_number†		Sequential record number
rule_id†		Rule # that caused the transaction to be selected
batch_number*	**Y**	Advance Batch Number
gl_transaction_type*	**Y**	'G'ift, 'P'ledge, or 'M'atching Gift, or other default or mapped one-character value, other than a space
transaction_number*	**Y**	Advance Receipt or Pledge Number
xsequence*	**Y**	Sequence number
processed_orig_occurrence*	**Y**	Occurrence number for the amount of times the gift has been modified
interface_system*	**Y**	Code representing Interface System (I)
run_date_time*	**Y**	Date process is run to create table (SYSDATE)
arbd_gl_gr	**Y**	General Ledger code
arbd_acct_id	**Y for AR**	Customer Account ID, required if arbd_tr_type = AR
arbd_at_term	**Y**	Term Code
arbd_gl_key	**Y**	GLKEY: Up to 10 characters which defined the major account or project number
arbd_gl_obj	**Y**	GLOBJ: Up to eight characters which identifies the minor account or object
arbd_jl_gr		Job Ledger Code
arbd_jl_key		Job Ledger Project Code
arbd_jl_obj		Job Ledger Object Code
arbd_div	**Y**	Division Code separates transactions into logical groups
arbd_curr_cd		Currency Code
arbd_msur		Units of Measure
arbd_qty	**Y**	Quantity

Column Name	Mapping Required	Comments
arbd_prin_amt	**Y**	Principal Amount
arbd_int_amt		Interest Amount; will default to ‘0’
arbd_disc_amt		Discount Amount; will default to ‘0’
arbd_ref	**Y for AR**	Source Document Number (AR Invoice number), required if arbd_tr_type = AR
arbd_ref2	**Y for CR**	Cash Receipt Number, required if arbd_tr_type = CR
arbd_wo		Work Order
arbd_desc		Up to 30 characters of description.
arbd_fin_cd		Finance Code. Identifies the Type of Charge (AID, FEE, TUIT)
arbd_misc		Miscellaneous Code
arbd_batch_id	**Y**	Unique Identifier used to group transactions - you should enter a placeholder <BATCH_NUMBER> in the default column on the mapping tab. This value cannot be numeric, so is prefaced with an alpha character, ‘A’
arbd_tr_type	**Y**	Transaction Type 'AR' - Charge, ‘CR’ - Payment
arbd_pay_type		Payment Type. User Defined, normally ‘CC’ Credit Card, ‘CK’ Check, & ‘CA’ Cash
arbd_pay_ref		Payment Reference. Up to 16 characters which uniquely identifies the type of payment, i.e., check or credit card number
arbd_bank_id		Bank ID. Two character Code which identifies which bank will receive this payment as a deposit. Validated from the CKID Common Codes only if ‘Y’ is set on GEN-MST R screen
arbd_bank_slip		Bank Deposit Slip Number
arbd_bank_dt		Date of Bank Deposit CCYYMMDD; will default to SYSDATE
arbd_cc_exp		Credit Card Expiration Date
arbd_cc_auth		Credit Card Authorization
arbd_po_ref		Purchase Order Reference
arbd_tax_amt		Tax Amount; will default to ‘0’
arbd_tax_amt2		Tax Amount 2; will default to ‘0’
arbd_tax_cd		Tax Code
arbd_tax_cd2		Tax Code 2
arbd_duty_amt		Duty Amount; will default to ‘0’
arbd_duty_cd		Duty Code
arbd_disc_terms		Discount Terms; will default to ‘0’
arbd_rel_one		Relate Code 1. This ‘relate to’ code indicates if this transaction is related to another subsystem or process, e.g., ‘AP’ or ‘1099’
arbd_rel_two		Relate Code 2. This ‘relate to’ code indicates if this transaction is related to another subsystem or process, e.g., ‘AP’ or ‘1099’

Column Name	Mapping Required	Comments
arbd_charge_amt		Charge Amount; will default to ‘0’
arbd_charge_cd		Charge Code
arbd_addr_cd		Address Code. If the transaction referenced should be sent to a specific address during the billing or invoice processing
arbd_ref_dt	Y	Transaction Reference Date
arbd_acct_name		Name
arbd_pedb_code		Person Entity Data Base Code
arbd_trns_format	Y	GL posting description. Almost always ‘NB’
arbd_hit_ar	Y	Hit AR Flag. Gifts that are pledge payments to ‘trackable’ (recordable) pledges
arbd_ref_amt		Reference Amount; will default to ‘0’
je_batch_id**	Y	Set ID: Multiple records are grouped together in what is called a Set or Batch.
je_ref_no**	Y	JE ID: This is the Journal Entry reference number
je_description**	Y	Description: The transaction description. This should be no larger than 30 characters
je_trans_ref_date**	Y	Date: The transaction reference date
je_ref_date2**		Secondary Date:
je_gl_gr**	Y	GL Ledger: The General Ledger code for this entry
je_gl_key**	Y	GL Organization Key: The General Ledger Organization Key for this entry
je_gl_obj**	Y	GL Object Code: The General Ledger Object Code for this entry
je_jl_gr**		JL Ledger: The Job Ledger code for this entry. If left blank, the user’s default code will be used
je_jl_key**		JL Organization Key: The Job Ledger Organization Key for this entry
je_jl_obj**		JL Object Code: The Job Ledger Object Code for this entry
je_wo**		Work Order: If applicable, a Work Order number can be passed into the Journal Entry transaction
je_dr**	Y	Debit: A positive Debit or Credit dollar amount should be provided, never both
je_cr**	Y	Credit: A positive Debit or Credit dollar amount should be provided, never both
je_units**		Units: A number of Units may optionally be provided along with or without a Debit/Credit amount
je_offset_intra**		Intrafund Offset: Set to ‘Y’ to apply intrafund offsets automatically when distributing this transaction
je_offset_inter**		Interfund Offset: Set to ‘Y’ to apply interfund offsets automatically when distributing this transaction
je_ref_no2**		Secondary Reference: A reference number related to the origin or subsystem of the source transaction
je_peid**		PE ID: A Person/Entity ID relating to this entry.

Column Name	Mapping Required	Comments
je_contract_no**		Contract No: A Contract number that relates to this entry
je_prep_id**		Prepared By: The User ID of the person who prepared this entry
je_misc**		Misc: A user-defined code that identifies the transaction for reporting purposes
je_hit_en**	**Y**	Hit Other: This field identifies the subsystem to which this transaction should be entered and this field is required: Value Purpose N General Ledger Only (Default) Y Encumbrances and General Ledger E Encumbrances Only B Budgets Only K Bank Reconciliation and General Ledger L Bank Reconciliation Only O Accounts Payable Open-Hold Only I Accounts Payable Immediate Pay Only
je_en_type**		Other Type: If the 'Subsystem Flag' is not 'N', then a transaction type must be entered in this field. The following transaction types are allowed by subsystem: Accounts Payable or Encumbrances: EN (Encumbrance), DE (Disencumbered), PP (Partial Payment), or FP (Full Payment) Bank Reconciliation: AJ (Adjustment), BF (Fee), BI (Interest), CK (Check), DP (Deposit), EF (Electronic Funds Transfer), or RV (Reversal). Budgets: A (Budget Adjustment) or N (New Budget)
je_ck_id**		Check ID: The Check Stock ID or Bank ID used with this entry. This code must be defined in Nucleus Common Codes file under the code category CKID. This field is required if 'HitEn' is 'K' or 'L'
je_ck_no**		Check Number: A Check Number that relates to this entry. This field is required if 'HitEn' is 'K' or 'L'
je_budget_over**		Budget Override: If users creating batches have the required Budget Override security defined in Nucleus Database Security, a 'Y' can be placed in this field to override budget blocks at data entry time allowing exit from and thus creation of an over-budget JE batch. • NOTE: This flag only allows creation of the batch. Correction of the over-budget condition is still required before the batch will post
je_subsys**	**Y**	Sub System: The IFAS sub system to which the entry will go into and this field is Required

Column Name	Mapping Required	Comments
je_views**		Views: Four check boxes to select the accounting views to which these transactions will relate: • Accrual • Cash • Modified Accrual • User Defined Click on the box to check the views you would like
je_post_state**	**Y**	Status: This field indicating whether or not the journal entry set has been distributed and this field is Required • DS The entry set has been distributed. • BE The entry set has not been distributed.
je_create_date**		Create Date: showing the date of creation. This is useful in determining current from non-current sets
je_create_user**		This field will display the name of the person creating the set file. The name used to log in will be the default name
je_text**		Test: Any notes referring to the transaction
misc1*		Miscellaneous data to be mapped, if you wish; not used by IFAS
misc2*		Miscellaneous data to be mapped, if you wish; not used by IFAS
misc3*		Miscellaneous data to be mapped, if you wish; not used by IFAS
misc4*		Miscellaneous data to be mapped, if you wish; not used by IFAS
date_added*	**Y**	SYSDATE
date_modified*	**Y**	SYSDATE
operator_name*	**Y**	Valid Advance Operator Name
user_group*	**Y**	Valid Advance User Group

As a part of this functionality, two TMS views, tms_alloc_fee_code and tms_credit_card, are available.

Two columns to note in the tms_alloc_fee_code view are:

fee_type This indicates whether the fee to be charged is a percentage of the transaction or a flat amount. Valid values are either a percent symbol (%) or a dollar sign ($).

fee_value This indicates the value to be associated with the fee_type. A fee_type of $ with a fee_value of 100.00 indicates there is a $100.00 fee associated with this code. A fee_type of % with a fee_value of 5.25 indicates there is a 5.25% fee associated with this code.

Examples of values for the columns in this table are:

Code	Description	Fee Type	Fee Value
2	2 Percent	%	2.00
3	3 Percent	%	3.00
5	5 Percent	%	5.00
C	$100	$	100.00

Two columns to note in the tms_credit_card view are:

fee_type This indicates whether the fee to be charged is a percentage of the transaction or a flat amount. Valid values are either a percent symbol (%) or a dollar sign ($).

fee_value This indicates the value to be associated with the fee_type. A fee_type of $ with a fee_value of 100.00 indicates there is a $100.00 fee associated with this code. A fee_type of % with a fee_value of 5.25 indicates there is a 5.25% fee associated with this code.

Examples of values for the columns in this table are:

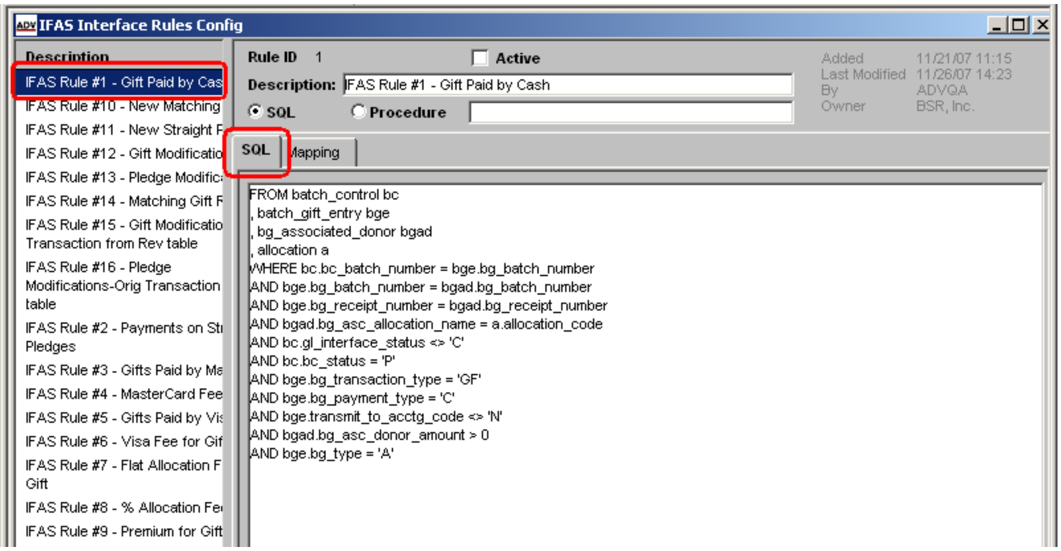
Code	Description	Fee Type	Fee Value
V	Visa	%	5.00
M	MasterCard	$	0.50
D	Discover	$	0.25
A	American Express	%	3.00

System Option 146, “Use TMS view for Allocation Fee instead of a check box and make Allocation Fee Minimum and Allocation Fee Alternate visible,” must be set to ‘Y’ for fee handling to function as described above.

IFAS Interface Rules Configuration

The Advance Configuration Utility is used to establish rules for the selection and mapping of Advance gift and pledge data to IFAS. To access the IFAS Interface Rules Configuration window and establish rules:

1. Log on to the Advance Configuration Utility and select Tools → Configuration → IFAS Interface Rules from the menu.



Description
- IFAS Rule #1 - Gift Paid by Cas
- IFAS Rule #10 - New Matching
- IFAS Rule #11 - New Straight F
- IFAS Rule #12 - Gift Modificatio
- IFAS Rule #13 - Pledge Modifica
- IFAS Rule #14 - Matching Gift F
- IFAS Rule #15 - Gift Modificatio Transaction from Rev table
- IFAS Rule #16 - Pledge Modifications-Orig Transaction table
- IFAS Rule #2 - Payments on Str Pledges
- IFAS Rule #3 - Gifts Paid by Ma
- IFAS Rule #4 - MasterCard Fee
- IFAS Rule #5 - Gifts Paid by Vis
- IFAS Rule #6 - Visa Fee for Gif
- IFAS Rule #7 - Flat Allocation F Gift
- IFAS Rule #8 - % Allocation Fee
- IFAS Rule #9 - Premium for Gift

Rule ID 1 □ Active

Description: IFAS Rule #1 - Gift Paid by Cash

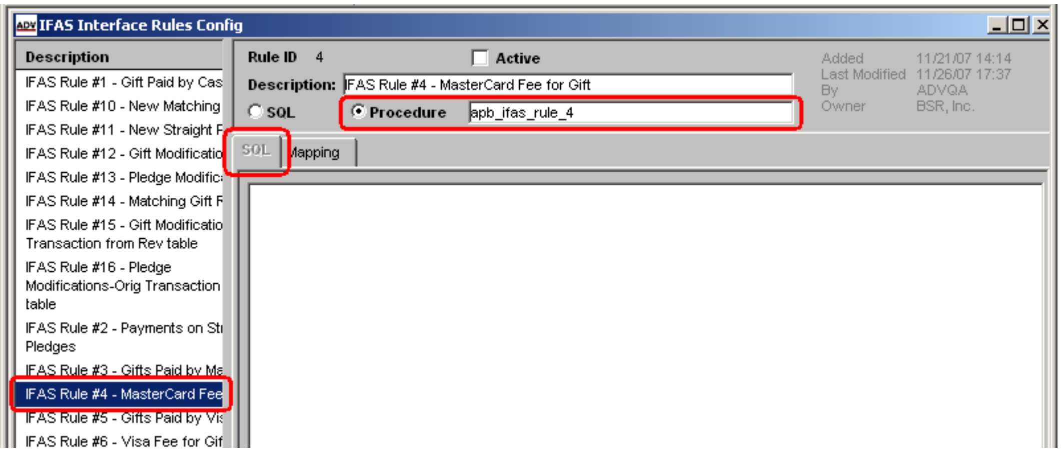
● SQL ○ Procedure

Added	11/21/07 11:15
Last Modified	11/26/07 14:23
By	ADVQA
Owner	BSR, Inc.

```
FROM batch_control bc
, batch_gift_entry bge
, bg_associated_donor bgad
, allocation a
WHERE bc.bc_batch_number = bge.bg_batch_number
AND bge.bg_batch_number = bgad.bg_batch_number
AND bge.bg_receipt_number = bgad.bg_receipt_number
AND bgad.bg_asc_allocation_name = a.allocation_code
AND bc.gl_interface_status <> 'C'
AND bc.bc_status = 'P'
AND bge.bg_transaction_type = 'GF'
AND bge.bg_payment_type = 'C'
AND bge.transmit_to_acctg_code <> 'N'
AND bgad.bg_asc_donor_amount > 0
AND bge.bg_type = 'A'
```

2. Sample rules are system-delivered as inactive and their descriptions display to the left of the window. Click on a rule to see its details. When the SQL tab is active, and the SQL radio button is selected, the code for the rule displays in the lower portion of the window (shown above).

   When the rule you select is a procedure, the SQL tab is inactive and the name of the procedure displays in the field to the right of the button label (as shown below).



Description
- IFAS Rule #1 - Gift Paid by Cas
- IFAS Rule #10 - New Matching
- IFAS Rule #11 - New Straight F
- IFAS Rule #12 - Gift Modificatio
- IFAS Rule #13 - Pledge Modifica
- IFAS Rule #14 - Matching Gift F
- IFAS Rule #15 - Gift Modificatio Transaction from Rev table
- IFAS Rule #16 - Pledge Modifications-Orig Transaction table
- IFAS Rule #2 - Payments on Str Pledges
- IFAS Rule #3 - Gifts Paid by Ma
- IFAS Rule #4 - MasterCard Fee
- IFAS Rule #5 - Gifts Paid by Vis
- IFAS Rule #6 - Visa Fee for Gif

Rule ID 4 □ Active

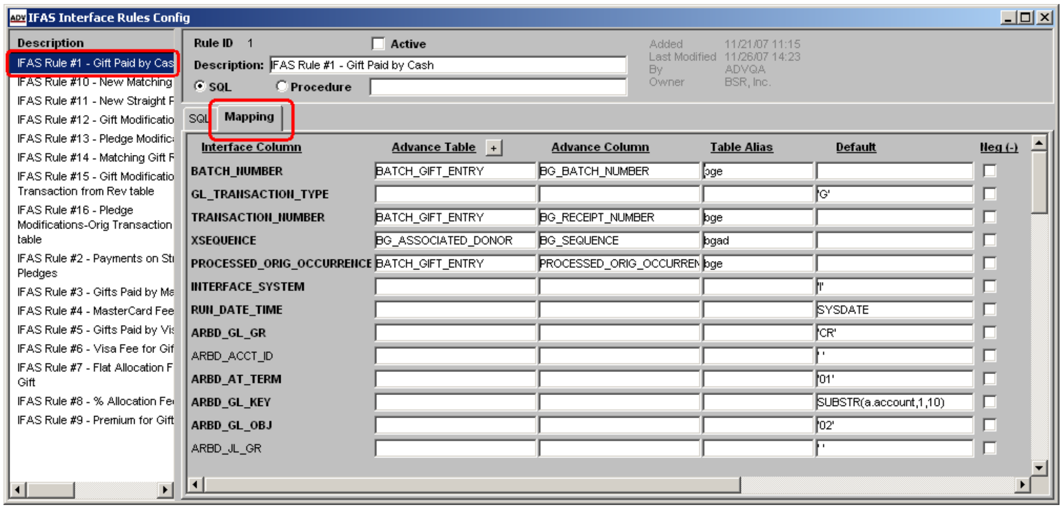
Description: IFAS Rule #4 - MasterCard Fee for Gift

○ SQL ● Procedure apb_ifas_rule_4

Added	11/21/07 14:14
Last Modified	11/26/07 17:37
By	ADVQA
Owner	BSR, Inc.

Figure 41: IFAS Interface Rules Coding

When the Mapping tab is selected for the rule, the appropriate data mappings are displayed on the right side of the window. Required columns display in **bold**.



IFAS Interface Rules Config

Description
IFAS Rule #1 - Gift Paid by Cas
IFAS Rule #10 - New Matching
IFAS Rule #11 - New Straight F
IFAS Rule #12 - Gift Modificatio
IFAS Rule #13 - Pledge Modific
IFAS Rule #14 - Matching Gift F
IFAS Rule #15 - Gift Modificatio Transaction from Rev table
IFAS Rule #16 - Pledge Modifications-Orig Transaction table
IFAS Rule #2 - Payments on Str Pledges
IFAS Rule #3 - Gifts Paid by Ma
IFAS Rule #4 - MasterCard Fee
IFAS Rule #5 - Gifts Paid by Vis
IFAS Rule #6 - Visa Fee for Gif
IFAS Rule #7 - Flat Allocation F Gift
IFAS Rule #8 - % Allocation Fe
IFAS Rule #9 - Premium for Gift

Rule ID 1 ☐ Active

Description: IFAS Rule #1 - Gift Paid by Cash

● SQL ○ Procedure

Added 11/21/07 11:15
Last Modified 11/26/07 14:23
By ADVQA
Owner BSR, Inc.

SQL | Mapping

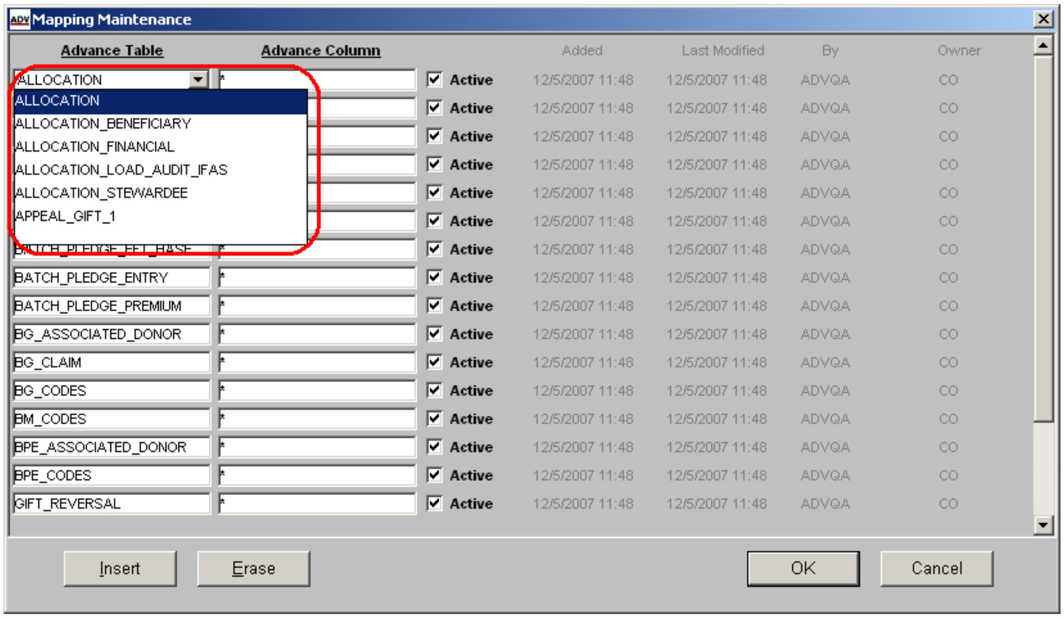
Interface Column	Advance Table +	Advance Column	Table Alias	Default	Neg (-)
BATCH_NUMBER	BATCH_GIFT_ENTRY	BG_BATCH_NUMBER	bge		☐
GL_TRANSACTION_TYPE				'G'	☐
TRANSACTION_NUMBER	BATCH_GIFT_ENTRY	BG_RECEIPT_NUMBER	bge		☐
XSEQUENCE	BG_ASSOCIATED_DONOR	BG_SEQUENCE	bgad		☐
PROCESSED_ORIG_OCCURRENCE	BATCH_GIFT_ENTRY	PROCESSED_ORIG_OCCURREN	bge		☐
INTERFACE_SYSTEM				'I'	☐
RUN_DATE_TIME				SYSDATE	☐
ARBD_GL_GR				'CR'	☐
ARBD_ACCT_ID				' '	☐
ARBD_AT_TERM				'01'	☐
ARBD_GL_KEY				SUBSTR(a.account,1,10)	☐
ARBD_GL_OBJ				'02'	☐
ARBD_JL_GR				' '	☐

Figure 42: IFAS Interface Rules Coding - Mapping

3. To create your own rule, click New on the tool bar and enter the information for the rule, i.e., a name for the rule, SQL or Procedure information, then, click the Mapping Tab.

ⓘ *Only the FROM and WHERE clauses and their associated logic should be entered in the SQL area. The combination of what you enter, combined with system coding, results in the complete SQL statement.*

4. To add mapping information, click + at the top of the Mapping tab form.



Mapping Maintenance

Advance Table	Advance Column		Added	Last Modified	By	Owner
ALLOCATION	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
		☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
		☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
		☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
		☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
		☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BATCH_PLEDGE_FEE_BASE	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BATCH_PLEDGE_ENTRY	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BATCH_PLEDGE_PREMIUM	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BG_ASSOCIATED_DONOR	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BG_CLAIM	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BG_CODES	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BM_CODES	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BPE_ASSOCIATED_DONOR	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
BPE_CODES	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO
GIFT_REVERSAL	*	☑ Active	12/5/2007 11:48	12/5/2007 11:48	ADVQA	CO

ALLOCATION
ALLOCATION_BENEFICIARY
ALLOCATION_FINANCIAL
ALLOCATION_LOAD_AUDIT_IFAS
ALLOCATION_STEWARDEE
APPEAL_GIFT_1

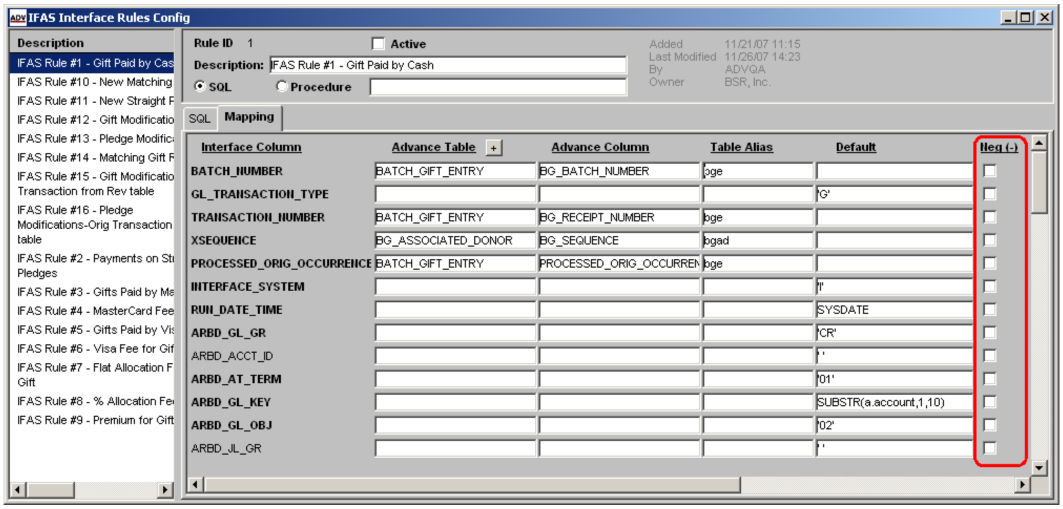
Figure 43: Mapping Maintenance

5. Select the fields you wish to map from the drop-down list, as shown in Figure 43. You can add or remove rows using the Insert and Erase buttons.
6. Click OK when done.
7. Optionally, when you return to the Mapping Maintenance window, enter a Table Alias and Default value for the rows. To provide greater flexibility, the following six variables are available for you to use as defaults:

Name	Description
value1	number(14,2)
value2	number(14,2)
value3	number(14,2)
text1	varchar2(60)
text2	varchar2(60)
text3	varchar2(60)

ⓘ *The value1, value2, and value3 columns allow you to feed calculated totals, such as premium and fee amounts, which are not stored elsewhere in Advance. The text1, text2, and text3 columns allow you to feed any additional data.*

8. The Negative (-) indicator, when checked for a row, allows you to pass negative values, e.g., for reversals, modifications, premiums and fees.



IFAS Interface Rules Config

Description
- IFAS Rule #1 - Gift Paid by Cas
- IFAS Rule #10 - New Matching
- IFAS Rule #11 - New Straight F
- IFAS Rule #12 - Gift Modificatio
- IFAS Rule #13 - Pledge Modific
- IFAS Rule #14 - Matching Gift F
- IFAS Rule #15 - Gift Modificatio Transaction from Rev table
- IFAS Rule #16 - Pledge Modifications-Orig Transaction table
- IFAS Rule #2 - Payments on Str Pledges
- IFAS Rule #3 - Gifts Paid by Ma
- IFAS Rule #4 - MasterCard Fee
- IFAS Rule #5 - Gifts Paid by Vis
- IFAS Rule #6 - Visa Fee for Gif
- IFAS Rule #7 - Flat Allocation F Gift
- IFAS Rule #8 - % Allocation Fe
- IFAS Rule #9 - Premium for Gift

Rule ID 1 □ Active

Description: IFAS Rule #1 - Gift Paid by Cash

● SQL ○ Procedure

Added 11/21/07 11:15
Last Modified 11/26/07 14:23
By ADVQA
Owner BSR, Inc.

SQL | Mapping

Interface Column	Advance Table	Advance Column	Table Alias	Default	Neg (-)
BATCH_NUMBER	BATCH_GIFT_ENTRY	BG_BATCH_NUMBER	bge		□
GL_TRANSACTION_TYPE				'G'	□
TRANSACTION_NUMBER	BATCH_GIFT_ENTRY	BG_RECEIPT_NUMBER	bge		□
XSEQUENCE	BG_ASSOCIATED_DONOR	BG_SEQUENCE	bgad		□
PROCESSED_ORIG_OCCURRENCE	BATCH_GIFT_ENTRY	PROCESSED_ORIG_OCCURREN	bge		□
INTERFACE_SYSTEM				'I'	□
RUN_DATE_TIME				SYSDATE	□
ARBD_GL_GR				'CR'	□
ARBD_ACCT_ID				' '	□
ARBD_AT_TERM				'01'	□
ARBD_GL_KEY				SUBSTR(a.account,1,10)	□
ARBD_GL_OBJ				'02'	□
ARBD_JL_GR				' '	□

Figure 44: Indicator for passing negative values

9. Check the □ Active check box to make this an active rule when you finish your edits.
10. Click Save to save your changes.

How to run the IFAS Financial Interface

The IFAS Financial Interface is launched by scheduling a task on your web server to start:

http://localhost/awa/processfinancialinterface.aspx

Any database process errors are logged in the zz_proc_log table. .Net process errors are logged in the AWALog Windows event log.

🛠 *Please note that setting up this interface will require many 'trial and error' checks based on the IFAS accounting rules and the mapping between the Advance and IFAS fields at your institution or organization.*

6 JAWS® Screen Reader Configuration

This section describes the JAWS configuration settings SunGard Higher Education used during development of Advance Web 9.1.1. In order to produce an optimal JAWS system configuration for use with Advance Web, SunGard Higher Education recommends that you implement the JAWS settings described below. However, these settings are only recommendations and are not required. Each institution or organization may, at its own discretion, change any or all JAWS settings in a manner that fits their policies and procedures.

ⓘ *Advance Web 9.1.1 and higher were developed specifically for use with JAWS version 8.0.1177.*

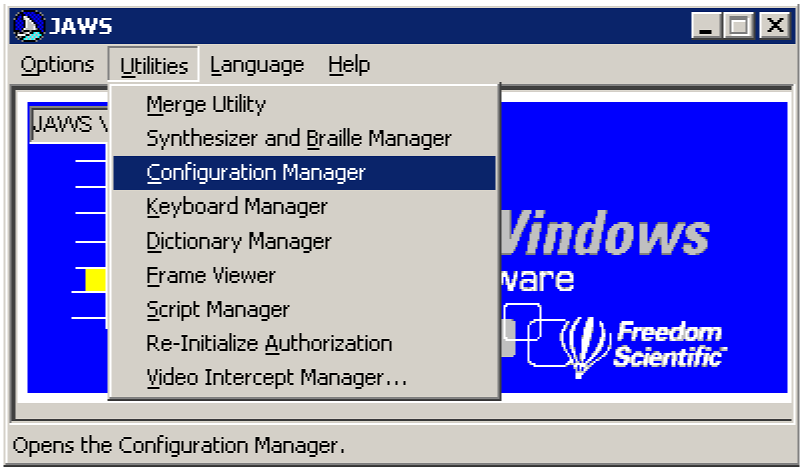
JAWS Utilities

Initial Configuration Manager settings can be established by using JAWS and following the instructions below.

Configuration Manager

The Configuration Manager provides access to various types of JAWS configuration options. To display the Configuration Manager:

1. Select Utilities → Configuration Manager and a new Configuration Manager window will open.



Opens the Configuration Manager.

Figure 45: Opening JAWS Configuration Manager

User Options

To prevent JAWS from repeating (echoing) values that are highlighted, a default setting in the Configuration Manager must be changed. This is accomplished in the User Options window.

To modify User Options:

1. Select Set Options → User Options.

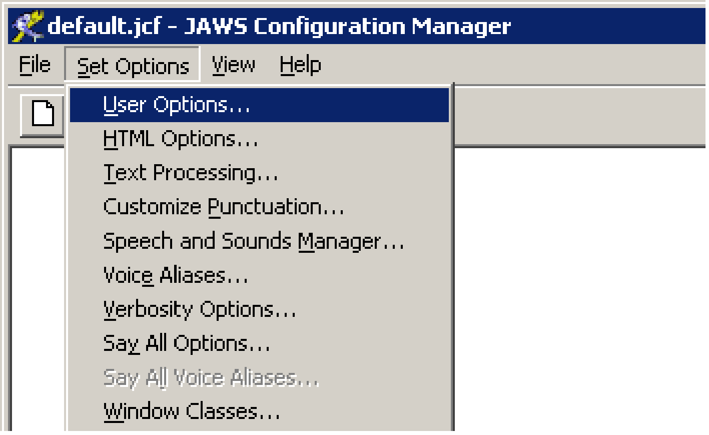
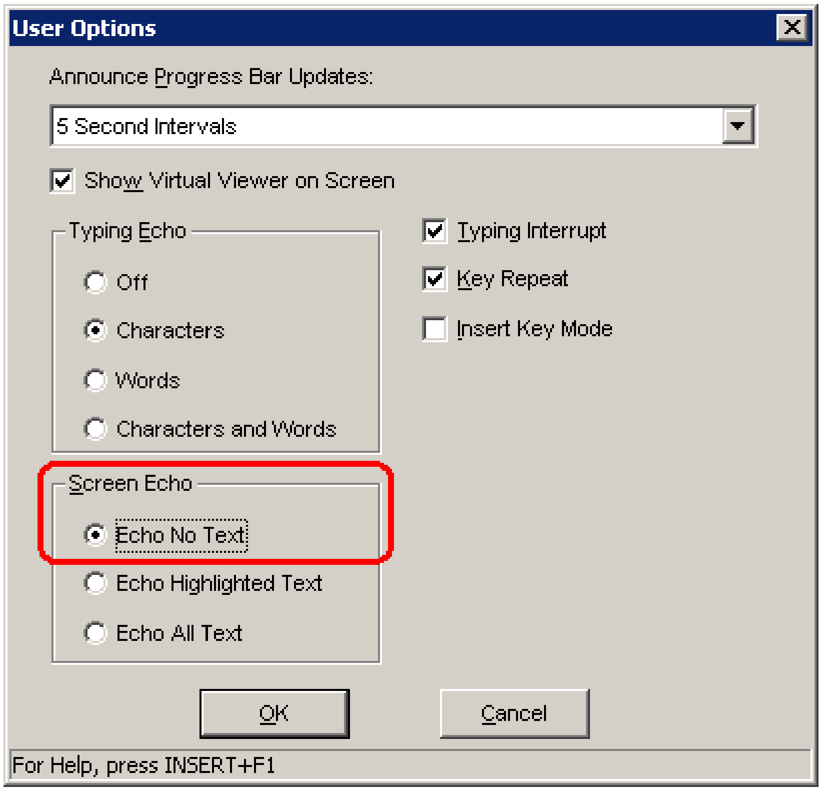


Figure 46: JAWS User Options

2. In the User Options window, select the 'Echo No Text' radio button under the Screen Echo option.


User Options

Announce Progress Bar Updates:

5 Second Intervals

☑ Show Virtual Viewer on Screen

Typing Echo

○ Off

● Characters

○ Words

○ Characters and Words

☑ Typing Interrupt

☑ Key Repeat

☐ Insert Key Mode

Screen Echo

● Echo No Text

○ Echo Highlighted Text

○ Echo All Text

For Help, press INSERT+F1

Figure 47: JAWS Screen Echo

3. Click OK to close the User Options window with your changes.

HTML Options

To modify HTML Options:

1. Select Set Options → HTML Options.

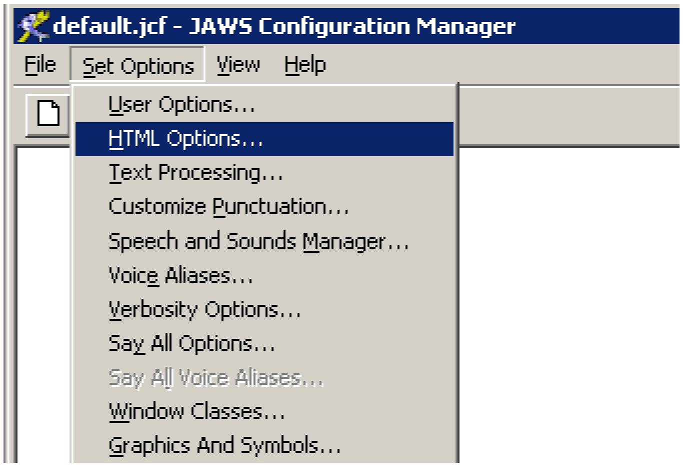
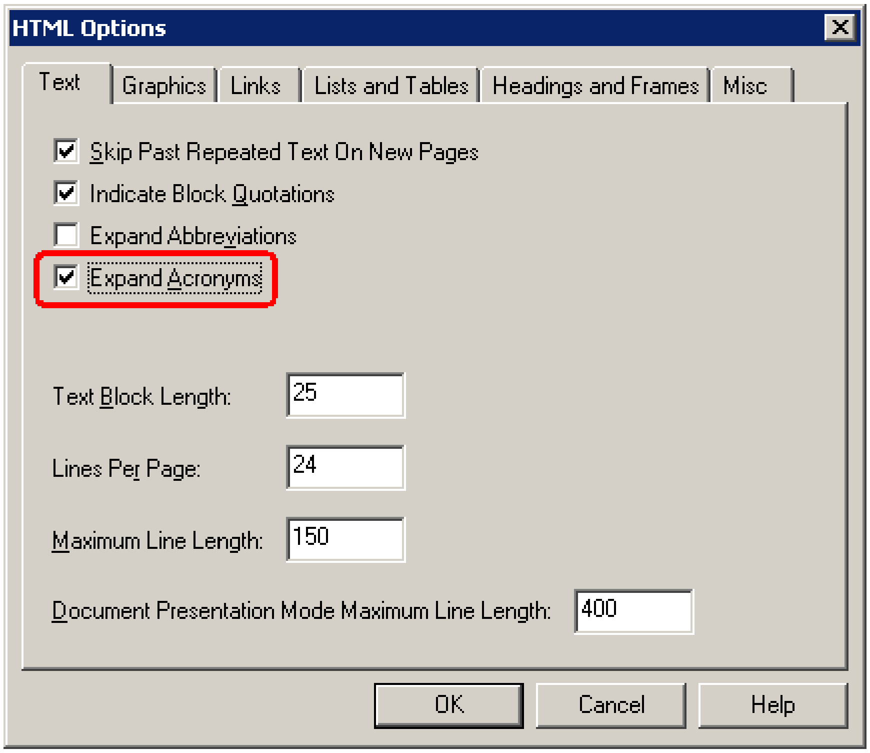


Figure 48: JAWS HTML Options

2. On the Text tab, check the 'Expand Acronyms' check box. Checking this check box will enable the functionality that performs lookups for abbreviations and acronyms so the full word is provided.

ⓘ *Please note that this will result in a difference, in some cases, as to what is read by JAWS versus the label that is displayed on the Web page.*



☑ Skip Past Repeated Text On New Pages

☑ Indicate Block Quotations

☐ Expand Abbreviations

☑ Expand Acronyms

Text Block Length: 25

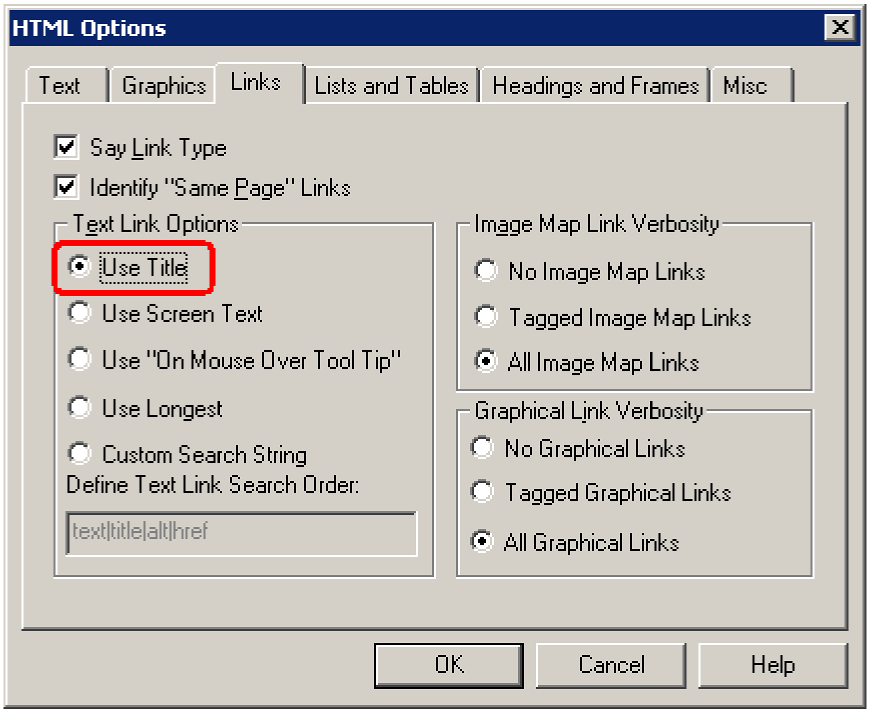
Lines Per Page: 24

Maximum Line Length: 150

Document Presentation Mode Maximum Line Length: 400

Figure 49: Expand Acronyms

3. On the Links tab, select the ‘Use Title’ radio button from within the Text Link options section. Selecting the ‘Use Title’ radio button ensures all special titles provided for links are read by the screen reader, e.g., special titles assigned for column headers that are also links that provide sorting functionality.



☑ Say Link Type

☑ Identify "Same Page" Links

Text Link Options

● Use Title

○ Use Screen Text

○ Use "On Mouse Over Tool Tip"

○ Use Longest

○ Custom Search String

Define Text Link Search Order:

text|title|alt|href

Image Map Link Verbosity

○ No Image Map Links

○ Tagged Image Map Links

● All Image Map Links

Graphical Link Verbosity

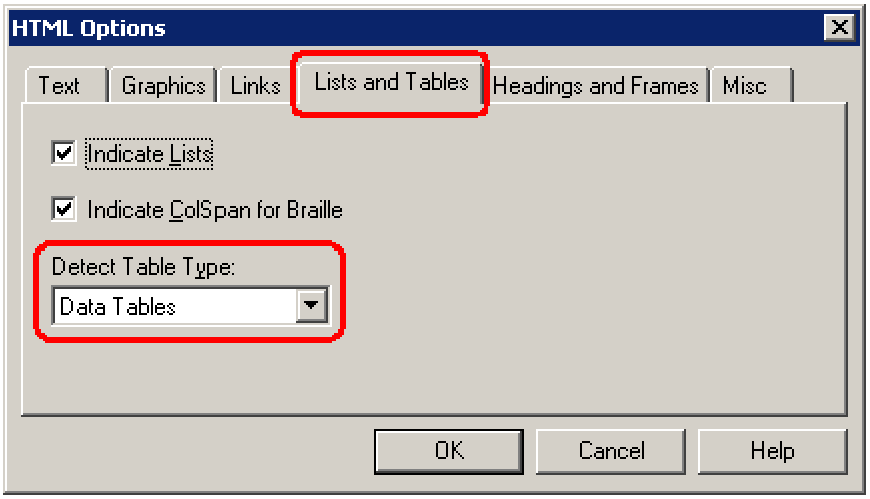
○ No Graphical Links

○ Tagged Graphical Links

● All Graphical Links

Figure 50: Links - Use Title

4. On the Lists and Tables tab, confirm the Detect Table Type is set to Data Tables, which is the default setting.



☑ Indicate Lists

☑ Indicate ColSpan for Braille

Detect Table Type:

Data Tables

Figure 51: Lists and Tables - Detect Table Type

ⓘ *Advance Web makes extensive use of tables for both data and formatting purposes. In order to filter text that has been formatted via tables, as opposed to the more traditional data tables for which a JAWS user would use table commands, SunGard Higher Education recommends setting this option to Data Tables.*

Important JAWS Information

This section provides additional notes regarding JAWS and its interaction with Advance Web.

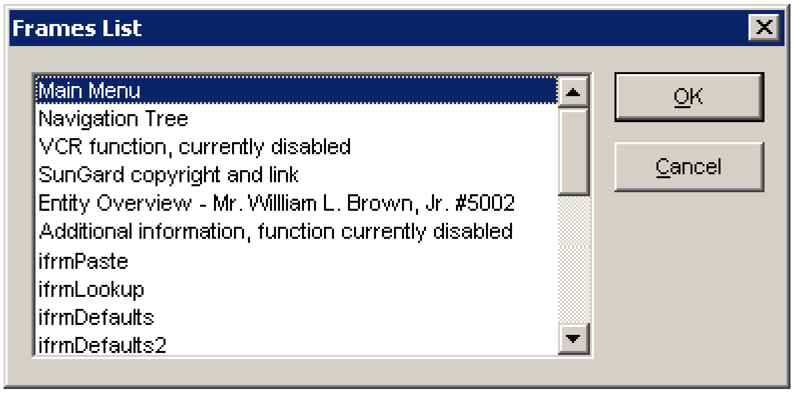
Response Time Impact on JAWS List Results

You may occasionally notice a delay in the time it takes JAWS to cache web pages, especially those that contain a large amount of information. Therefore, if you perform a JAWS list command prior to the page being fully loaded, the List dialog box may be incomplete.

For example, if you have access to a large number of Go To applications, there may be a slight delay in the availability of forms via the JAWS “Links List.” If the values you are expecting to display in the JAWS “Links List” window are not immediately available, wait a few seconds, then execute the JAWS “List Links” command again to refresh the list.

Frames List in Forms Mode

If a JAWS end-user performs the JAWS “List Frames” command while in “Forms Mode,” they may receive unexpected results. This is consistent with JAWS behavior in that certain JAWS commands cannot be performed successfully when in “Forms Mode.” This might include Frame names being listed multiple times as well as some frame names that start with “ifrm” as illustrated below. SunGard Higher Education recommends only executing the List Frames command when “Forms Mode” is turned off, per standard JAWS usage. Performing this action will provide a standard and more accurate “Frames List.”



Main Menu
Navigation Tree
VCR function, currently disabled
SunGard copyright and link
Entity Overview - Mr. Willliam L. Brown, Jr. #5002
Additional information, function currently disabled
ifrmPaste
ifrmLookup
ifrmDefaults
ifrmDefaults2

Figure 52: JAWS Frames List

Advance Configuration Utility Setup

This section describes setup procedures that must be performed via the Advance Configuration Utility, independent of the aforementioned JAWS configuration. Unlike the JAWS setup that must be performed on each individual machine, the procedures in this section only need to be performed in the Configuration Utility via one machine. The only requirement is the Configuration Utility for Advance must be current for your version of Advance Web (do not use prior versions of the Configuration Utility), must be installed on the machine and you must have a working connection from that machine to the Advance database.

The procedures described in this section only need to be performed once, upon initial installation. From that point forward, you will only need to perform these procedures if you wish to add or remove accessibility users from the system, or install a second instance (production, test or training) of Advance Web.

Accessibility Profile Maintenance

The term Profile is used to describe a group or area within your organization, or your institution or organization as a whole. When configurations are made to Pages, Forms and Applications in Advance Web, they are applied to specific profiles. Thus, you can configure Advance Web to appear one way for users that are associated with one profile and another way for users that are associated with a different profile.

Advance Web contains four different levels of profiles. The first three, Site, Department and Role, are used to categorize your entire organization, schools or programs within your organization, and responsibilities at your organization, respectively. The fourth profile level is used to identify Accessibility users that will use the JAWS screen reader with Advance Web.

When a user is added to the Accessibility profile, Advance Web will recognize them as an Accessibility user. When a user in the Accessibility profile logs into Advance Web, certain behavior will occur in a manner consistent and appropriate for Accessibility users. This behavior is described starting on page 100.

ⓘ *While adding a user to the Accessibility profile identifies them as an accessibility user, it does not identify them as part of a profile that will incur additional Department or Role changes. If you wish to implement profile-specific changes at the form, page, application, etc. level, you must ensure those Accessibility users are also included in the appropriate Department or Role profile. These profiles must be created as part of the initial implementation of Advance Web, based on the various needs of your institution or organization.*

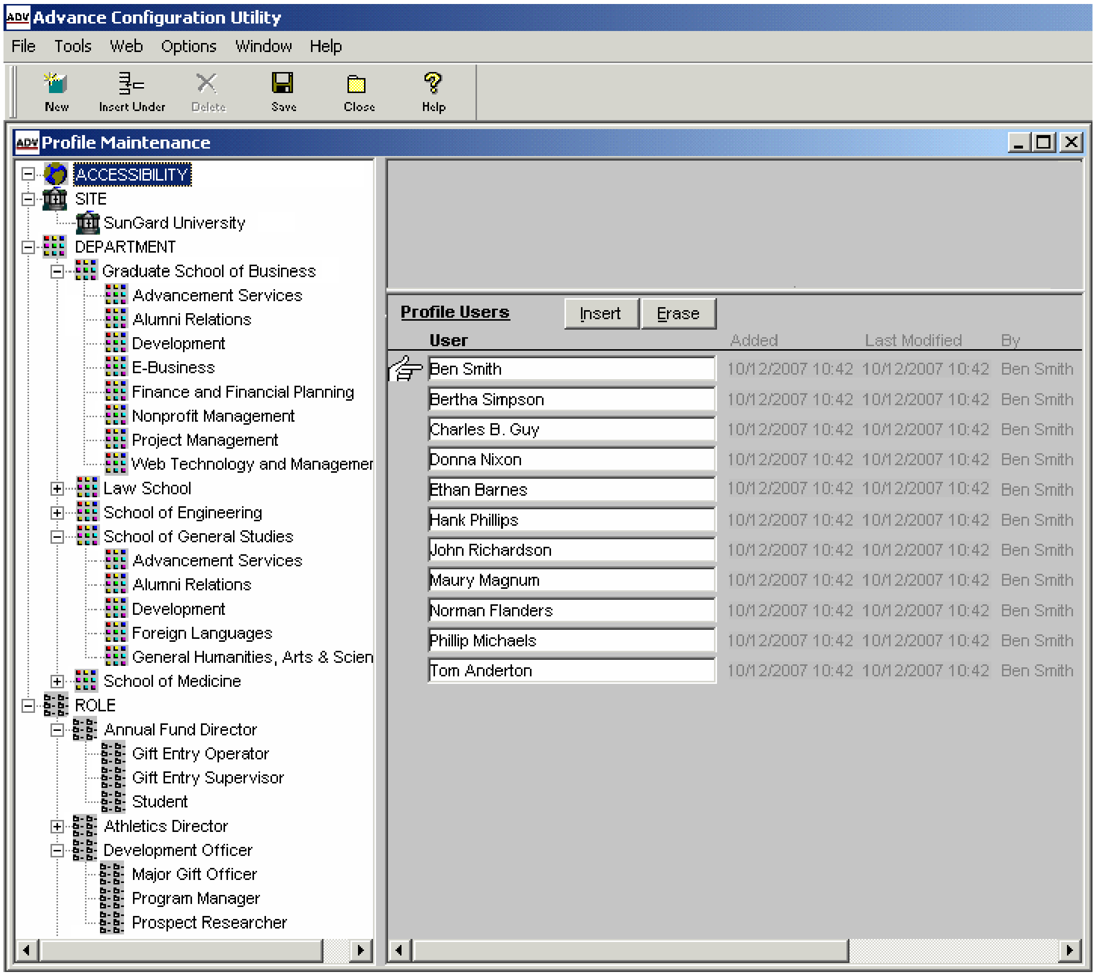
To access the Profile Maintenance window and add or remove Accessibility Users from the Accessibility Profile:

1. Open the Advance Configuration Utility and select Web→Web Admin→Profiles.
2. Select the 'ACCESSIBILITY' profile in the tree on the left.
3. On the right side of the window, click 'Insert' to add a user record.
4. Select the name of the Accessibility user from the list of users.

   ⓘ *If a user's name does not appear in the drop-down list, you must navigate to Tools→Security→Users and first create the user.*

5. Click 'Save' to save your changes.

To remove an accessibility user from the list, select their user name from the list on the right, click 'Erase', then click 'Save.'



- ACCESSIBILITY
- SITE
  - SunGard University
- DEPARTMENT
  - Graduate School of Business
    - Advancement Services
    - Alumni Relations
    - Development
    - E-Business
    - Finance and Financial Planning
    - Nonprofit Management
    - Project Management
    - Web Technology and Managemer
  - Law School
  - School of Engineering
  - School of General Studies
    - Advancement Services
    - Alumni Relations
    - Development
    - Foreign Languages
    - General Humanities, Arts & Scien
  - School of Medicine
- ROLE
  - Annual Fund Director
    - Gift Entry Operator
    - Gift Entry Supervisor
    - Student
  - Athletics Director
  - Development Officer
    - Major Gift Officer
    - Program Manager
    - Prospect Researcher

User	Added	Last Modified	By
Ben Smith	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Bertha Simpson	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Charles B. Guy	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Donna Nixon	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Ethan Barnes	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Hank Phillips	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
John Richardson	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Maury Magnum	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Norman Flanders	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Phillip Michaels	10/12/2007 10:42	10/12/2007 10:42	Ben Smith
Tom Anderton	10/12/2007 10:42	10/12/2007 10:42	Ben Smith

Figure 53: Accessibility Profile Users Maintenance

The Accessibility profile contains a zz_mt_profile.profile_id code of -1. This value is not visible in Advance Web and is only pertinent to system administrators and technical support staff responsible for system maintenance.

Using Advance Web

This section focuses on navigation techniques that describe how Advance Web was developed (designed and tested) for use with JAWS. While the primary focus of this section is to describe the SunGard Higher Education developed and tested navigation techniques, it is important to keep in mind that you may develop your own techniques for maneuvering around. Together these techniques you develop and the techniques described in this document will help to provide the tools necessary to effectively use Advance Web.

Navigation

Orientation

After you successfully log into Advance Web, the Home Page will be presented along with standard information provided by JAWS that counts for Frames, Headings and Links.

The Home page provides you with logon information, messages, tasks, and hyperlinks to entities associated with your logon, either because they are assigned to you or because you have recently viewed them. Essentially, everything you need to perform your daily duties starts at the Home page. This is different from Go To, which acts as a Site Map that allows you to navigate to areas that allow you to perform your duties. Use of Go To with JAWS is described on page 105.

Over the next several pages, this document describes each of the components of the Home Page, as well as other components of Advance Web and their interaction with JAWS.

Page Tree

The Page Tree, on the top left of the page, is one of several navigation tools for use with Advance Web. The elements in the Page Tree are defined as clickable elements.

To access an item in the Page Tree, use one of the following techniques:

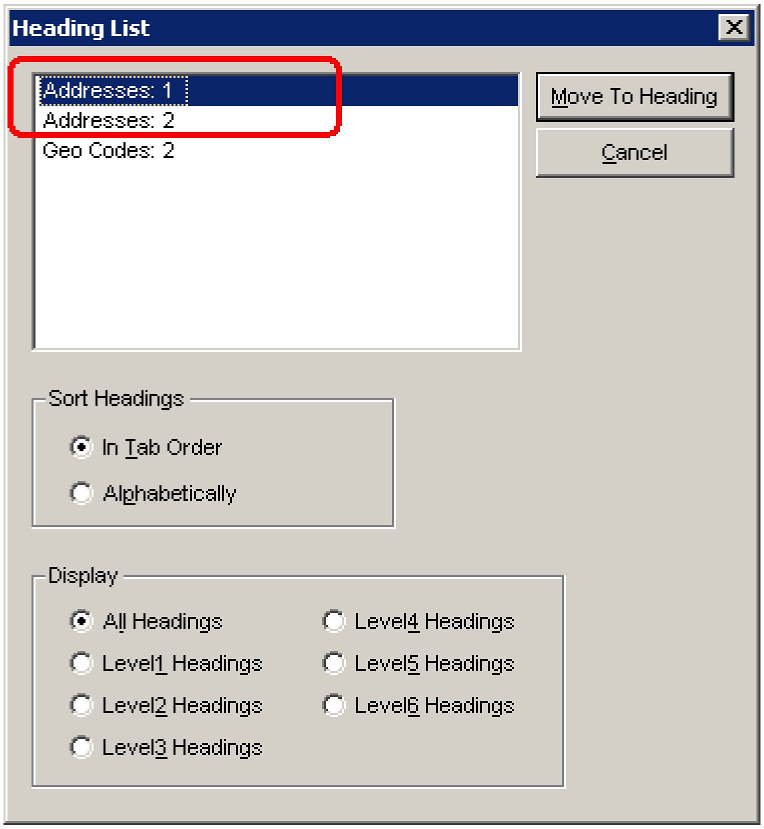
- Use the JAWS “List clickable element” command to access the list of clickable elements on the page. After you select the desired clickable element by pressing Enter, press Enter a second time to activate the element.
- Use the JAWS “Move to next clickable element” command to navigate through the various clickable elements and press Enter when the applicable element is selected.

As you navigate through Advance Web, the Page Tree will change and display information relevant to your general location. For example, if you navigate to the Biographic Summary form, the Page Tree will display Entity-related information, followed by a slash and the term “Biographic,” to indicate you are in the Biographic subsystem. The term that follows the slash will change as you navigate through the various subsystems, to reflect the subsystem to which you are current. If an item in the Page Tree has at least one associated record, the label in the page tree for this item will also contain a count of records to indicate how many records, which will be read by JAWS.

Application and Form Headers

Application Headers display at the top of the Application Frame and contain the title of the application. To distinguish Application and Form Headers, all headers contain heading numbers to uniquely identify them and allow JAWS to recognize the hierarchy on the page. Application Headers always contain a heading level of 1 and Form Headers always contain a heading level of 2. This is of particular importance to help distinguish between Application Headers and Form Headers that contain the same title.

For example, the Addresses Form contains an Application Header of Addresses, and the first Form on the page contains a Form Header of Addresses. When you use the JAWS “List Headers” command, both items will appear. The Application Header will contain the number 1 and the Form Header will contain the number 2.



Addresses: 1
Addresses: 2
Geo Codes: 2

Sort Headings

● In Tab Order
○ Alphabetically

Display

● All Headings
○ Level1 Headings
○ Level2 Headings
○ Level3 Headings
○ Level4 Headings
○ Level5 Headings
○ Level6 Headings

Figure 54: JAWS Heading List

To navigate between different headers on the page:

- Use the JAWS “List Headers” command to access the Heading List. The Application Header will always be the first heading in the list, followed by all other headings in the order in which they appear on the screen.

  ⓘ *In addition to the functionality described above, the “Skip to Application Header” link on the Main Menu allows you to navigate directly to the Application Header, thereby bypassing the Heading List.*

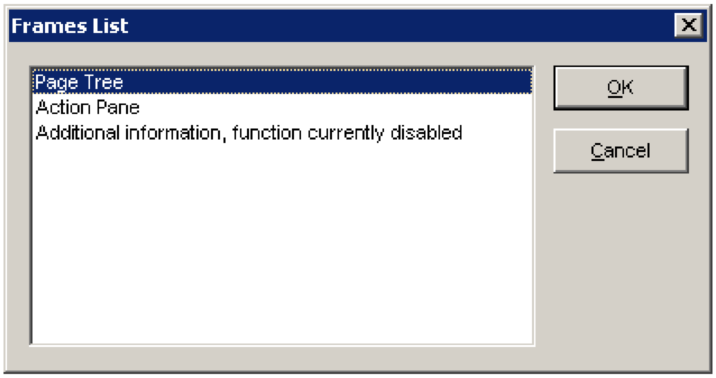
Main Menu

The Main Menu frame includes numerous Navigation Controls that provide access to areas (Home, Logoff, etc.) and tools (Go To, Lookups, Reports, Back, Forward, Refresh) as well as a list of Currently Open Applications. All items in the Main Menu frame are also links, with the exception of the Open Applications List combo box.

Using your browser's Forward, Back, and Refresh buttons may result in unexpected behavior, so use of the Navigation Controls in the Main Menu is highly recommended.

To access the Main Menu frame, use one of the following techniques:

- Use the JAWS "List Frames" command to access the list of Frames. Select the Main Menu frame and your cursor will be positioned in the Main Menu frame. Use the Tab key to move through items in the frame.

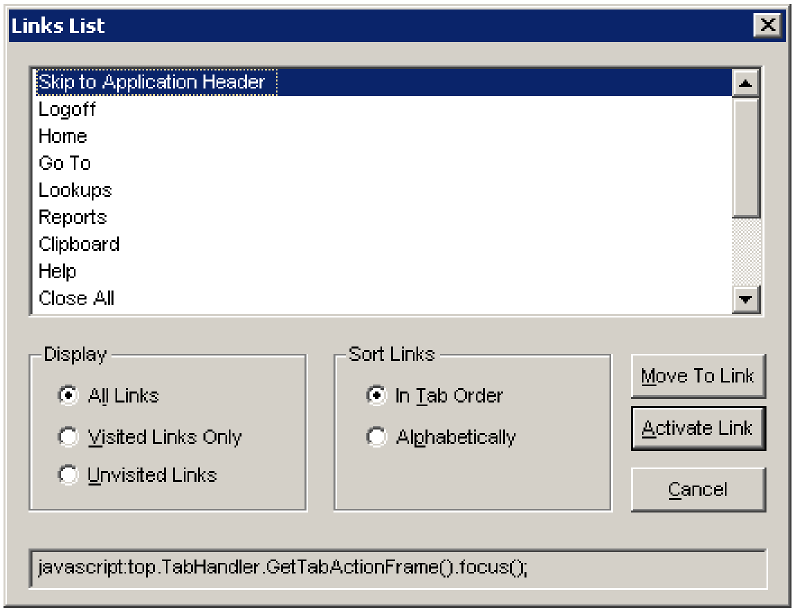


Frames List

Page Tree
Action Pane
Additional information, function currently disabled

Figure 55: JAWS Frames List

- Use the JAWS "List Links" command to access the list of Links and use arrow keys to move through the list. Once you navigate to the desired location, use any other navigation technique described in this document to continue. The first set of links will always represent the links in the Main Menu frame when you invoke the JAWS "List Links" command.



Links List

Skip to Application Header
Logoff
Home
Go To
Lookups
Reports
Clipboard
Help
Close All

Display
● All Links
○ Visited Links Only
○ Unvisited Links

Sort Links
● In Tab Order
○ Alphabetically

javascript:top.TabHandler.GetTabActionFrame().focus();

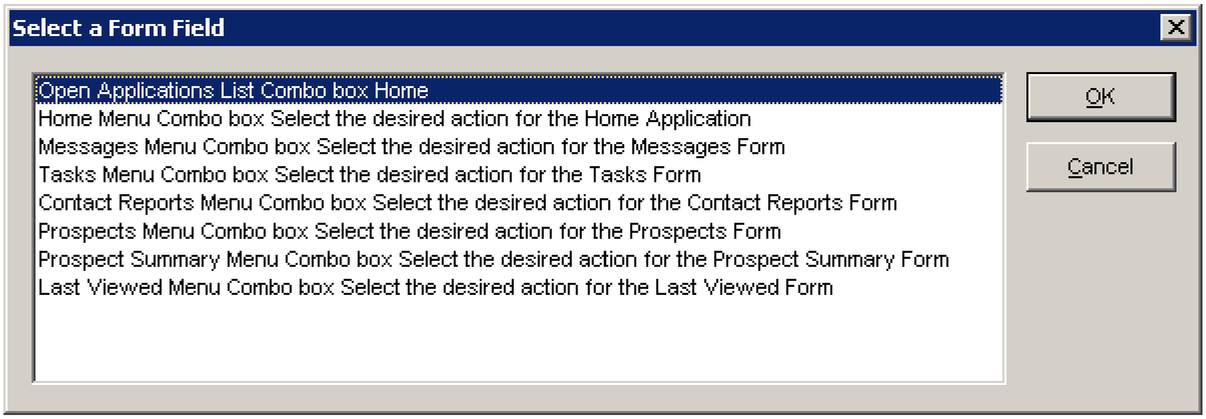
Figure 56: JAWS Links List

Open Application List

The Open Application List provides you with a navigation method to quickly access a list of open applications. These are applications you recently visited, but did not close.

To access the Open Application List, use one of the following methods:

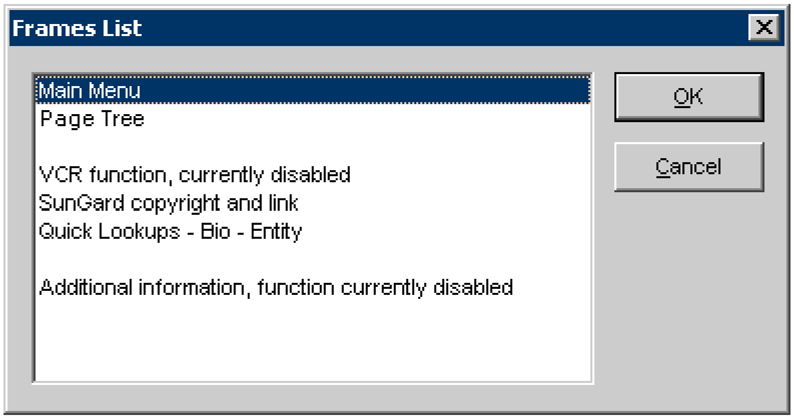
- Use the JAWS “List Form Fields” command. The first item on the list will begin with ‘Open Applications List combo box’. Select this value. Any additional text read by JAWS for this particular combo box should be ignored as JAWS may provide invalid information for the Open Applications combo box value.



Open Applications List Combo box Home
Home Menu Combo box Select the desired action for the Home Application
Messages Menu Combo box Select the desired action for the Messages Form
Tasks Menu Combo box Select the desired action for the Tasks Form
Contact Reports Menu Combo box Select the desired action for the Contact Reports Form
Prospects Menu Combo box Select the desired action for the Prospects Form
Prospect Summary Menu Combo box Select the desired action for the Prospect Summary Form
Last Viewed Menu Combo box Select the desired action for the Last Viewed Form

Figure 57: JAWS Select a Form Field

- Use the JAWS “List Frames” command to access the list of Frames. Use arrow keys to scroll through the list and select “Main Menu.” When Advance Web returns, press Tab to navigate to the Open Application List combo box. Use the normal combo box JAWS commands to work with this control.



Main Menu
Page Tree

VCR function, currently disabled
SunGard copyright and link
Quick Lookups - Bio - Entity

Additional information, function currently disabled

Figure 58: JAWS Frames List

If you are using Advance Web 9.4.0.0, you can use the following steps to navigate to tabs:

1. Perform the List Links (Insert + F7) command.
2. Scroll down to one of the open tab names in the list. Press Enter to navigate to the selected tab and the tab’s associated application will display.
3. List the headers (Insert + F6) and choose the application header.
4. Tab to the drop down box and press Enter to go into forms mode.

5. Use Alt + ↓ to move through the options in the combo box.
6. Select Close and press Enter to close the application.

Form Header Controls

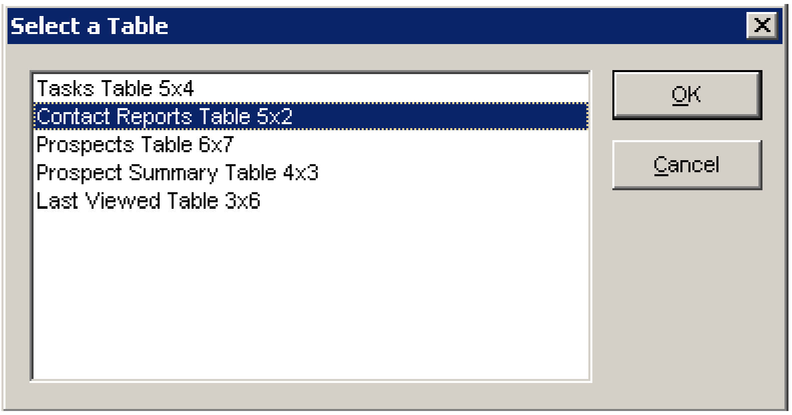
Form Headers, also known as Headings in JAWS, display at the top of a Form and contain the title of the form, along with any actions associated with the form. Form Header controls vary from form-to-form. Examples of what controls are available include, but may not be limited to Sort, Print, Export, Help, New, Save and Delete.

To access the Form Header list:

- Use the JAWS "List Headings" command to access the list of Headings. Use arrow keys to scroll through the list and select the desired heading. When Advance Web returns, press Tab to navigate to the Actions combo box and JAWS will state "Select the desired action for the form." Use the JAWS "Forms Mode" command to switch to forms mode. Use the standard combo box commands to select the desired action and proceed.

Tables

Advance Web uses tables as a primary method to format both forms with narrative as well as forms that contain data tables. Therefore, it is important that the default of 'Data Tables', as described earlier in the JAWS Screen Reader Configuration section of this document, be maintained. Otherwise, it is likely that more tables than desired will be listed when the JAWS List Tables command is invoked.



Select a Table

Tasks Table 5x4
Contact Reports Table 5x2
Prospects Table 6x7
Prospect Summary Table 4x3
Last Viewed Table 3x6

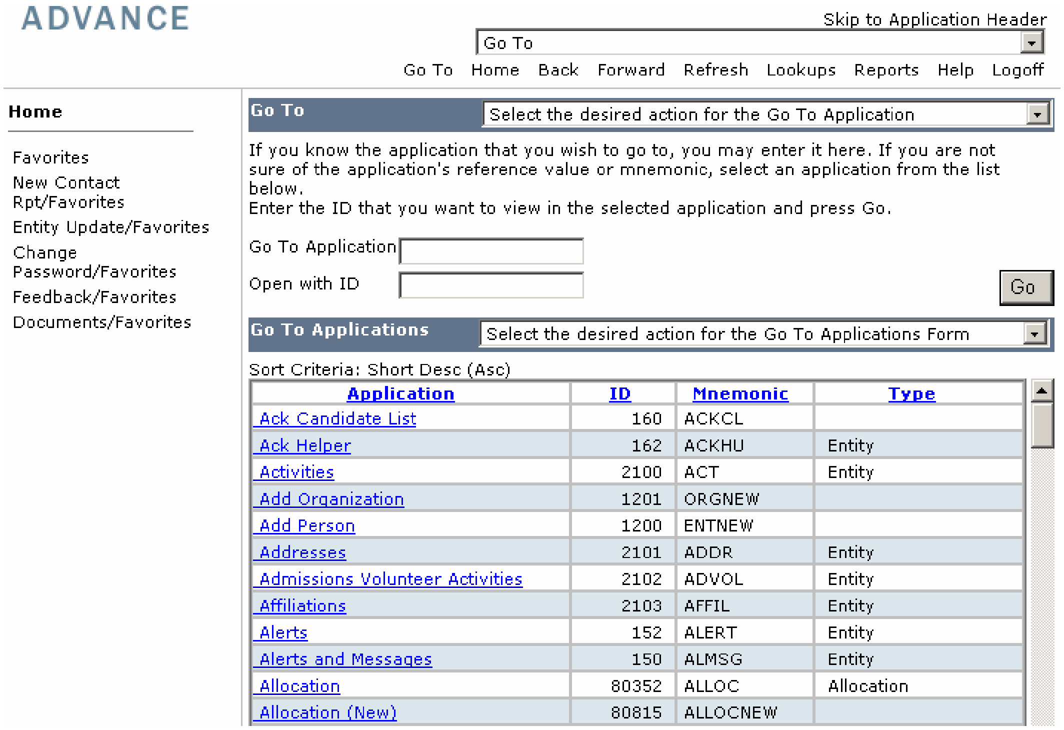
Figure 59: Select a Table

Go To

Go To acts as a site map that allows you to navigate to various areas in Advance Web.

To access and use Go To:

1. While in Forms Mode, use the JAWS “Links List” command to access the Links List and select Go To. It is important to be in “Forms Mode’ prior to accessing this application. If not, unexpected results may occur.
2. When Go To appears, the cursor will be placed in the “Go To Application” field. This is where you can specify the application you wish to access. If you know the mnemonic or application ID, enter it, otherwise access the Links List in order to invoke the list of available applications.
3. After you select the application, press Tab and the cursor will be placed in the “Input ID Field,” which is where you can specify the ID for which you wish to access the application. If you know the ID number, enter it, otherwise, press F2 to invoke the Lookup Application and locate the ID. JAWS will indicate when a value is required and that F2 can be used to perform a lookup. When you press F2, you navigate to, and the cursor is positioned on, the appropriate lookup form to perform the lookup.
4. After you locate the ID number and return to Go To, press Tab to navigate to the Go button, then press Enter to continue. If an ID number is not required for the selected application, you can navigate directly to the application with just the Application ID or mnemonic.


ADVANCE
Skip to Application Header
Go To
Go To
Home
Back
Forward
Refresh
Lookups
Reports
Help
Logoff
Home
Favorites
New Contact Rpt/Favorites
Entity Update/Favorites
Change Password/Favorites
Feedback/Favorites
Documents/Favorites
Go To
Select the desired action for the Go To Application
Go
Go To Applications
Select the desired action for the Go To Applications Form

If you know the application that you wish to go to, you may enter it here. If you are not sure of the application's reference value or mnemonic, select an application from the list below.

Enter the ID that you want to view in the selected application and press Go.

Go To Application

Open with ID

Sort Criteria: Short Desc (Asc)

Application	ID	Mnemonic	Type
Ack Candidate List	160	ACKCL	
Ack Helper	162	ACKHU	Entity
Activities	2100	ACT	Entity
Add Organization	1201	ORGNEW	
Add Person	1200	ENTNEW	
Addresses	2101	ADDR	Entity
Admissions Volunteer Activities	2102	ADVOL	Entity
Affiliations	2103	AFFIL	Entity
Alerts	152	ALERT	Entity
Alerts and Messages	150	ALMSG	Entity
Allocation	80352	ALLOC	Allocation
Allocation (New)	80815	ALLOCNEW	

Figure 60: Go To Applications

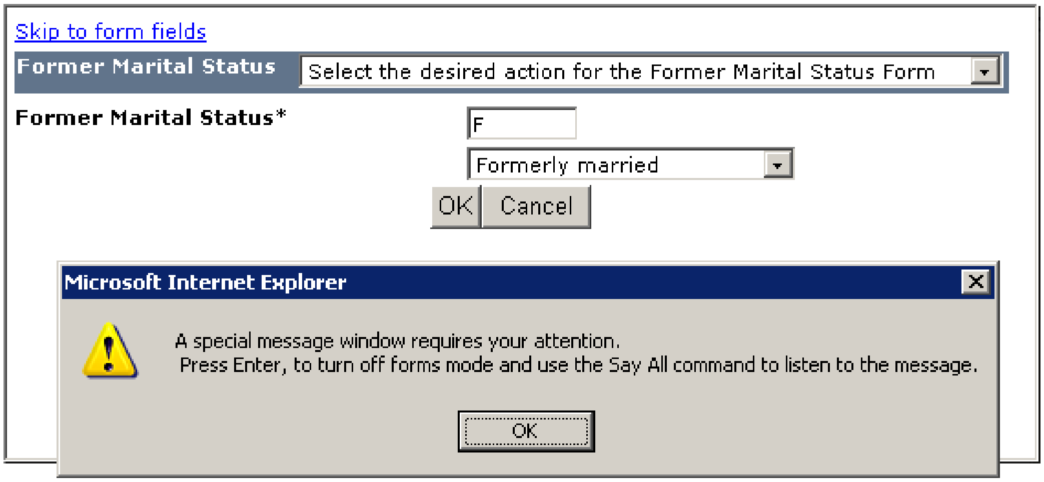
Please note the following in regard to Go To:

- Depending on the number of Go To applications to which you have access, there may or may not be a slight delay in all of the forms being available in the JAWS "Links List" window. If the application you are interested in is not immediately available, execute the List Links command again to refresh the list. This delay in display may also be true for any other application in Advance Web that may have a large number of items displayed in the Links List window.
- To navigate quickly to an area on the Links List, you may type the first letter of an application to go to the next occurrence of that letter at the beginning of a line, e.g., 'M' will navigate you to the beginning of applications beginning with 'M'.
- To navigate to the beginning of the Links List, press 'Home'. To navigate to the end of the Links List, press 'End'. This is per standard JAWS functionality.
- The 'Input ID Field' will change, depending on the selected Application. For example, if you select an application that requires an Entity ID, such as Addresses, you will need to enter an Entity ID. Likewise, if you select an application that requires a Prospect ID, such as Prospect Overview, you will need to enter a Prospect ID.

iFrames

Advance Web uses iFrames (Web browser dialog boxes) to indicate when feedback is required to complete an action. For example, when changing the marital status for a married entity, this type of dialog box appears and you are prompted to indicate what the marital status should be for the former spouse record. When the dialog box appears, JAWS will indicate a special message window requires your attention, followed by text specific to that form. To hear the text, press Enter to turn off forms mode and use the Say All command to listen to the message.

If the dialog box requires you to enter a value before proceeding, a "Skip to Form Fields" link will appear at the top of the form. When you select this link, you will be taken directly to the first entry field, thereby allowing you to bypass other information on the form.



Skip to form fields

Former Marital Status: Select the desired action for the Former Marital Status Form

Former Marital Status*: F

Formerly married

Microsoft Internet Explorer

A special message window requires your attention.
Press Enter, to turn off forms mode and use the Say All command to listen to the message.

Figure 61: 'Special message' dialog

Prospect Tracking Summary Hierarchy

The following are the suggested steps for navigating through the Prospect Information Hierarchy on the Prospect Tracking Summary application:

1. Access the Prospect Tracking Summary application from the Page Tree frame.
2. Use the JAWS List Headers command to access the Prospect Tracking Summary application header.
3. Press Tab to navigate to the Actions combo box.
4. Select the ‘Expand All’ action to expand the entire hierarchy.
5. Use JAWS table commands to navigate through the table and select the record and make the appropriate selections.

   ⓘ *As you navigate through the table, you can expand or collapse individual records by pressing Enter while current on a graphic in the first column of the table. If you press Enter to expand or collapse a particular record, JAWS focus is no longer on the table. Thus, you will need to use JAWS Table commands to re-position yourself back in the table, in the desired cell.*

Data Entry

Detail Forms

Detail forms are used to inquire and maintain details for a specific record. If you have maintenance rights, you may edit the detail.

To perform data entry on a Detail form, when one record or more already exists:

1. Use Go To or any other navigation method described in this document to select the desired application for which you wish to add the data.
2. When the detail form appears, the cursor will be placed in the first data entry field.
3. Depending on the navigation technique used to access the form, the first data entry field may or may not be announced. If the field is not announced, refresh the screen.
4. Enter information in all applicable and required fields.
5. Press F8 to Save the record. If the record is successfully saved, JAWS will re- position the cursor back in the first data entry field for the saved record and read the value in that field. If the record is not saved successfully, JAWS will read the applicable error message.

ⓘ *If you wish to use F6 to add the first record in a detail form, you must use the List Headings command to access the Form Header, then press Tab, followed by the F6 shortcut. This will ensure your cursor is positioned properly in the form so the F6 command can be recognized and performed.*

👁 *When working with forms that contain helpful text in the application header (heading level 1), e.g. Prospect, Proposals, Contact Report, etc., links in the helpful text may not be recognized by JAWS after you add or modify form details. You must refresh the page in order to ensure the links display properly.*

Master/Detail Forms

Master/Detail forms display the records of data available in the top portion of the form and the detail of the selected record in the bottom portion of the form. If you have maintenance rights, you may edit the detail.

To navigate through a Master/Detail form:

1. Use the JAWS "List Tables" command to access the master portion of the form.
2. Use JAWS "Table Navigation" commands to move through the cells to determine which record in the master list is the desired record, if selecting an existing record for update.
3. When on the desired record, press Tab to move to the button that indicates 'Press enter to select this record.'
4. Press Enter to select the record.
5. Use the tab key, F key or the JAWS "List form fields" command to navigate to the appropriate data entry field(s).

Multi-Master/Detail Forms

A few applications in Advance Web contain more than one detail form. For example, the Address Application contains the Address Master/Detail forms along with the Geo Codes master/detail form. When a record in the Master portion of the Address application is selected, the appropriate Address and Geo Code detail forms will display below. When a different Address record is selected, both the Address and Geo Code details will change accordingly.

When adding new records using the F6 shortcut key, Advance Web will perform the following logic to determine the form to which you wish to add the new record:

- If your cursor is in a particular detail form or positioned in a form's header, (in this example, Address or Geo Code) when F6 (Advance Web keyboard shortcut for adding a new record) is pressed, the system will assume you wish to enter a new record for the form on which your cursor is currently positioned.
- If the cursor is anywhere else on the page, it will search for the first 'New' action on the page. For example, if you press F6 on the Address application after selecting it from the Page Tree, it is assumed you wish to add a new Address.
- The system will also verify that the 'New' action is available for this particular form. If it is not, you will not be presented with the empty detail form.

To add a record in a Multi-Master/Detail record:

1. Use the JAWS "List Tables" command to access the master form and navigate through the records to ensure the record does not already exist.
2. Press F6 to insert a new record and complete data entry as necessary.
3. Press F8 to save the record.
4. Use the JAWS "List Headers" command to access the header record for the second detail form, e.g., after adding an initial task record and the associated task responsible information, perhaps a second task responsible record needs to be added. This is the second detail form.
5. Press Tab to navigate to the header's action combo box.
6. Press F6 to insert a new record and complete data entry as necessary.
7. Press F8 to save the record.

Drop-Down Combo Boxes

Data entry fields that contain drop-down combo boxes are contained in two separate fields. The first field contains a text box that allows you to enter the exact code, if known. The second field contains a drop-down list, which can be activated using standard JAWS commands. This dual approach allows data entry staff to choose whether they prefer to manually enter a code or navigate through the combo box using normal JAWS navigation techniques.

ⓘ *SunGard Higher Education does not recommend using the JAWS "Open Combo Box" (Alt + Down Arrow) command when navigating through combo boxes. Using this command will create an error condition because the corresponding code field for this combo box will not be populated.*

Miscellaneous

Error Messages

When you encounter an error while using JAWS, a popup will display the validation and/or error message feedback, and the content will be read by JAWS. If there are multiple errors, they will all display in this one popup message box and be read one at a time. These popups display in addition to the standard error and warning message text that is displayed within the detail form.

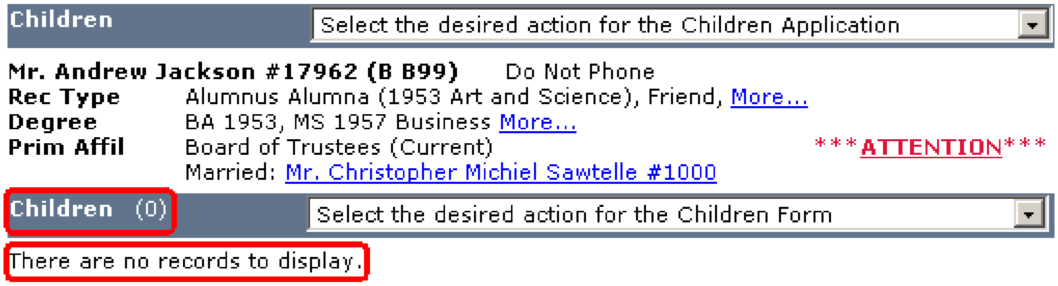
ⓘ *The error described above will not appear if you encounter an error during the logon process, as you have not yet been identified as an accessibility user. If this occurs, use the JAWS "Say All" command to hear the error.*

Empty Forms

When JAWS encounters a form that contains no records, it will read the form as usual, followed by the text "There are no records to display." It is expected that the JAWS user would use the "Say All" command to hear this text read.

For example, when you navigate to the Children Form for an entity that does not have children, JAWS will read the title "Children Form", followed by "There are no records to display." The form also displays the announced text below the header.

If the form typically contains a data table, the table may not appear when the List Tables command is invoked, as there is no table for JAWS to recognize and include in the list.



Children Select the desired action for the Children Application

Mr. Andrew Jackson #17962 (B B99) Do Not Phone
Rec Type Alumnus Alumna (1953 Art and Science), Friend, More...
Degree BA 1953, MS 1957 Business More...
Prim Affil Board of Trustees (Current) ***ATTENTION***
Married: Mr. Christopher Michiel Sawtelle #1000

Children (0) Select the desired action for the Children Form

There are no records to display.

Figure 62: 'No records' message

Keyboard Shortcuts

Advance Web contains numerous keyboard shortcuts that allow you to navigate through the application. For a list of keyboard shortcuts, please refer to Advance Web Navigation Guide.

In addition to standard Advance Web and JAWS keyboard shortcuts, please note that you must be in Forms Mode when working on the forms that allow the CTRL + S (activate the Associated Donors Hyperlink) and CTRL + K (activate the Acknowledgements Hyperlink) hyperlinks. This is necessary in order to ensure Advance Web keyboard shortcuts are performed. If you are not in forms mode, these will be interpreted as JAWS commands.

Performing Lookups

To perform a lookup:

1. Use the JAWS “Links List” command to access the Links List and select Lookups. This is one of the links contained in the Main Menu frame.

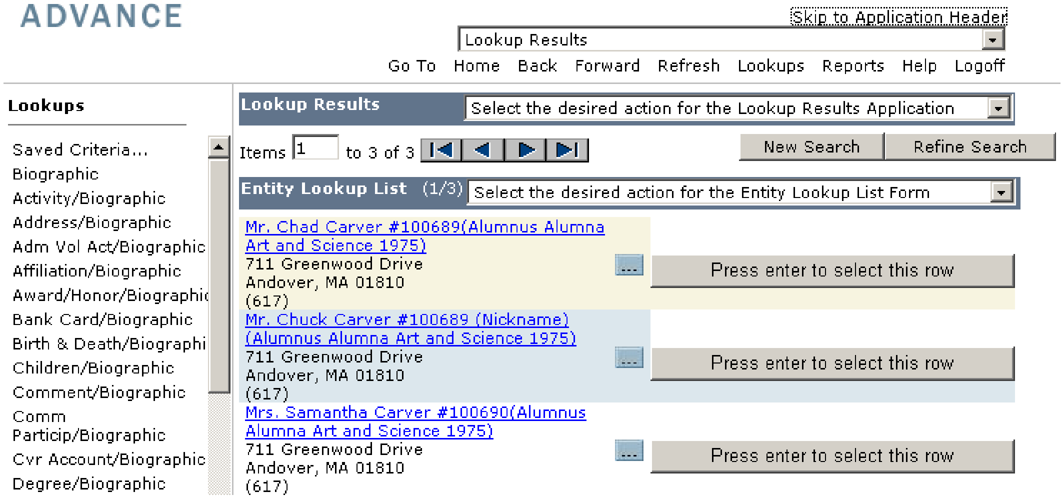
As this point you have two main options:

1. Use the JAWS “List Form” fields command to list all of the form fields on the application.
2. Use the JAWS “List Headings” command to select a particular area/form in which to be placed on the form. Then use the tab command to move through the data elements, starting within this form.

Once the desired criteria have been entered, the following steps should be performed to execute the search and navigate the search results:

1. Use the shortcut key Alt + S to activate the search button and initiate the search.
2. When the lookup result list returns, use the JAWS “Say All” command to hear the contents of the records in the lookup result list. The lookup result list forms are structured so that each element contains a label, followed by a corresponding data field value. Thus, they are read in pairs, from left to right and then down to the next line.
3. Towards the end of the record, JAWS will reach the ellipsis button and read “Press Enter to select this record.” If the current record is not the desired record, continue to use Say All until you hear the information for the desired record.
4. While on the desired record, stop ‘Say All’ mode and press tab to move to the button that says “Press enter to display details”. Performing this action will position the cursor on the selection button associated with that record. Press enter while on this button to select this record.

5. If the desired record is bypassed, use JAWS table commands to reposition the cursor on the correct record. Please note that a record may span multiple rows.



Items 1 to 3 of 3

Mr. Chad Carver #100689(Alumnus Alumna Art and Science 1975)
711 Greenwood Drive
Andover, MA 01810
(617)

Mr. Chuck Carver #100689 (Nickname) (Alumnus Alumna Art and Science 1975)
711 Greenwood Drive
Andover, MA 01810
(617)

Mrs. Samantha Carver #100690(Alumnus Alumna Art and Science 1975)
711 Greenwood Drive
Andover, MA 01810
(617)

Figure 63: Entity Lookup List

Sorting Columns

Some of the data tables in Advance Web allow you to sort the table by selecting the table's column titles that display as links. For example, on the Home Page, the column titles in the Tasks table (Scheduled, Name, Tasks, Status) are links and allow the table to be sorted by selecting the appropriate column link.

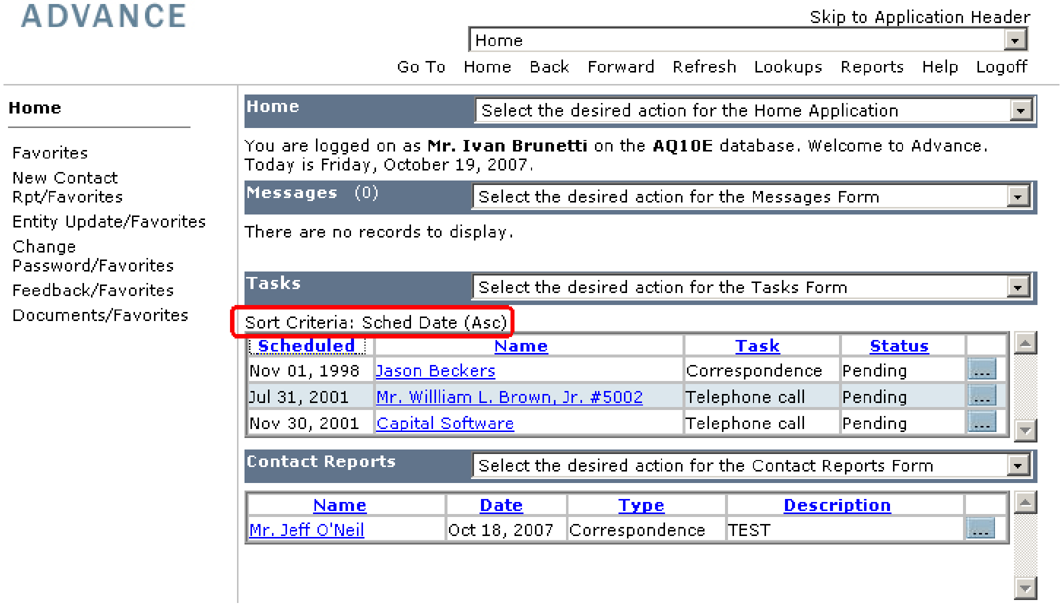
If you perform JAWS table commands to move into one of these title cells, you would hear the tool tip associated with this link. The text of the tool tip actually changes dynamically as these sort links can be toggled through three sort options.

- The first sort state is to sort the table by this column in ascending order. Therefore the tool tip text heard would include the column title and then the words "Press Enter to Sort Ascending."
- If you press this link and then move back into this cell, you would then hear that the tool tip text has changed to "Press Enter to sort Descending."
- If you again press this link to execute the sort and then move back into this cell, you would then hear the tool text changed to "Press Enter for default sort."

Forms that do not read "Press Enter to sort ascending," when first accessed, cannot be sorted using this mechanism.

If you press Enter to perform a sort on a particular column heading, JAWS focus is no longer on the table after the sort is performed. Thus, you must use JAWS Table commands to re-position yourself back in the table, in the desired cell.

As you navigate through the columns in the table, keep in mind the sort functionality is only available when positioned in the title cell.


ADVANCE
Skip to Application Header
Home
Go To Home Back Forward Refresh Lookups Reports Help Logoff
Home
Favorites
New Contact Rpt/Favorites
Entity Update/Favorites
Change Password/Favorites
Feedback/Favorites
Documents/Favorites
Home
Select the desired action for the Home Application
Messages (0)
Select the desired action for the Messages Form
Tasks
Select the desired action for the Tasks Form
Contact Reports
Select the desired action for the Contact Reports Form

You are logged on as **Mr. Ivan Brunetti** on the **AQ10E** database. Welcome to Advance. Today is Friday, October 19, 2007.

There are no records to display.

Sort Criteria: Sched Date (Asc)

Scheduled	Name	Task	Status	
Nov 01, 1998	Jason Beckers	Correspondence	Pending	...
Jul 31, 2001	Mr. Willliam L. Brown, Jr. #5002	Telephone call	Pending	...
Nov 30, 2001	Capital Software	Telephone call	Pending	...

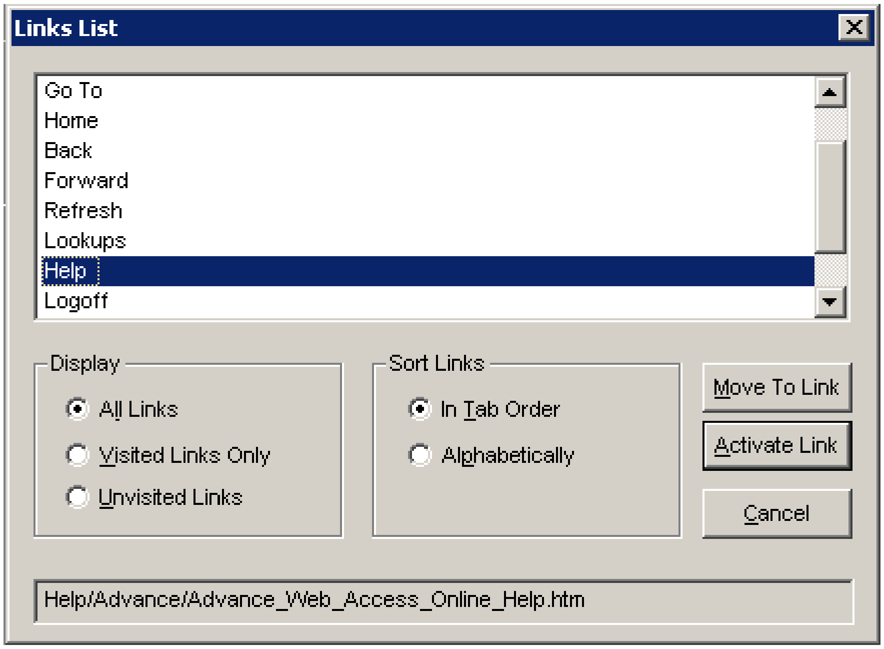
Name	Date	Type	Description	
Mr. Jeff O'Neil	Oct 18, 2007	Correspondence	TEST	...

Figure 64: Title cell

Online Help

Advance Web contains online help that can be accessed and read using JAWS. To access online help and listen to any topic:

1. Use the JAWS "Links List" command and select Help. When online help appears, JAWS recognizes help in a new browser window and announces the names and counts of several elements, including Frames, Links, Headings, etc.



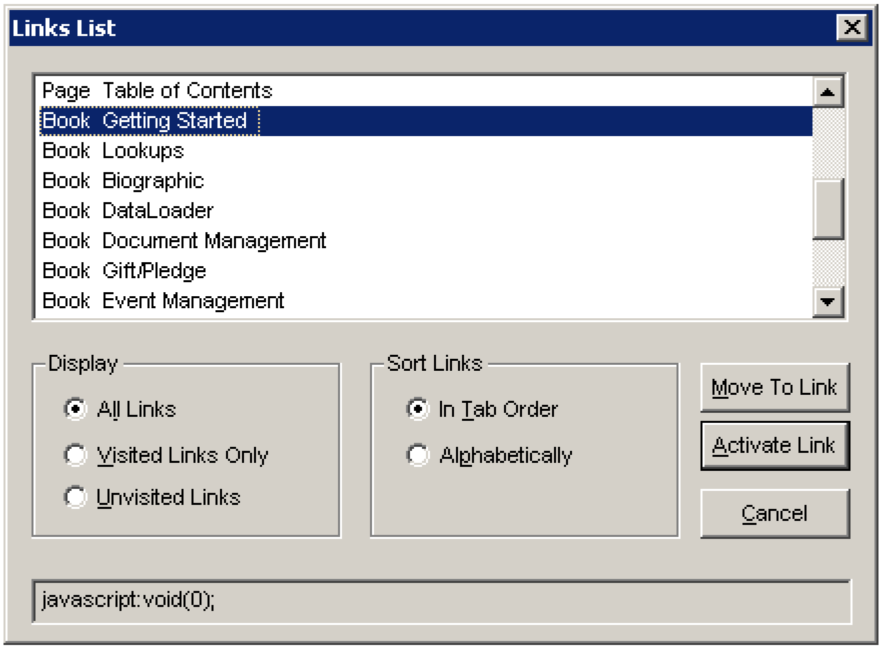
Display
● All Links
○ Visited Links Only
○ Unvisited Links

Sort Links
● In Tab Order
○ Alphabetically

Help/Advance/Advance_Web_Access_Online_Help.htm

Figure 65: Links List - Help

2. Use the JAWS “Links List” command to access the book entitled “Book Accessibility,” which can be found at the end of the list.



Page Table of Contents
Book Getting Started
Book Lookups
Book Biographic
Book DataLoader
Book Document Management
Book Gift/Pledge
Book Event Management

Display
● All Links
○ Visited Links Only
○ Unvisited Links

Sort Links
● In Tab Order
○ Alphabetically

javascript:void(0);

Figure 66: Links List - Book Accessibility

3. Scroll through the list of topics and use standard JAWS commands to select and read the appropriate topic. As you navigate through the Table of Contents and select each book, the links on the page and in the Links List will change accordingly.

Since online help opens in a new browser, you can toggle back to Advance Web using the Windows standard Alt + Tab key combination. Additionally, you can use the Advance Web Alt + F4 key combination to close the current window. When you close help, you are returned to the Advance Web form where you initiated help. Although this is not a new browser window, JAWS will announce you are entering a new browser window.

7

Grenzebach Glier and Associates DonorScape™ March 2008 Interface

Effective use of resources and cultivation efforts on individuals with the greatest potential to make annual, major, and/or planned gifts becomes increasingly more important for institutions and organizations. To address this, many Advance Web sites use the Grenzebach Glier and Associates (GG+A) DonorScape service to conduct predictive modeling and wealth screening in support of prospect identification and management. This service helps to identify the strongest potential supporters, based on a combination of their capacity to make a large gift and their likelihood to do so.

Functionality is included to address the transfer of data from Advance Web to DonorScape and from DonorScape back to Advance Web. This feature is included in the Advance Web Core license, but a DonorScape license for the version designated as March 2008 is required to use this functionality. The assumption is that this process will be run on a large scale for all or most entities in the database once every few years. However, this process may also be run on an ad hoc basis for newly added entities, such as new graduates or new parents, or for any other group of records desired by your institution or organization.

Procedure in a Nutshell

The procedure for accomplishing this transfer of data starts with the generation of a file containing a group of records found either by the existing Lookup functionality in Advance Web, or by some other method developed at your institution or organization, and captured in a Saved ID List. This file is then analyzed by GG+A. Subsequently, selected data generated by the DonorScape process is imported back into Advance Web via DataLoader, which has functionality to enable you to set certain specifications, such as mapping of certain fields. The data imported from DonorScape into Advance Web includes prospect giving capacity and affinity ratings, demographic/lifestyle ratings, other philanthropic activity, and wealth and asset information.

Initial Setup

Prior to running the report that creates the file for GG+A DonorScape, the output directory must be defined and the data must be mapped.

Setup for the data file returned from GG+A DonorScape involves defining the mapping of the data from DonorScape and creating DataLoader batch types, which are described in the section titled ‘Loading the DonorScape Data into Advance’, found later in this document.

Creating the Directory Object

Using your Oracle database tool of choice, define the directory object. The name of the directory object must be ‘DONORSCAPE1’. The directory path chosen must be accessible to the Web Server, as well as the Advance Web user(s) uploading the DonorScape file to GG+A.

The following permissions should be granted to the directory object:

```
GRANT READ, WRITE ON DIRECTORY donorscape1 TO advrole;
```

DonorScape Input File Mapping

Metadata identifies the sources of data to be extracted from Advance into the file for delivery to GG+A for processing by DonorScape. This mapping is predefined by SunGard Higher Education based on specifications provided by GG+A, but does provide flexibility. Although no interface is provided for maintaining the mapping data, the metadata may be modified by using your Oracle database tool of choice.

The following information is mapped:

Description	GG+A Preferred Header
Unique Prospect ID	ProspectId
Title (e.g. Mr., Mrs., Dr.)	Title
First Name	FirstName
Middle Name / Initial	MiddleName
Last Name	LastName
Suffix (e.g. Jr., Sr., III)	Suffix
Date of Birth (MM/DD/YYYY)	DateOfBirth
Age	Age
Gender	Gender
Marital Status	MaritalStatus
Spouse ID	SpouseId
Spouse First Name	SpouseFirst
Spouse Middle Name / Initial	SpouseMiddle
Spouse Last Name	SpouseLast
Home Address Line 1 (Street Name & Number)	HomeStreetAddress
Home Address Line 2 (Apartment, Unit, or Suite)	HomeAptUnitSuite
Home Address Line 3	HomeLine3
Home City	HomeCity
Home State	HomeState
Home Zip Code (5 digit)	HomeZipCode
Home Telephone Number	HomePhone
Employer/Business Name	Employer
Job Title	JobTitle
Business Address Line 1 (Street Name & Number)	BusinessStreetAddress
Business Address Line 2 (Apartment, Unit, or Suite)	BusinessAptUnitSuite
Business Address Line 3	BusinessLine3
Business City	BusinessCity
Business State	BusinessState
Business Zip Code (5 digit)	BusinessZipCode
Business Telephone Number	BusinessPhone
Seasonal Address Line 1 (Street Name & Number)	SeasonalStreetAddress
Seasonal Address Line 2 (Apartment, Unit, or Suite)	SeasonalAptUnitSuite
Seasonal Address Line 3	SeasonalLine3
Seasonal City	SeasonalCity
Seasonal State	SeasonalState
Seasonal Zip Code (5 digit)	SeasonalZipCode
Primary Relationship Code	Relationship

Description	GG+A Preferred Header
Constituent Type Codes	ConstituentCode1
Constituent Type Codes	ConstituentCode2
Constituent Type Codes	ConstituentCode3
Preferred Class Year	PreferredClassYear
Preferred School	PreferredSchool
Preferred Department or Degree	PreferredDepartment
Name of Assigned Development Officer	AssignedSolicitor
Additional Class Year (repeats 5 times)	None
Additional School (repeats 5 times)	None
Additional Department (repeats 5 times)	None
Volunteer Activity / Board Membership	None
Planned Gift Indicator (bequest intention)	None
Survey Participant	None
Count of Events Attended	None
Currently Under Active Management	None
Existing Wealth / Capacity / Affinity Rating	None
Other relevant/useful information (see note above)	None
Client-Defined Field #20	None
Cumulative Total Giving	TotalGivingDollars
Number of Total Gifts	TotalNumberofGifts
Largest Fiscal Gift Amount	LargestGiftAmount
Largest Single Gift Date (MM/DD/YYYY)	LargestGiftDate
Most Recent Fiscal Gift Amount	MostRecentGiftAmount
Most Recent Gift Date (MM/DD/YYYY)	MostRecentGiftDate
Total Gift Dollars-Current FY	None
Total Gift Dollars-Current FY minus 1	None
Total Gift Dollars minus Current FY minus 2	None
Total Gift Dollars minus Current FY minus 3	None
Total Gift Dollars minus Current FY minus 4	None
Total Gift Dollars minus Current FY minus 5	None
Total Gift Donors minus Current FY minus 6+	None
First Gift Amount	None
First Gift Date (MM/DD/YYYY)	None

A more detailed table showing the DonorScape Input File layout and the mapping delivered with this feature is in Appendix C.

Populating the DonorScape Options Table

The Advance donorscape_options table should be populated with the appropriate metadata to ensure accurate calculations in the report. This table is delivered with defaults that can be replaced with your own data or procedures, as shown below:

Option	Default Value
prospect_types†	I
volunteer_board_counts†	123456
activity_participation_codes†	P
event_participation_codes†	P
activity_part_fee_amt†, ***	0
event_part_fee_amt†, ***	0
active_mgmt_stage_codes†	
total_giving_proc*	adv_donorscape.total_giving
largest_fiscal_gift_proc*	adv_donorscape.largest_fiscal_gift
most_recent_fisc_gift_proc*	adv_donorscape.most_recent_fiscal_gift
tot_gift_amt_fy_proc*	adv_donorscape.tot_gift_amt_fiscal_year
first_gift_proc*	adv_donorscape.first_gift
preferred_degree_proc*	adv_donorscape.preferred_degree
addl_degree_proc*	adv_donorscape.addl_degree
client_defined_1_19_proc*	adv_donorscape.client_1_19
client_defined_20_value**	There is no default for this column

† Multiple prospect_types values and '_codes' values must be separated by a comma; volunteer_board_counts values should not be separated by a comma.

*If you want to change the '_proc' procedures, you may write your own and replace the name in the table with the name of the new procedure. Your procedure must be written without parameters and the net result is to update the donorscape_temp table so the data can be extracted.

**The client_defined_20_value is a unique identifier for the current batch; must be modified each time a batch is run.

***This is the minimum threshold used to determine if an event participation/activity should be included.

Generating the DonorScape Interface File

A report, titled "Create DonorScape Interface File," is available via Reports. This report is run from a saved Entity-based ID List, which will enable an authorized end-user to generate a text file to be used for processing by DonorScape. This report will neither require nor allow any input by the user, other than a prompt to select the desired ID List. All variables for the report are established 'behind the scenes', using SQL and metadata predefined by an authorized programmer/system administrator.

If you are generating a large volume of report results, it is possible that your system will timeout before the report is finished. If you encounter this situation, you should consider dividing your data across smaller ID Lists. You may also consider determining the list_id and running the report from the backend. The best way to determine the list_id is to use your SQL tool of choice to query the list_headers table for the description of the list you want to run. The following is an example of how to run the report procedure for list_id = 32:

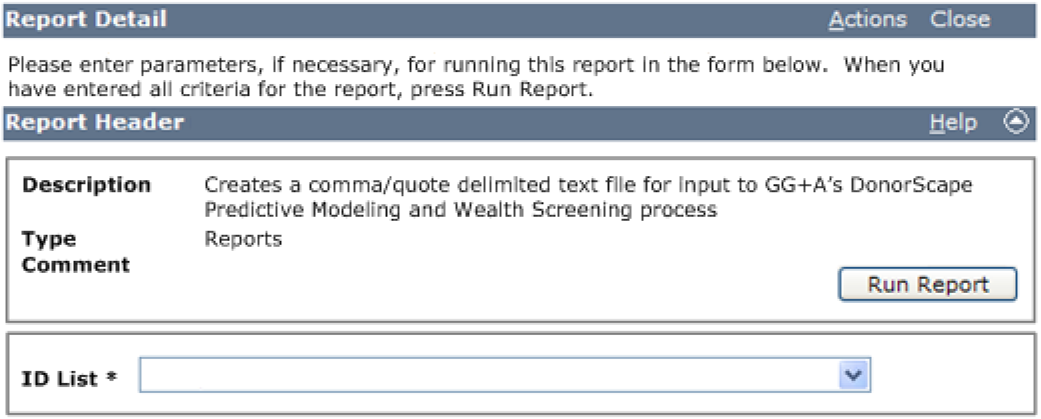
```
DECLARE o_rc   refcur_pkg.rc;
BEGIN
   adv_donorscape.donorscape_report(32, o_rc);
END;
/
```

To run this report from within Advance Web:

1. Generate a group of records via either Advance Web Lookup functionality, or by some other method developed at your institution or organization and Save the results as a Saved ID List.

   It is important to test your mapping with small Saved ID lists prior to running this report to confirm you have the appropriate data.

2. Click ‘Reports’ on the Main Menu and select ‘Create DonorScape Interface File’ from the list of available reports.



Please enter parameters, if necessary, for running this report in the form below. When you have entered all criteria for the report, press Run Report.

Description	Creates a comma/quote delimited text file for input to GG+A's DonorScape Predictive Modeling and Wealth Screening process
Type	Reports
Comment	
ID List *	

Figure 67: Donorscape Report Detail

3. Select your Saved ID List from the ID List drop-down.

   Only Entity-based ID lists will appear in the drop-down. Standard ID List security applies, i.e., the list must be Public and/or you must belong to a Rights Group which is granted access to the list in order to select it for the report. If you are not authorized to view the list, it will not appear in the drop-down.

4. Click Run Report to generate the GG+A DonorScape file. When the report is run, the following message will display:

“The process has completed successfully. The results file *(name of file)* has been saved.”

The name of the file is ‘dscape<timestamp>.csv’, where <timestamp> is in the form yyyymmddhhmiss.

ⓘ *The report results will not display on the screen within Advance Web, but are written to a file in a pre-defined location.*

🛠 *Since the name of the directory object (the pre-defined location of the output file) is hard-coded in the procedure used to run this report, it is necessary as part of implementation to ensure that the Web Server and the user who will upload the file to the GG+A FTP site have access to the directory.*

5. If you wish to view the .CSV output file, you may do so by navigating to the directory where it was saved by Advance Web and open it in the appropriate application.
6. Upload the file to the GG+A site.

DonorScape Output File

DonorScape provides several output data sets that could potentially be imported into various Advance tables. The data set identified as ‘Constituent List Export’ contains the most critical data elements for import into Advance and is the set addressed here.

This export file is obtained by you from GG+A’s website via http transfer to a location of your choice. You will then be required to transfer this data into a table of the same format, the donorscape_import table. This table is used as the source data for the “Create Batch from Tables” process. A layout of this export is shown in the table below:

ⓘ *The fields marked with an asterisk (*) are the relevant fields for import into Advance.*

DonorScape Constituent Export List Output Fields
Prospectid *
firstname
middleName
lastname
suffix
constituentCode1
constituentCode2
constituentCode3
ggaMajorGiftCode *
ggaAnnualGivingCode *
ggaPlannedGivingCode *
prizmClusterCode *
ggaSpecialFocusIndicator *
pcrCode *
spouseFirst
spouseMiddle
spouseLast
homestreetaddress
homeaptunitsuite
homecity
homestate
homezipcode

DonorScape Constituent Export List Output Fields
homemsa
homePhone
dateOfBirth
age
estimatedHouseholdIncome *
homeMarketValue *
relationship
assignedSolicitor
Employer
jobTitle
businessstreetaddress
businessaptunitsuite
businesscity
businessstate
businesszipcode
businessmsa
businessPhone
seasonalstreetaddress
seasonalaptunitsuite
seasonalcity
seasonalstate
seasonalzipcode
seasonalmsa
preferredSchool
preferredClassYear
preferredDepartment
campusSite
graduationDate
membershipType
totalGivingDollars
totalNumberOfGifts
largestGiftAmount
largestGiftDate
mostRecentGiftAmount
mostRecentGiftDate
clientDefinedField_1
clientDefinedField_2
clientDefinedField_3
clientDefinedField_4
clientDefinedField_5
clientDefinedField_6

DonorScape Constituent Export List Output Fields
clientDefinedField_7
clientDefinedField_8
clientDefinedField_9
clientDefinedField_10
clientDefinedField_11
clientDefinedField_12
clientDefinedField_13
clientDefinedField_14
clientDefinedField_15
clientDefinedField_16
clientDefinedField_17
clientDefinedField_18
clientDefinedField_19
clientDefinedField_20
AllFECTotalContributions *
AllGiftCapacityEstimate *
AllGiftCapacityRating *
AllIncome *
AllPension *
AllRealEstate *
AllSECDirHldgs&Sales *
AllSECIndHldgs *
AllDnBSalesVolume *
Exact&Near&PossibleFECTotalContributions *
Exact&Near&PossibleGiftCapacityEstimate *
Exact&Near&PossibleGiftCapacityRating *
Exact&Near&PossibleIncome *
Exact&Near&PossiblePension *
Exact&Near&PossibleRealEstate *
Exact&Near&PossibleSECDirHldgs&Sales *
Exact&Near&PossibleSECIndHldgs *
Exact&Near&PossibleDnBSalesVolume *
Exact&NearFECTotalContributions *
Exact&NearGiftCapacityEstimate *
Exact&NearGiftCapacityRating *
Exact&NearIncome *
Exact&NearPension *
Exact&NearRealEstate *
Exact&NearSECDirHldgs&Sales *
Exact&NearSECIndHldgs *
Exact&NearDnBSalesVolume *

DonorScape Constituent Export List Output Fields
ExactFECTotalContributions *
ExactGiftCapacityEstimate *
ExactGiftCapacityRating *
ExactIncome *
ExactPension *
ExactRealEstate *
ExactSECDirHldgs&Sales *
ExactSECIndHldgs *
ExactDnBSalesVolume *

Loading the DonorScape Data into Advance

To accommodate the import of the data file provided by DonorScape via DataLoader, values in a source table can be mapped into Advance Web. In addition, default values can be assigned, cross-reference tables (XREF) can be used and a 'WHERE' clause can be included to test which rows should be formed in the batch from the source data.

👁 *It is important to test your setup with small batches prior entering all of the DonorScape data into your database to confirm you have the correct data mapping.*

👁 *Advance System Option 184 controls the maximum number of transactions in DataLoader batches. Depending on the setting of this system option and the size of the DonorScape Constituent List Export file, it is likely multiple DataLoader Batches will be created when importing DonorScape data. It is important to establish the optimum value for this setting prior to importing your DonorScape data. Once you confirm the mapping is correct, as noted above, the recommendation is to start with batches of 10,000 records and adjust according to your system capabilities.*

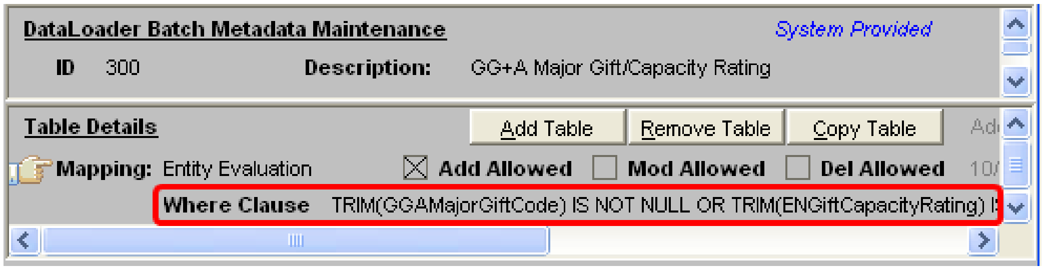
DataLoader Batch Metadata

Initial configuration involved for bringing the DonorScape output file data into Advance is to define the mapping of your data from the DonorScape Constituent List Export file into the appropriate areas in Advance. You need to create a Batch Metadata entry for each distinct set of data to be loaded. This is done via the Configuration Utility, as described on the following pages.

Batch Metadata Functionality

The Configuration Utility ‘DataLoader Batch Metadata’ window is used to define the mapping of data from the donorscape_import table to the desired table(s) in Advance. The ‘Table Details’ window defines the target Advance table into which the data will be loaded, and whether to allow Add, Modify, and/or Delete capability. The sample metadata delivered is Add only.

The ‘Where Clause’ column is used to enter the logic for importing data when creating batches from tables. For example, you may enter a ‘Where Clause’ to indicate that an import row should be created if a given column in the source table is not blank and not 0. If no ‘Where Clause’ is provided, all rows in the table will create a corresponding row in the batch.



DataLoader Batch Metadata Maintenance — System Provided

ID 300 Description: GG+A Major Gift/Capacity Rating

Table Details

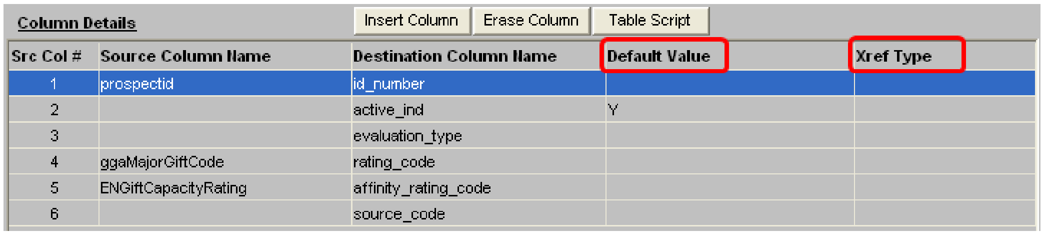
Mapping: Entity Evaluation ☒ Add Allowed □ Mod Allowed □ Del Allowed 10/

Where Clause TRIM(GGAMajorGiftCode) IS NOT NULL OR TRIM(ENGiftCapacityRating) I

Figure 68: 'Where Clause'

Column Details on the ‘DataLoader Batch Metadata’ window lists the desired destination columns to be populated in the target Advance table under ‘Destination Column Name’, along with an entry under ‘Source Column Name’ for a specific column in the Constituent List Export, and/or a Default Value.

The ‘Default Value’ column is used to define the desired default value to be imported into the destination column; when creating batches from tables, either a Default Value or a Source Column Name should be provided. If both are entered, the Default Value is used only if no data exists in the source column.



Column Details

Src Col #	Source Column Name	Destination Column Name	Default Value	Xref Type
1	prospectid	id_number		
2		active_ind	Y	
3		evaluation_type		
4	ggaMajorGiftCode	rating_code		
5	ENGiftCapacityRating	affinity_rating_code		
6		source_code		

Figure 69: Default Value and Xref Type

Another column, ‘XRef Type’, is used to reference the ldr_xref table, in which you may enter table translation values which are maintained outside of the Configuration Utility using your SQL tool of choice. Typically, there is no Default Value in rows which have a reference value in Xref Type. However, you may have a value in Default Value, as well as an Xref Type entry, if you wish to insert a Default Value for values that do not translate in the ldr_xref table.

ⓘ *When creating batches from tables, a value in source_column_nbr is required; it does not matter what numeric values are in source_column_nbr, as long as they are unique and there is a value.*

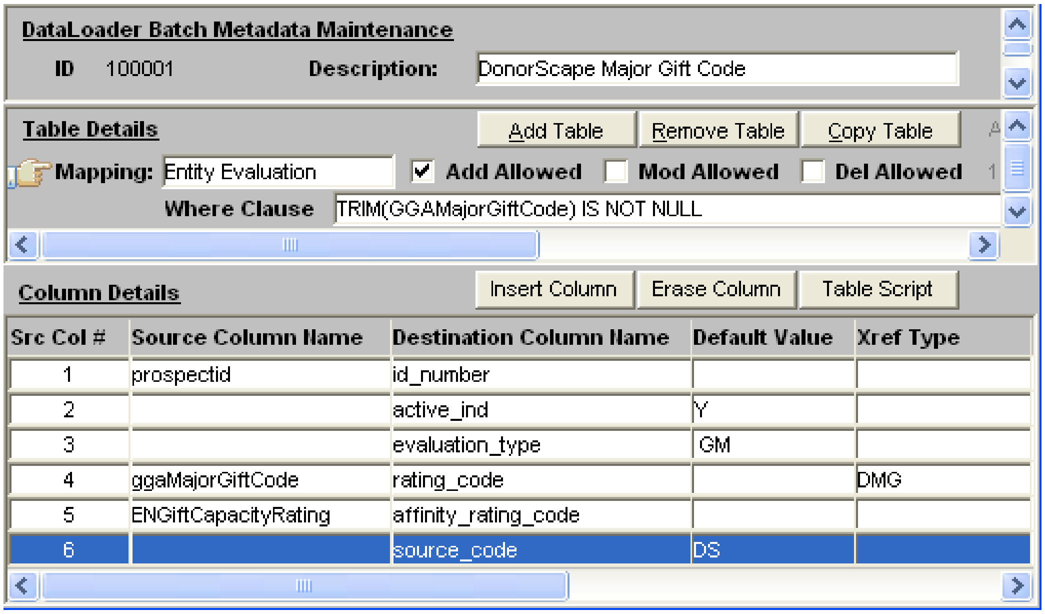
For example, let's say you wish to map the ggaMajorGiftCode into the rating_code column of the Advance evaluation table. The ggaMajorGiftCode value is a letter from A-E representing various levels of the prospect's affinity to make a gift. These letters have been translated to valid codes from tms_affinity_rating using the new table referenced above, with an XRef Type of 'DMG', as shown in the table below:

xref_type	source_value	advance_value
DMG	A	DA
DMG	B	DB
DMG	C	DC
DMG	D	DD
DMG	E	DE

In addition, you wish to load a Default Value of GM into the evaluation_type column and a Default Value of DS into the source_code column of the evaluation table. Also, the Default Value of 'Y' is entered for active_ind. There are no Source Column Names for these data elements, as they do not exist in the Constituent List Export.

You also want to map ENGiftCapacityRating to affinity_rating_code.

The setup for this scenario would be similar to what is shown below:



DataLoader Batch Metadata Maintenance

ID 100001 Description: DonorScape Major Gift Code

Table Details

Mapping: Entity Evaluation ☑ Add Allowed □ Mod Allowed □ Del Allowed

Where Clause TRIM(GGAMajorGiftCode) IS NOT NULL

Column Details

Src Col #	Source Column Name	Destination Column Name	Default Value	Xref Type
1	prospectid	id_number		
2		active_ind	Y	
3		evaluation_type	GM	
4	ggaMajorGiftCode	rating_code		DMG
5	ENGiftCapacityRating	affinity_rating_code		
6		source_code	DS	

Figure 70: Example mapping

If you wish to map another data element or set of data elements from the Constituent List Export into Advance, you may create another set of Batch Metadata, similar to the one shown above, to define the mapping of that data.

The table below shows the sample Batch Metadata delivered. The goal of delivering sample metadata is to provide you with examples as a starting point. Default Values and XRef Types are not provided, as they will need to be populated with your values, therefore this sample metadata will not be usable out-of-the-box; you need to clone and modify this metadata or create your own 'from scratch'.

Description	Mapping
DonorScape Annual Giving Code	Entity Evaluation
DonorScape EN FEC Total Contributions	Philanthropic Affinity
DonorScape EN Income	Wealth and Assets
DonorScape Major Gift/Capacity Rating	Entity Evaluation
DonorScape Prizm Cluster Code	Demographic Profile

Creating Batch Metadata

ⓘ *As part of this functionality, SunGard Higher Education delivers five rows of batch metadata for you to use for setting up your own, as mentioned earlier. This sample metadata is not usable as delivered, nor is it modifiable. You must clone and modify the samples, and subsequently modify the sample batch type delivered, or you must create your own.*

Using the descriptions on the previous pages as a reference guide, you can create your DataLoader batch metadata using the samples provided:

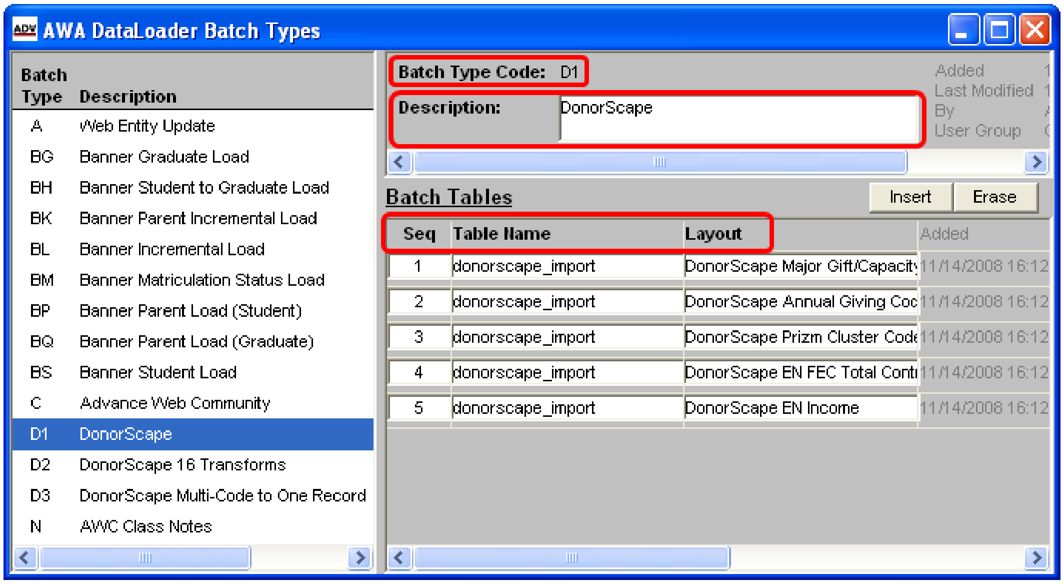
1. Log into the Configuration Utility and access the DataLoader Batch Metadata window via Web → Web Admin → DataLoader → Batch Metadata.
2. Choose the Batch Metadata type you would like to clone.
3. Click Clone to modify the selected type.
4. Enter a Description.
5. Either use the existing 'Where Clause' value or enter a new one.
6. Indicate whether to allow Add, Modify, and/or Delete of this type of data.
7. Confirm or change the following data:
    a. **Src Column #**: This must be a unique number for this metadata entry. **Required**
    b. **Source Column Name**: The name of the column from the donorscape_import table from which the data will be mapped. This is not required if a Default Value is provided.
    c. **Destination Column Name**: The name of the destination column in the Advance table. **Required**
    d. **Default Value**: A default value to be imported into the Advance destination column, if there is no corresponding source column data to be mapped from the donorscape_import table, or if the source column is blank. **Required if no Source Column Name is provided.**
    e. **XRef Type**: The xref_type from the ldr_xref table, which indicates a code translation, if necessary. Not required.
8. Repeat the previous step for each Advance destination column to be populated with either a source column value or a default value.
9. Click Save to save the mapping metadata.

10. Repeat steps 2-9 to create Batch Metadata for additional data to be loaded.
11. Close the DataLoader Batch Metadata window when you create all of the Batch Metadata necessary to load the desired data from the donorscape_import table into the desired locations in Advance.

Creating DataLoader Batch Types for Importing DonorScape Data

The next step in the process is to create one or more DataLoader Batch Types for loading your DonorScape data into Advance.

1. Log into the Configuration Utility and access the AWA DataLoader Batch Types window via Web → Web Admin → DataLoader → Batch Types.
2. Click New to add a new Batch Type.
3. Enter Batch Type Code and a Description.



AWA DataLoader Batch Types

Batch Type	Description
A	Web Entity Update
BG	Banner Graduate Load
BH	Banner Student to Graduate Load
BK	Banner Parent Incremental Load
BL	Banner Incremental Load
BM	Banner Matriculation Status Load
BP	Banner Parent Load (Student)
BQ	Banner Parent Load (Graduate)
BS	Banner Student Load
C	Advance Web Community
D1	DonorScape
D2	DonorScape 16 Transforms
D3	DonorScape Multi-Code to One Record
N	AWC Class Notes

Batch Type Code: D1

Description: DonorScape

Added 1
Last Modified 1
By A
User Group (

Batch Tables

Seq	Table Name	Layout	Added
1	donorscape_import	DonorScape Major Gift/Capacity	11/14/2008 16:12
2	donorscape_import	DonorScape Annual Giving Coc	11/14/2008 16:12
3	donorscape_import	DonorScape Prizm Cluster Code	11/14/2008 16:12
4	donorscape_import	DonorScape EN FEC Total Conti	11/14/2008 16:12
5	donorscape_import	DonorScape EN Income	11/14/2008 16:12

Figure 71: DataLoader Batch Types

4. Click Insert and enter the following data:
   a. **Seq**: The sequence in which the corresponding data will be assembled in the batch.
   b. **Table Name**: The source table name, which in this case is donorscape_import.
   c. **Layout**: The description of the DataLoader Batch Metadata.
5. Repeat the previous step for each set of Batch Metadata to be included in this Batch Type.
6. Click Save to save the Batch Type.

Importing Your DonorScape Data

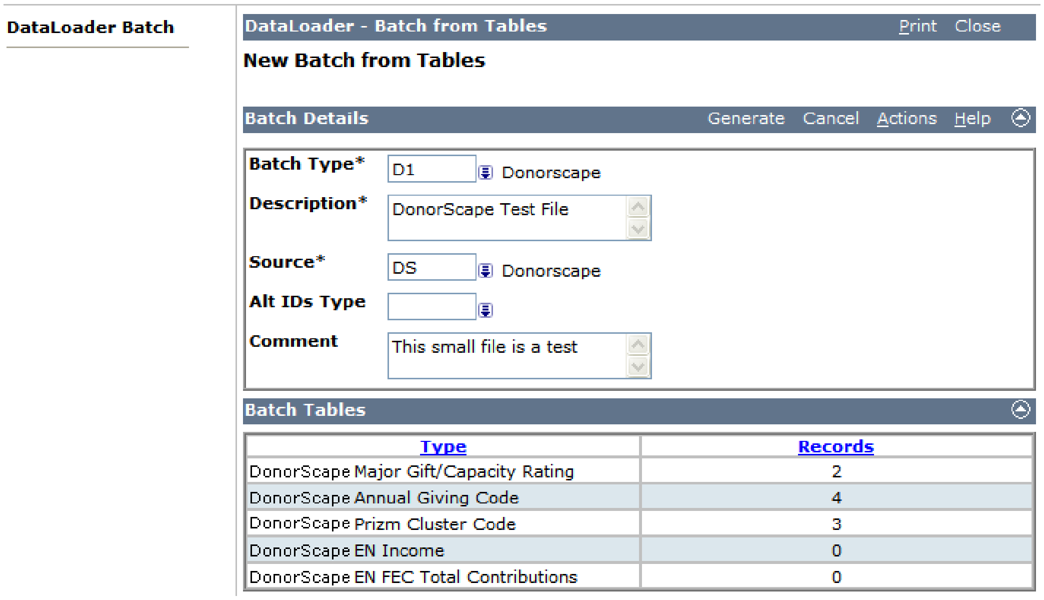
Once the system is set up to map the data, a system administrator populates the donorscape_import table with data received in the DonorScape Constituent List Export file. Once the table is populated, you can generate batches to import data into Advance.

If the volume of data to be loaded from DonorScape into Advance is very large, you may need to perform the batch creation, validation, and posting processes in groups. This may be handled by segmenting your DonorScape Constituent List Export File into smaller groups prior to running the batch creation process. Alternatively, or in addition to this step, you may wish to create multiple batch types to handle the various types of data to be loaded, rather than attempting to load all of your data using a single batch type. Some trial and error may be required to determine the appropriate approach depending on your file size, the amount of data to be loaded, and your technical environment.

To generate the batches with the DonorScape data:

1. In Go To, select DataLoader Batch from Tables (CBFT, Application ID 47) and click ‘Go’.
2. Select the Batch Type from the drop-down list, enter a Description and identify the Source as ‘DonorScape’ from the drop-down list. Enter additional information for the batch and click Generate. Note that the Batch Types display on the form with the corresponding number of records for each.

   For ‘DS – DonorScape’ to display as a choice on the Source drop-down list, it must be added to the tms_advloader_batch_source tms view.



New Batch from Tables

Batch Type* D1 Donorscape

Description* DonorScape Test File

Source* DS Donorscape

Alt IDs Type

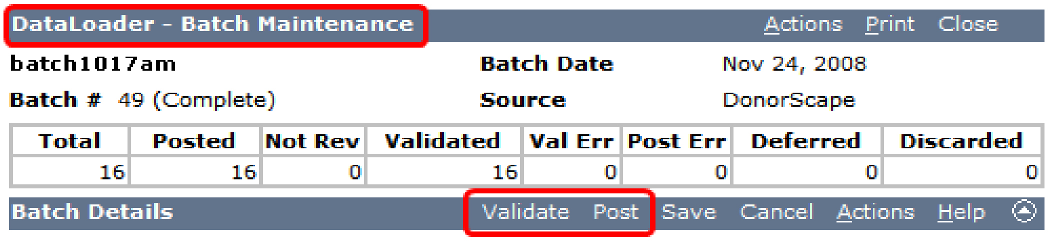
Comment This small file is a test

Type	Records
DonorScape Major Gift/Capacity Rating	2
DonorScape Annual Giving Code	4
DonorScape Prizm Cluster Code	3
DonorScape EN Income	0
DonorScape EN FEC Total Contributions	0

Figure 72: DataLoader - Batch from Tables

3. To complete importing the data into Advance, follow standard DataLoader procedures outlined in the *Advance Web DataLoader User's Guide*.

ⓘ *As a part of this functionality, if your Transaction List is not accessible because of the size of the batch, you can also Validate and Post batches from the Batch Maintenance application, as shown below:*



batch1017am **Batch Date** Nov 24, 2008

Batch # 49 (Complete) **Source** DonorScape

Total	Posted	Not Rev	Validated	Val Err	Post Err	Deferred	Discarded
16	16	0	16	0	0	0	0

Figure 73: DataLoader - Batch Maintenance

A Appendix: iModules Data Mapping

Mapping the data that is being transferred between Advance Web and iModules Encompass is key to the integrity of the Advance database. The following tables provide information relative to the information passed between the two products.

ⓘ *Fields identified as 'xsequence' are used to establish matching records on Advance and iModules where multiple records may exist, e.g., an entity may have multiple degrees and make a modification to one of those degrees using their iModules account. The xsequence value for each degree uniquely identifies each degree record. When the data pull occurs, the modification can be specifically identified by using its xsequence and the corresponding degree record in Advance can be updated.*

🛠 *There are two fields that identify what records are pushed to and pulled from iModules, push_this_field (push) and query_this_field (pull). A value of 'N' in push_this_field for a column indicates that you want to ignore the column and data will not be pushed. Likewise, a value of 'N' in query_this_field for a column indicates you do not want to pull the column from iModules. If a value of 'Y' is in these fields, you are indicating which columns of data are to be to pushed/pulled. These columns can be modified to 'Y' or 'N' to suit the needs of your institution or organization. To view or modify the values in these columns, please refer to the imodules_mapping table.*

Member Information Category **|Exporting data**

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
prefix	Title	Text Box			25	name	prefix
first_name	First Name	Text Box	Yes	Yes	25	name	first_name
middle_name	Middle Name	Text Box			25	name	middle_name
last_name	Last Name	Text Box	Yes	Yes	25	name	last_name
pers_suffix	Suffix	Text Box			25	name	pers_suffix
prof_suffix	Prof Suffix	Text Box			25	name	prof_suffix
gender_code	Gender	Drop Down List			1	entity	gender_code
birth_dt	Birth Date	Date Box			8	entity	birth_dt
salutation	Salutation	Text Box			40	name	salutation
ethnic_code	Ethnicity	Drop Down List			8	entity	ethnic_code
organization_name	Organization Name	Text Box			60	name	org_long_name1

Home Contact Information

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
addr_status	Address Status [hidden field]	Drop Down List			1	address	addr_status_ code
email_address	Primary E-mail	Email Text Box Confirm	Yes		320	email	email_address**
address_1	Address 1	Text Box	Yes		40	address	street1
address_2	Address 2	Text Box			40	address	street2
street3	Address 3	Text Box			40	address	street3
city	City	Text Box	Yes		30	address	city
state	State	Drop Down List	Yes		3	address	state_code
zipcode	Zip	Text Box	Yes		10	address	zipcode
country	Country	Drop Down List	Yes		5	address	country_code
foreign_cityzip	Foreign City	Text Box			40	address	foreign_cityzip
phone_number	Phone Number	Text Box	Yes		10	address	phone_area_ code
fax_number	Fax Number	Text Box			10	address	fax_area_code
county_code	COUNTY_CODE [hidden field]	Text Box			5	address	county_code

**By default, the email_address 'Required' value is set to 'Y' and these records are pushed to iModules. If you do not want email_address records pushed to iModules, set this value to 'N'.

Account Information Category (iModules fields)

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
is_lost	Is Lost [hidden field]	Hidden Field					

Education Information (your institution)

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
pref_school_code	Preferred School	Drop Down List			4	entity	pref_school_ code
pref_class_year	Preferred Class Year	Drop Down List	Yes	Yes	4	entity	pref_class_year
degree1_xseq	Degree 1 Import ID	Label			6	degrees	xsequence
degree1_code	1st Degree	Drop Down List			5	degrees	degree_code
degree1_year	1st Degree Year	Text Box			4	degrees	degree_year
degree1_institution	1st Degree Institution	Drop Down List			10	degrees	institution_code
degree1_school	1st Degree School	Drop Down List			4	degrees	school_code
degree1_campus	1st Degree Campus	Drop Down List			3	degrees	campus_code
degree1_major_code1	1st Degree Major 1	Drop Down List			6	degrees	major_code1
degree1_major_code2	1st Degree Major 2	Drop Down List			6	degrees	major_code2
degree1_major_code3	1st Degree Major 3	Drop Down List			6	degrees	major_code3
degree1_minor_code1	1st Degree Minor	Drop Down List			7	degrees	minor_code1
degree2_xseq	Degree 2 Import ID	Label			6	degrees	xsequence
degree2_code	2nd Degree	Drop Down List			5	degrees	degree_code
degree2_year	2nd Degree Year	Text Box			4	degrees	degree_year
degree2_institution	2nd Degree Institution	Drop Down List			10	degrees	institution_code
degree2_school	2nd Degree School	Drop Down List			4	degrees	school_code

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
degree2_campus	2nd Degree Campus	Drop Down List			3	degrees	campus_code
degree2_major_code1	2nd Degree Major 1	Drop Down List			6	degrees	major_code1
degree2_major_code2	2nd Degree Major 2	Drop Down List			6	degrees	major_code2
degree2_major_code3	2nd Degree Major 3	Drop Down List			6	degrees	major_code3
degree2_minor_code1	2nd Degree Minor	Drop Down List			7	degrees	minor_code1
degree3_xseq	Degree 3 Import ID	Label			6	degrees	xsequence
degree3_code	3rd Degree	Drop Down List			5	degrees	degree_code
degree3_year	3rd Degree Year	Text Box			4	degrees	degree_year
degree3_institution	3rd Degree Institution	Drop Down List			10	degrees	institution_code
degree3_school	3rd Degree School	Drop Down List			4	degrees	school_code
degree3_campus	3rd Degree Campus	Drop Down List			3	degrees	campus_code
degree3_major_code1	3rd Degree Major 1	Drop Down List			6	degrees	major_code1
degree3_major_code2	3rd Degree Major 2	Drop Down List			6	degrees	major_code2
degree3_major_code3	3rd Degree Major 3	Drop Down List			6	degrees	major_code3
degree3_minor_code1	3rd Degree Minor	Drop Down List			7	degrees	minor_code1
degree4_xseq	Degree 4 Import ID	Label			6	degrees	xsequence
degree4_code	4th Degree	Drop Down List			5	degrees	degree_code
degree4_year	4th Degree Year	Text Box			4	degrees	degree_year
degree4_institution	4th Degree Institution	Drop Down List			10	degrees	institution_code
degree4_school	4th Degree School	Drop Down List			4	degrees	school_code
degree4_campus	4th Degree Campus	Drop Down List			3	degrees	campus_code
degree4_major_code1	4th Degree Major 1	Drop Down List			6	degrees	major_code1
degree4_major_code2	4th Degree Major 2	Drop Down List			6	degrees	major_code2
degree4_major_code3	4th Degree Major 3	Drop Down List			6	degrees	major_code3
degree4_minor_code1	4th Degree Minor	Drop Down List			7	degrees	minor_code1

Other Education Information

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
other_degree1_xseq	Other Degree 1 Import ID	Label			6	degrees	xsequence
other_degree1_code	Other 1st Degree	Drop Down List			5	degrees	degree_code
other_degree1_year	Other 1st Degree Year	Text Box			4	degrees	degree_year
other_degree1_institution	Other 1st Degree Institution	Drop Down List			10	degrees	institution_code
other_degree1_ school	Other 1st Degree School	Drop Down List			4	degrees	school_code
other_degree1_campus	Other 1st Degree Campus	Drop Down List			3	degrees	campus_code
other_degree1_major_ code1	Other 1st Degree Major 1	Drop Down List			6	degrees	major_code1
other_degree1_major_ code2	Other 1st Degree Major 2	Drop Down List			6	degrees	major_code2
other_degree1_major_ code3	Other 1st Degree Major 3	Drop Down List			6	degrees	major_code3
other_degree1_minor_ code1	Other 1st Degree Minor	Drop Down List			7	degrees	minor_code1
other_degree2_xseq	Other Degree 2 Import ID	Label			6	degrees	xsequence
other_degree2_code	Other 2nd Degree	Drop Down List			5	degrees	degree_code

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
other_degree2_year	Other 2nd Degree Year	Text Box			4	degrees	degree_year
other_degree2_institution	Other 2nd Degree Institution	Drop Down List			10	degrees	institution_code
other_degree2_school	Other 2nd Degree School	Drop Down List			4	degrees	school_code
other_degree2_campus	Other 2nd Degree Campus	Drop Down List			3	degrees	campus_code
other_degree2_major_code1	Other 2nd Degree Major 1	Drop Down List			6	degrees	major_code1
other_degree2_major_code2	Other 2nd Degree Major 2	Drop Down List			6	degrees	major_code2
other_degree2_major_code3	Other 2nd Degree Major 3	Drop Down List			6	degrees	major_code3
other_degree2_minor_code1	Other 2nd Degree Minor	Drop Down List			7	degrees	minor_code1
other_degree3_xseq	Other Degree 3 Import ID	Label			6	degrees	xsequence
other_degree3_code	Other 3rd Degree	Drop Down List			5	degrees	degree_code
other_degree3_year	Other 3rd Degree Year	Text Box			4	degrees	degree_year
other_degree3_institution	Other 3rd Degree Institution	Drop Down List			10	degrees	institution_code
other_degree3_school	Other 3rd Degree School	Drop Down List			4	degrees	school_code
other_degree3_campus	Other 3rd Degree Campus	Drop Down List			3	degrees	campus_code
other_degree3_major_code1	Other 3rd Degree Major 1	Drop Down List			6	degrees	major_code1
other_degree3_major_code2	Other 3rd Degree Major 2	Drop Down List			6	degrees	major_code2
other_degree3_major_code3	Other 3rd Degree Major 3	Drop Down List			6	degrees	major_code3
other_degree3_minor_code1	Other 3rd Degree Minor	Drop Down List			7	degrees	minor_code1
other_degree4_xseq	Other Degree 4 Import ID	Label			6	degrees	xsequence
other_degree4_code	Other 4th Degree	Drop Down List			5	degrees	degree_code
other_degree4_year	Other 4th Degree Year	Text Box			4	degrees	degree_year
other_degree4_institution	Other 4th Degree Institution	Drop Down List			10	degrees	institution_code
other_degree4_school	Other 4th Degree School	Drop Down List			4	degrees	school_code
other_degree4_campus	Other 4th Degree Campus	Drop Down List			3	degrees	campus_code
other_degree4_major_code1	Other 4th Degree Major 1	Drop Down List			6	degrees	major_code1
other_degree4_major_code2	Other 4th Degree Major 2	Drop Down List			6	degrees	major_code2
other_degree4_major_code3	Other 4th Degree Major 3	Drop Down List			6	degrees	major_code3

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
other_degree4_minor_code1	Other 4th Degree Minor	Drop Down List			7	degrees	minor_code1

Spouse and Family Information

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
marital_status_code	Marital Status	Drop Down List			1	entity	spouse_status_code
jnt_salutation	Joint Salutation	Text Box			80	entity	jnt_salutation
marriage_dt	Date of Marriage	Date Box			10	entity	marriage_dt
children_nbr	Number of Children	Drop Down List			6		

Business Information

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
job_title	Job Title	Text Box			60	employment	job_title
employer	Employer	Text Box			60	employment	employer_name1
fld_of_work_code	Occupation	Drop Down List			4	employment	fld_of_work_code
employ_relat_code	Job Type	Drop Down List			2	employment	employ_relat_code

Business Contact Information

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
busaddr_status	Business Address Status [hidden field]	Drop Down List			1	address	addr_status_code
busstreet1	Business Address 1	Text Box			40	address	street1
busstreet2	Business Address 2	Text Box			40	address	street2
busstreet3	Business Address 3	Text Box			40	address	street3
buscity	Business City	Text Box			30	address	city
busstate	Business State	Drop Down List			3	address	state_code
buszip	Business Zip	Text Box			10	address	zipcode
buscountry	Business Country	Drop Down List			5	address	country_code
busforeign_cityzip	Business Foreign City	Text Box			40	address	foreign_cityzip
busemail	Business E-mail	Text Box			320	email	email_address
busphone_number	Business Phone Number	Text Box			10	address	phone_area_code
busfax_number	Business Fax Number	Text Box			10	address	fax_area_code

Other

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
board_current	Board Current	Drop Down List			3		
board_past	Board Past	Drop Down List			3		
board_all	Board All	Drop Down List			3		
boo_current	BOO Current	Drop Down List			3		
boo_past	BOO Past	Drop Down List			3		
boo_all	BOO All	Drop Down List			3		

Activities & Honors

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
interests_xseq1	Interests 1 Import ID	Label			6	interest	xsequence
interests_code1	Interests_code1	Drop Down List				interest	Interests_code
interests_xseq2	Interests 2 Import ID	Label			6	interest	xsequence
interests_code2	Interests_code2	Drop Down List				interest	Interests_code
interests_xseq3	Interests 3 Import ID	Label			6	interest	xsequence
interests_code3	Interests_code3	Drop Down List				interest	Interests_code
student_activity_xseq1	Student Activity 1 Import ID	Label			6	student_ activity	xsequence
student_activity_ code1	Student_activity_ code1	Drop Down List				student_ activity	Student_activity_ code
student_particip_ code1	Student_particip_ code1	Drop Down List				student_ activity	Student_particip_ code
student_activity_xseq2	Student Activity 2 Import ID	Label			6	student_ activity	xsequence
student_activity_ code2	Student_activity_ code2	Drop Down List				student_ activity	Student_activity_ code
student_particip_ code2	Student_particip_ code2	Drop Down List				student_ activity	Student_particip_ code
student_activity_xseq3	Student Activity 3 Import ID	Label			6	student_ activity	xsequence
student_activity_ code3	Student_activity_ code3	Drop Down List				student_ activity	Student_activity_ code
student_particip_ code3	Student_particip_ code3	Drop Down List				student_ activity	Student_particip_ code
activity_xseq1	Activity Code 1 Import ID	Label			6	activity	xsequence
activity_code1	Activity_code1	Drop Down List				activity	Activity_code
activity_participation _code1	activity_participation_ code1	Drop Down List				activity	activity_ participation_code
activity_xseq2	Activity Code 2 Import ID	Label			6	activity	xsequence
activity_code2	Activity_code2	Drop Down List				activity	Activity_code
activity_participation _code2	activity_participation_ code2	Drop Down List				activity	activity_ participation_code
activity_xseq3	Activity Code 3 Import ID	Label			6	activity	xsequence
activity_code3	Activity_code3	Drop Down List				activity	Activity_code
activity_participation _code3	activity_participation_ code3	Drop Down List				activity	activity_ participation_code

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
awd_honor_xseq1	Award Honor Import ID 1	Label			6	awards_ and_honors	xsequence
awd_honor_dt1	awd_honor_dt1	Drop Down List				awards_ and_honors	awd_honor_dt
awd_honor_code1	awd_honor_code1	Drop Down List				awards_ and_honors	awd_honor_code
awd_honor_xseq2	Award Honor Import ID 2	Label			6	awards_ and_honors	xsequence
awd_honor_dt2	awd_honor_dt2	Drop Down List				awards_ and_honors	awd_honor_dt
awd_honor_code2	awd_honor_code2	Drop Down List				awards_ and_honors	awd_honor_code
awd_honor_xseq3	Award Honor Import ID 3	Label			6	awards_ and_honors	xsequence
awd_honor_dt3	awd_honor_dt3	Drop Down List				awards_ and_honors	awd_honor_dt
awd_honor_code3	awd_honor_code3	Drop Down List				awards_ and_honors	awd_honor_code

Donation Information

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
current_fiscal_year_ gift	Current Fiscal Year Gift	Text Box					
current_fiscal_year_ mg	Current Fiscal Year Mg	Text Box					
current_fiscal_year_ mg_paid	Current Fiscal Year Mg Paid	Text Box					
current_fiscal_total_ recog	Current Fiscal Total Recognition	Text Box					
previous_fiscal_year_gift	Previous Fiscal Year Gift	Text Box					
previous_fiscal_year_ mg	Previous Fiscal Year Mg	Text Box					
previous_fiscal_year_ mg_paid	Previous Fiscal Year Mg Paid	Text Box					
previous_fiscal_total_ recog	Previous Fiscal Total Recognition	Text Box					
total_gifts_pp	Total Gifts pp	Text Box					
total_recog_credit	Total Recognition Credit	Text Box					
largest_trans	Largest Trans	Text Box					
largest_fy_total	Largest Fly Total	Text Box					
largest_pledge	Largest Pledge	Text Box					

Hidden Fields (Admin Only Category) * In addition there are several Hidden Fields that iModules Creates by default for the system.

iModules Database Column Name	Friendly Name	Type	Required	Default to Profile Page	Max Field Size	Advance Table	Field
id_number	Constituent ID	Text Box			10	entity	id_number
deceased	Deceased	Drop Down List			8	entity	death_dt
ruid	RUID	Text Box			9		
record_types	Record types	Text Box			99		
lost	Lost	Text Box			Yes/No		
	Member Controls						
is_deleted	Delete this member from the community:	Checkbox					
is_disabled	Disable membership:	Checkbox					
is_hidden	Hide this member from the directory:	Checkbox					
is_nonmember	This member is a Guest of the community (Non-member):	Checkbox					
	Communication Settings						
is_email_valid	E-mail address is valid:	Checkbox					
email_receive_comm	Wishes to receive e-mail from the community:	Checkbox					
email_receive_ member	Wishes to receive e-mail from other members:	Checkbox					
email_receive_notes	Wishes to receive Instant Notes:	Checkbox					
email_receive_updates	Wishes to receive Auto Updates:	Checkbox					
	iModules Member Info						
member_id	Member ID	Computed					
date_added	Date Added	Computed					
last_updated	Last Updated	Computed					

B

Appendix: Banner Data Mapping

Advance Table/Column	Banner Table	Banner Column	Banner Validation Table	Related Banner Forms	Comments
thirdpty_entity				**Student > General Person: SPAIDEN, SPAPERS**	
alternate_id_number	SPRIDEN	SPRIDEN_PIDM			This is the internal identification number of person.
mail_name					This is generated by Advance
salutation					This is generated by Advance
sort_name					This is generated by Advance
person_or_org_ind					Indicates person or non-person record. Always set to P
postnet_zip					This is generated by Advance
primary_type					This the record type from batch_app_type
status					Banner Integration option 16 (Default record status)
pref_class	SGBSTDN	SGBSTDN_ACYR_CODE			
pref_school	SGBSTDN	SGBSTDN_COLL_CODE_1	STVCOLL		
xlast	SPRIDEN	SPRIDEN_LAST_NAME			
xfirst	SPRIDEN	SPRIDEN_FIRST_NAME			
middle	SPRIDEN	SPRIDEN_MI			
prefix	SPBPERS	SPBPERS_NAME_PREFIX			
suffix	SPBPERS	SPBPERS_NAME_SUFFIX			
gender	SPBPERS	SPBPERS_SEX			
entity_source					This is set to Banner Integration option 4 (Data source code)
report_name					This is generated by Advance

Advance Table/Column	Banner Table	Banner Column	Banner Validation Table	Related Banner Forms	Comments
thirdpty_awards_and_honors				**Student >Academic History: SHADEGR - Honors tab**	
alternate_id_number	SHRDIGH and SHRDGDH	SHRDGIH_PIDM, SHRDGDH_PIDM			This is student internal identification number
award_date	SHRDGMR	SHRDGMR_GRAD_DATE			
xsequence					This is generated by Advance
award_honor_code		SHRDGIH_HONR_CODE, SHRDGDH_HOND_CODE	STVHOND, STVHONR		
data_source_code					This is set to Banner Integration option 4 (Data source code)
thirdpty_address				**Student - General Person: SOADDRQ; also, General Person: SPATELE**	
alternate_id_number	SPRADDR	SPRADDR_PIDM			This is the internal identification number of person.
person_or_org					This is always set to P
stop_date	SPRADDR	SPRADDR_TO_DATE			This is the effective end date of address associated.
xsequence					This is generated by Advance
type_code	SPRADDR	SPRADDR_ATYP_CODE	STVATYP		This is the Address type code
status_code	SPRADDR	SPRADDR_STATUS_IND			This is the address status - active or inactive
zipcode	SPRADDR	SPRADDR_ZIP			This is the zip code
postnet_zip					This is generated by Advance
preferred_indicator					This is set to 'Y' if it is a Home address, otherwise set to 'N'.
street_1	SPRADDR	SPRADDR_STREET_LINE1			First line of the address
street_2	SPRADDR	SPRADDR_STREET_LINE2			Second line of the address
street_3	SPRADDR	SPRADDR_STREET_LINE3			Third line of the address
city	SPRADDR	SPRADDR_CITY			City associated with the address
state	SPRADDR	SPRADDR_STAT_CODE	STVSTAT		State associated with the address
county	SPRADDR	SPRADDR_CNTY_CODE	STVCNTY		County associated with the address
country	SPRADDR	SPRADDR_NATN_CODE	STVNATN		Nation/Country associated with the address

Advance Table/Column	Banner Table	Banner Column	Banner Validation Table	Related Banner Forms	Comments
area_code	SPRTELE	SPRTELE_PHONE_AREA			This is the area code of the phone number linked with address
xnumber	SPRTELE	SPRTELE_PHONE_NUMBER			This is the phone number linked with address
exten	SPRTELE	SPRTELE_PHONE_EXT			This is the extension of the phone number linked with address
unlist_ind	SPRTELE	SPRTELE_UNLIST_IND			Unlisted telephone number indicator - only for phone numbers from SPRTELE
start_date	SPRADDR	SPRADDR_FROM_DATE			This is the effective end date of address
source_code					This is set to Banner Integration option 4 (Data source code)
phone_date_modified	SPRADDR	SPRADDR_ACTIVITY_DATE			
email_address	GOREMAL	GOREMAL_EMAIL_ADDRESS			Preferred email address
phone_status	SPRTELE	SPRTELE_STATUS_IND			Phone status
foreign_phone	SPRTELE	SPRTELE_INTL_ACCESS			Free format International access code for telephone number including country and city code
change_source_code					This is set to Banner Integration option 4 (Data source code)
pref_mail_name					This is generated by Advance
label1					This is generated by Advance
label2					This is generated by Advance
label3					This is generated by Advance
label4					This is generated by Advance
label5					This is generated by Advance
label6					This is generated by Advance
label7					This is generated by Advance
label8					This is generated by Advance
phone_src_code					This is set to Banner Integration option 4 (Data source code)
phone_pref_ind	SPRTELE	SPRTELE_PRIMARY_IND			This is set for phone numbers from SPRTELE

Advance Table/Column	Banner Table	Banner Column	Banner Validation Table	Related Banner Forms	Comments
thirdpty_degrees				**Inside degrees > Student - Academic History: SHADEGR; Outside degrees > Student - General Student - Educational Background: SOAPCOL**	
alternate_id_number		SHRDGMR_PIDM / SORDEGR_PIDM			This is the student internal identification number
stop_dt	STVTERM	STVTERM_END_DATE			
xsequence					This is generate by Advance
local_ind					Set to Y when the degree is a local degree. Set to N when the degree is non-local
institution_code	SORDEGR	SORDEGR_SBGI_CODE	STVSGBI		Non-Local degrees only (SORDEGR)
degree_level_code	SHRDGMR	SHRDGMR_LEVL_CODE			Degree Level Code.
school_code	SHRDGMR	SHRDGMR_COLL_CODE_1	STVCOLL		College offering the degree
campus_code	SHRDGMR	SHRDGMR_CAMP_CODE	STVCAMP		Campus offering the degree - local degrees only
degree_code	SHRDGMR	SHRDGMR_DEGC_CODE	STVDEGC		Students degree code (BS, BA, etc.).
degree_year	SHRDGMR	SHRDGMR_ACYR_CODE			Year in which the student graduates.
start_dt	STVTERM	STVTERM_START_DATE			
grad_date	SHRDGMR	SHRDGMR_GRAD_DATE			Graduation Date:SORDEGR_DEGC_DATE for outside degrees
major_1	SHRDGMR	SHRDGMR_MAJR_CODE_1	STVMAJR		First major associated with the degree - inside degrees only
major_2	SHRDGMR	SHRDGMR_MAJR_CODE_1_2	STVMAJR		Second major associated with the degree - inside degrees only
concentration_code	SHRDGMR	SHRDGMR_MAJR_CODE_CONC_1			Local degrees only (SHRDGMR)
honor_code_1	SORDEGR	SORDEGR_HONR_CODE			Non-Local degrees only (SORDEGR)
honor_code_2					Honors are linked with Degrees
xcomment	SHRDGCM	SHRDGCM_COMMENT			
minor_code1	SHRDGMR	SHRDGMR_MAJR_CODE_MINR_1	STVMAJR		First minor associated with the degree - local degrees only
minor_code2	SHRDGMR	SHRDGMR_MAJR_CODE_MINR_1_2	STVMAJR		Second minor associated with the degree - local degrees only
data_source_code					This is set to Banner Integration option 4 (Data source code)

Advance Table/Column	Banner Table	Banner Column	Banner Validation Table	Related Banner Forms	Comments
thirdpty_ids				**Student - General Person: SPAIDEN - Biographical tab / Alternate Identification tab**	
alternate_id_number	SPBPERS / SPRIDEN	SPBPERS_PIDM / SPRIDEN_PIDM			This is the student internal identification number
xsequence					This is generated by Advance
xtype					User designated type for SSN or Student ID
other_id		SPBPERS_SSN / SPRIDEN_ID			
thirdpty_sports				**Student - General Student: SGASPRT**	
alternate_id_number	SGRSPRT	SGRSPRT_PIDM			This is the student internal identification number
stop_date	STVTERM	STVTERM_END_DATE			
xsequence					This is generated by Advance
sport	SGRSPRT	SGRSPRT_ACTC_CODE	STVACTC		Activity Code
particip_code			STVSPST		Banner Integration Option 9 (Participation Code for Student Activities)
start_date	STVTERM	STVTERM_START_DATE			
xcomment	SGRCMNT	SGRCMNT_COMMENT_TEXT			
thirdpty_student_act				**Student - General Student: SGASTDN - Activities tab**	
alternate_id_number	SGRSACT	SGRSACT_PIDM			This is the student internal identification number
stop_date	STVTERM	STVTERM_END_DATE			
xsequence					This is generated by Advance
activity_code	SGRSACT	SGRSACT_ACTC_CODE	STVACTC		Activity type
activity_parti_code					Banner Integration option 10 (Participation code for student activities)
start_date	STVTERM	STVTERM_START_DATE			

Advance Table/Column	Banner Table	Banner Column	Banner Validation Table	Related Banner Forms	Comments
thirdpty_record_type				**Used for Matriculation and Student-Graduate Loads**	
alternate_id_number		SPRIDEN_PIDM			
record_type_code					Record type (Example AL for Graduates and ST for Student)
xsequence					This is generated by Advance
thirdpty_entity_birth_death				**Student > General Person: SPAPERS** **Student > Gen. Student: SGASTDN - Academic, Grad. Status tab;** **Guardian form: SOAFOLK**	
alternate_id_number	SPBPERS	SPBPERS_PIDM			Internal identification number of person.
ethnic_code	SPBPERS	SPBPERS_ETHN_CODE	STVETHN		The ethnicity code associated with person.
gender	SPBPERS	SPBPERS_SEX			gender code
religion	SPBPERS	SPBPERS_RELG_CODE	STVRELG		The religious affiliation associated with person.
birthdate	SPBPERS	SPBPERS_BIRTH_DATE			The person's birth date
death_date	SPBPERS	SPBPERS_DEAD_DATE			Person's deceased date
citizenship_1	GOBINTL	GOBINTL_NATN_CODE_LEGAL	STVNATN		Country of citizenship
obit_code					Banner system option 11 (Obit code for deceased individuals)
thirdpty_children				**Used when loading parents**	
child_relation_code					BI system option 12 (Parent child relationship code)
gender_code		GENDER_CODE			Gender Code from BNR_PARENT
child_id_number		ID_NUMBER			Student's entity ID Number
data_source_code					Banner Integration Option 4 (Data source code)
alternate_id_number		ORIG_ID_NUMBER			ID assigned to parent entities in Advance

C

Appendix: DonorScape Options

The following table defines the DonorScape options.

DonorScape Option	Description	Default
PROSPECT_TYPES	This is a comma separated list of Prospect Type Codes, used in the population of the following fields: • AssignedSolicitor • ExistingWealthRating • UnderActiveMgmtInd	I
VOLUNTEER_BOARD_COUNTS	This is a concatenated list of which counts to include for the VolunteerActivityCount field, e.g., 12356, where 1 = vol_activity 2 = committee 3 = affiliation 4 = mentor 5 = admissions_volunteer 6 = record_type CFAE type = 'B'	123456
ACTIVITY_PARTICIPATION_CODES	This is a comma-separated list of Activity Participation Codes, e.g., P, D for determining which, if any, Activities are counted for the EventsAttended field.	P
EVENT_PARTICIPATION_CODES	This is a comma-separated list of Event Participation Status Codes, e.g., P, D for determining which, if any, Events are counted for the EventsAttended field.	P

DonorScape Option	Description	Default
ACTIVITY_PART_FEE_AMT	This is a number greater than or equal to 0 which is compared against the Activity Fee for determining which, if any, Activities are counted for the EventsAttended field.	0
EVENT_PART_FEE_AMT	This is a number greater than or equal to 0 which is compared against the Event Fee for determining which, if any, Events are counted for the EventsAttended field.	0
ACTIVE_MGMT_STAGE_CODES	This is a comma-separated list of Prospect Stage codes used in the population of the UnderActiveMgmtInd field.	
TOTAL_GIVING_PROC	This column holds the name of a procedure that populates the TotalGivingDollars and TotalNumberofGifts fields.	adv_donorscape.total_giving
LARGEST_FISCAL_GIFT_PROC	This column holds the name of a procedure that populates the LargestFiscalAmount and LargestGiftDate fields.	adv_donorscape.largest_fiscal_gift
MOST_RECENT_FISC_GIFT_PROC	This column holds the name of a procedure that populates the MostRecentFiscalAmount and MostRecentGiftDate fields.	adv_donorscape.most_recent_fiscal_gift
TOT_GIFT_AMT_FY_PROC	This column holds the name of a procedure that populates the following fields: • TotalGiftDollarsCurrentFY • TotalGiftDollarsFY_1 • TotalGiftDollarsFY_2 • TotalGiftDollarsFY_3 • TotalGiftDollarsFY_4 • TotalGiftDollarsFY_5 • TotalGiftDollarsFY_Rest	adv_donorscape.tot_gift_amt_fiscal_year
FIRST_GIFT_PROC	This column holds the name of a procedure that populates the FirstGiftAmount and FirstGiftDate fields.	adv_donorscape.first_gift

DonorScape Option	Description	Default
PREFERRED_DEGREE_PROC	This column holds the name of a procedure that populates the PreferredDegree field.	adv_donorscape.preferred_degree
ADDL_DEGREE_PROC	This column holds the name of a procedure that populates the following fields: • AddlClassYear1-5 • AddlSchool1-5 • AddlDept1-5	adv_donorscape.addl_degree
CLIENT_DEFINED_1_19_PROC	This column holds the name of the procedure to populate the fields ClientDefined1-19. The default procedure delivered, adv_donorscape.client_1_19, will return a string of 19 commas. It will be up to the client to modify the procedure to return a string of up to 19 comma-separated values, as desired. There must be 19 commas even if fewer than 19 values are provided. Values should follow the commas. Strings and dates should be enclosed within two sets of two single quotes. Numeric data does not require quotes. There can be a mix of value types, for example: ',1,"A",3,"B","12/31/2008", …	adv_donorscape.client_1_19
CLIENT_DEFINED_20_VALUE	This column holds a value that populates the field ClientDefined20	

#####