Using the File Parser Process

Understanding the File Parser Process

The File Parser process enables you to convert data from an external file into PeopleSoft data and place it into tables in your Campus Solutions database. The external file can be a delimited file or a flat file. It can be a simple file with one row type or a complex file with several row types. If your institution routinely receives an external file from which you need to move data into your database, consider using the File Parser process to expedite the entry of that data. Whether the file is a small one that your institution receives daily or a large one that is received annually, the one-time setup of the File Parser process for that file can save significant data entry time.

Note: Suggested best practice: When creating File Parser definitions, it is recommended that the targeted tables be of an interim or staging classification, rather than have the File Parser process import data directly into application tables. Using interim or staging tables enable you to use existing data validation processes or create your own data validation processes. These validation processes can ensure that the data is clean before you load the data into one or more application specific tables.

To set up and use the File Parser process for an external file:

- Set up a field conversion definition that identifies a field values to convert from the external file to the target staging table.
- Set up context definitions that identify how to convert the data and in which staging tables to place the converted data to hold it for processing. Each context must also identify the parent-to-child relationship of the staging tables.
- Set up a file mapping definition that associates fields on the external file with the context definition to use.
- View the layout of each staging table, modify it, and map the fields from it to the target tables.
- Preview the hierarchical record layout and the converted data before you move the data into the staging tables.
- Run the File Parser process to convert the data and place it into the staging tables.

Setting Up Field Conversion Definitions

To set up field conversion definitions, use the Field Conversion (SCCFP_CNVR) and Copy Field Conversion Definition (SCCFP_CPY_FLD_CNVR) components.

This section discusses how to:

- Set up a field conversion definition.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

- Copy a field conversion definition.

Pages Used to Set Up Field Conversion Definitions

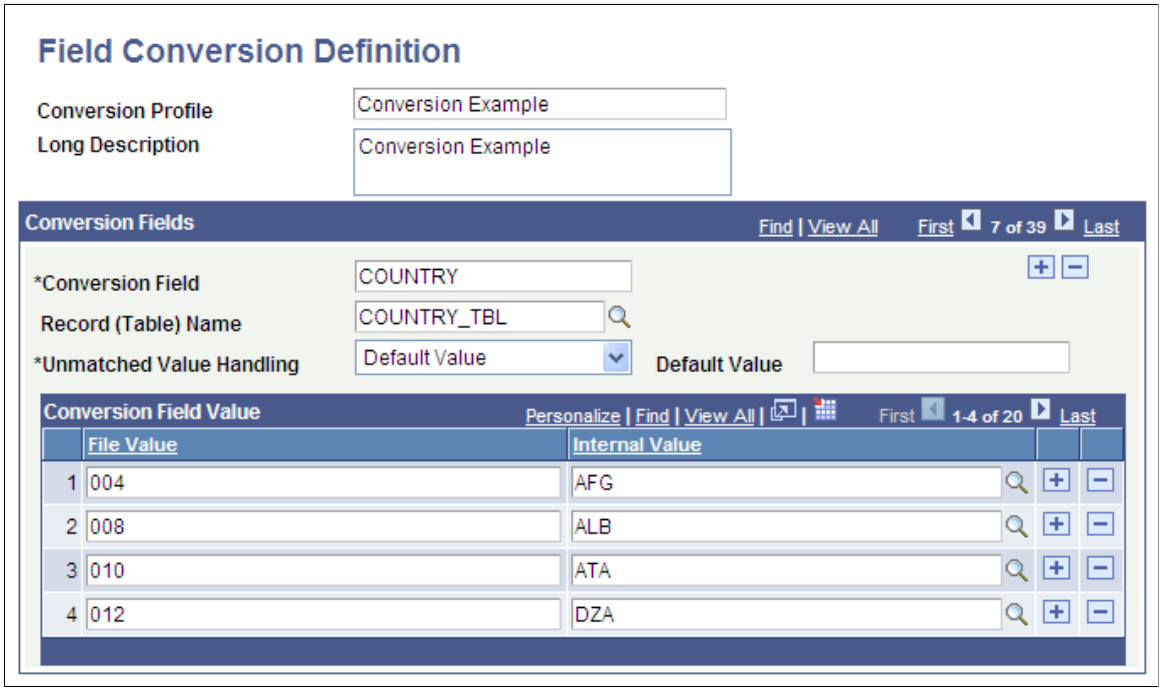
Page Name	*Definition Name*	*Navigation*	*Usage*
Field Conversion Definition	SCCFP_CNVR	Set Up SACR, System Administration, Utilities, File Parser, Field Conversion Definition	Identify field values to convert from an external file.
Copy Field Value Conversion	SCCFP_CPY_FLD_CNVR	Set Up SACR, System Administration, Utilities, File Parser, Copy Field Value Conversion	Copy an existing field conversion definition, save it with a new definition name, and change it as necessary to create another definition.

Setting Up a Field Conversion Definition

Access the Field Conversion Definition page (Set Up SACR, System Administration, Utilities, File Parser, Field Conversion Definition).

Image: Field Conversion Definition page

This example illustrates the fields and controls on the Field Conversion Definition page. You can find definitions for the fields and controls later on this page.



Field conversion definitions are optional. They are relevant only if the external file contains field values that are different from the PeopleSoft system values. Use the Field Conversion definition to define the mapping of external file values to the internal PeopleSoft system values. For example, if the external file uses the value *FALSE* but the internal value is *F,* you should create a mapping to convert the value when the file is loaded.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Conversion Profile

Enter a name for the conversion of this external file. Consider naming the profile the same name as the external file or a name that otherwise reflects the type of data to load.

(Optional) Enter a long description to further identify this conversion profile.

Conversion Fields

Conversion Field

Enter a name for converting each field.

Add as many conversion fields as necessary to identify each field to convert from the external file.

Record (Table) Name

Enter a record from which to prompt for target field names.

Note: This field can only be used for setup tables where the Internal Value is the only key.

Unmatched Value Handling

Indicate what the File Parser process should do if it cannot map the field value of the conversion field to the field in the target staging table field value. The choices are:

Default Value: Use the default value that you specify.

Error - Halt Processing: Cease loading data into the staging table and display the message that you specify.

Default Value

When you enter *Default Value* in the Unmatched Value Handling field, the Default Value field appears. You must specify the default value to use.

If you do not enter a default value, a blank or null value becomes the default value. If the value of the field cannot be mapped, the process will leave the value blank and continue mapping.

Message Set Number and **Message Number**

When you enter *Error - Halt Processing* in the Unmatched Value Handling field, the Message Set Number and Message Number fields appear. You must identify the error message to use.

Conversion Field Value

File Value

Enter each possible value for this field from the external file.

Internal Value

Enter the correlating PeopleSoft value. Select the prompt only if you have entered a value in the Record (Table) Name field. Otherwise, manually enter the internal PeopleSoft value.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Copying a Field Value Conversion Definition

Access the Copy Field Value Conversion page ((Set Up SACR, System Administration, Utilities, File Parser, Copy Field Value Conversion).

After you copy a field conversion definition and save it with a different name, you can modify it as necessary to create another definition.

On the search page, select the field conversion definition to copy, and then enter the name of the new conversion definition and save the page. To edit the new definition, access it from the Field Conversion Definition page.

Setting Up Context Definitions

This section discusses how to:

- Set up a context definition.
- View the record tree.
- Copy a context definition.

Pages Used to Set Up Context Definitions

Page Name	*Definition Name*	*Navigation*	*Usage*
Context Definition	SCCFP_CNTXT_DFN	Set Up SACR, System Administration, Utilities, File Parser, Context Definition	Identify the field names and formats to use for converted data and the staging tables in which to place the converted data and hold for processing.
Record Tree	SCCFP_TREE_RECPRVW	Set Up SACR, System Administration, Utilities, File Parser, Context Definition, Record Tree	View the record tree to determine the hierarchical layout of the staging tables.
Copy Context Definition	SCCFP_CPY_CNTXT	Set Up SACR, System Administration, Utilities, File Parser, Copy Context Definition	Copy an existing context definition with a new definition name.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Setting Up a Context Definition

Access the Context Definition page (Set Up SACR, System Administration, Utilities, File Parser, Context Definition).

Image: Context Definition page (1 of 2)

This example illustrates the fields and controls on the Context Definition page (1 of 2). You can find definitions for the fields and controls later on this page.

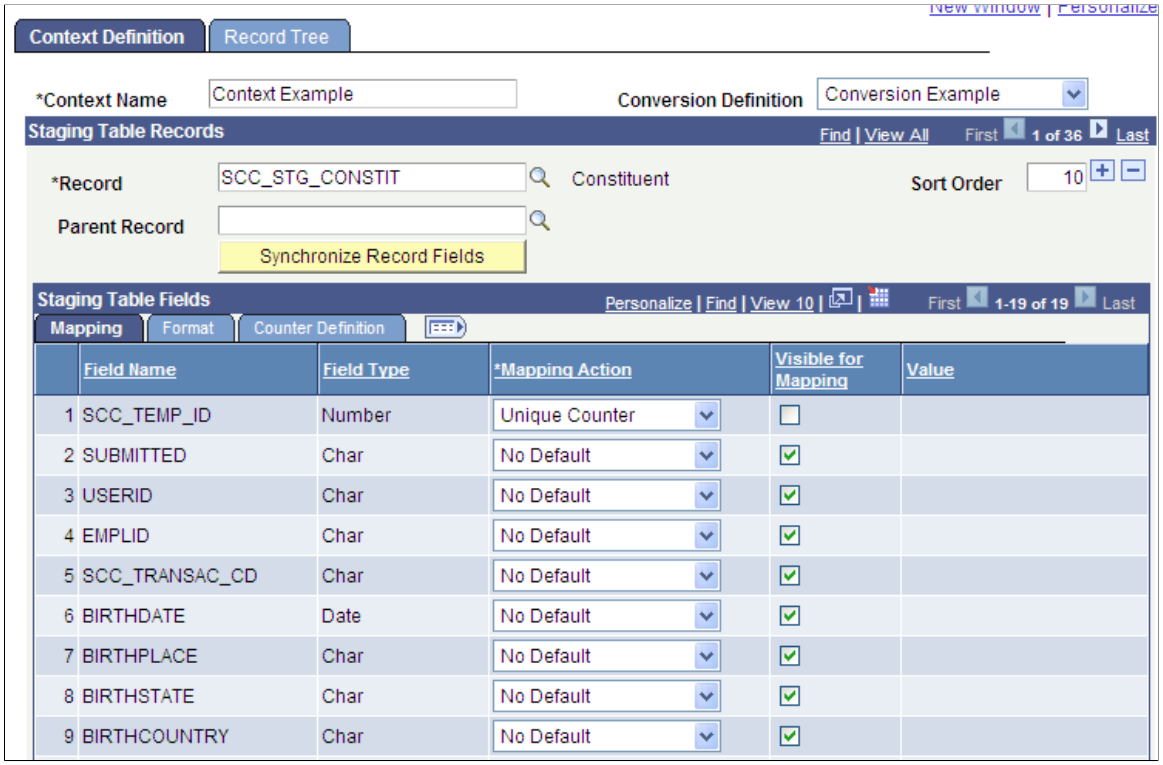


Image: Context Definition page (2 of 2)

This example illustrates the fields and controls on the Context Definition page (2 of 2). You can find definitions for the fields and controls later on this page.

Context Name Enter a name for this set of staging tables.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Conversion Profile

Enter the field conversion profile to associate with this context.

You can enter the profile here or on the File Mapping Definition page when you map the files. If you enter a profile here, it appears as the default profile on the File Definition page. You can change it on the File Mapping Definition page. The conversion profile entered on the File Mapping Definition page takes precedence.

Staging Table Records

Record

Enter the staging table to use. When you enter a table name, the system displays the fields from that table.

Add as many staging tables as needed for this conversion context.

Sort Order

Enter the order in which the table should appear in the hierarchy of staging files for mapping.

Parent Record

Enter the name of this staging table's parent record, if any.

Synchronize Record Fields

Click to compare fields in the staging table to fields in the target table.

If the fields in the tables are different, the system adds and deletes fields in the staging table until it contains only the fields that are in the target table. The changes are for the staging table as associated with this context definition only. The system displays a message indicating the changes made to the record.

Important! You must perform a thorough impact analysis when applying any software updates in order to identify delivered record definitions that are being referenced by current File Parser definitions.

Note: If changes are made to the record, it is required that you click this button in order to successfully run the File Parser process.

Staging Table Fields - Mapping Tab

Field Name

The system lists each field in the specified staging table.

When you synchronize the tables, the system updates the list to include only the synchronized fields.

Mapping Action

Enter the value to use for this field. Values are:

Default Value: Enter a hard-coded value set for this field, including any variables that are supported. Supported variables include *%Date, %DateTime, %OperatorID, %EmployeeID,*

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

%MapID (the current file parser map ID), *%MapName* (the current file parser map name), *%FileName,* and a number representing the file row number.

Inherit from Parent: Enter the value from the field of the same name on the parent record.

No Default: Enter no value. The field must be mapped from the File Mapping Definition component.

Process Instance: Use the current process instance.

Sequence: Enter a numerical sequence that is incremented by one for each record inserted.

Unique Counter: Use the next number from the supplied counter record or field pair.

Calculated: Define a user-written PeopleCode application class to calculate the field value. Examples of calculated mapping include concatenation of fields, unique sequencing, and deriving the field value from other staged values (for instance, establishing an External Org ID based on an incoming CEEB code, subject description, activity description, and so on). For more examples, see Creating Calculated Mapping Application Class Objects.

When you select *Calculated*, the Calculated Field Application Class tab appears. Specify the application class details on this tab.

Important! Do not base calculated field values on other calculated fields.

Visible for Mapping

Select this check box for the field to be available on pages in the File Mapping Definition component.

If a field is premapped and the user should not change the mapping, clear the check box to prevent it from appearing on pages in the File Mapping Definition component page.

Value

The corresponding hard-coded value appears when the Mapping Action field is set to *Default Value.*

Staging Table Fields - Format Tab

The Format tab is available only if *Date* or *Number* is selected for one or more field types.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access the Context Definition page - Format tab.

Image: Context Definition page - Format tab (1 of 2)

This example illustrates the fields and controls on the Context Definition page - Format tab (1 of 2). You can find definitions for the fields and controls later on this page.

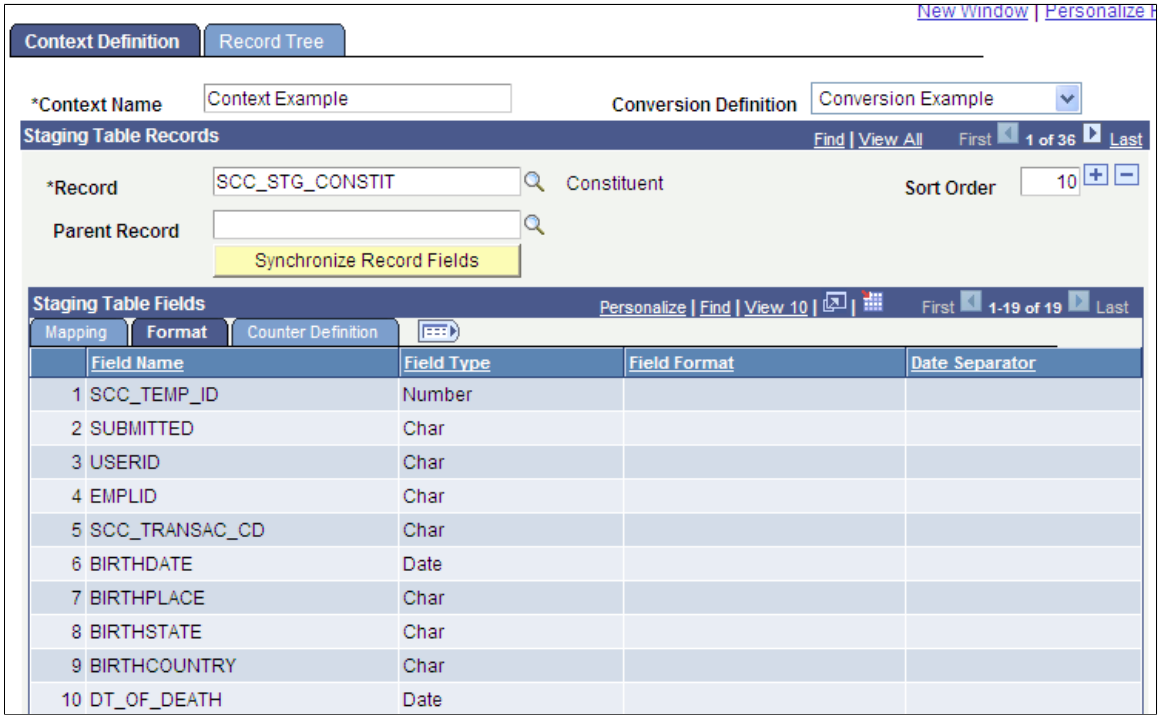
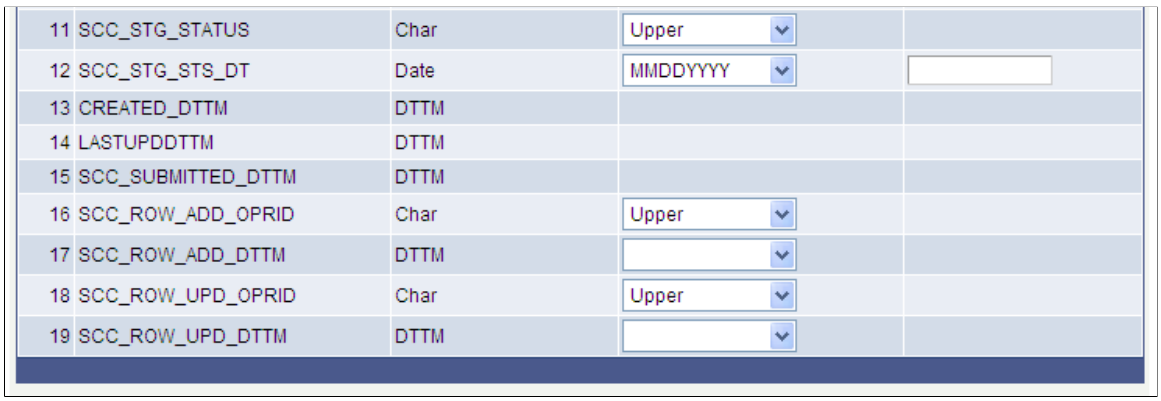


Image: Context Definition page - Format tab (2 of 2)

This example illustrates the fields and controls on the Context Definition page - Format tab (2 of 2). You can find definitions for the fields and controls later on this page.



Field Format This field appears if formatting options exist. Enter the format to use for the field value. Each field has options relevant to that data type. For example, a date type format can be DDMMMYY, DDMMMYYYY,DDMMYY, and so on. A character type format uses the case controls *Lower, Proper,* and *Upper.*

Date Separator This field appears only for date fields.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Enter the separator, if any, to use to separate day month and year in whatever date format you enter. For example, if you enter a format of *MMDDYYYY* and a separator of */,* the value might be 06/25/2007.

The system ignores leading zeros when a separator is used. If a separator is not used, leading zeros are required.

Staging Table Fields - Counter Definition Tab

The Counter Definition tab is available only if a Mapping Action of *Unique Counter* is selected for one or more field types.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access the Context Definition page - Counter Definition tab.

Image: Context Definition page - Counter Definition tab (1 of 2)

This example illustrates the fields and controls on the Context Definition page - Counter Definition tab (1 of 2). You can find definitions for the fields and controls later on this page.

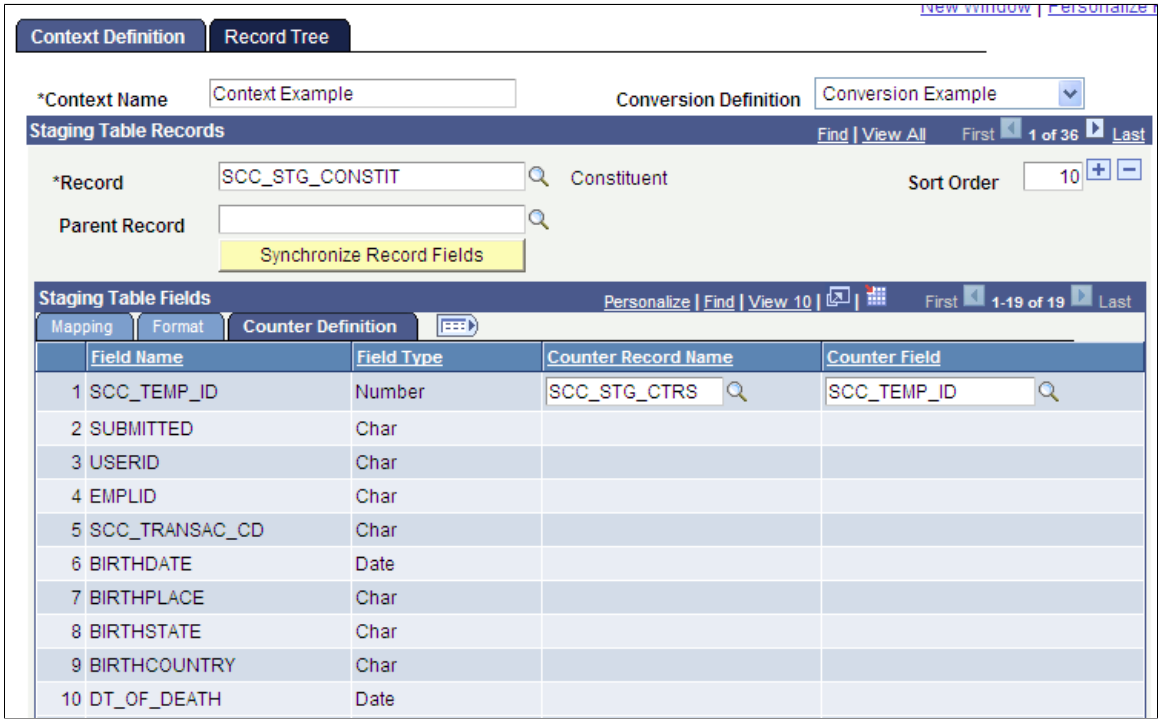
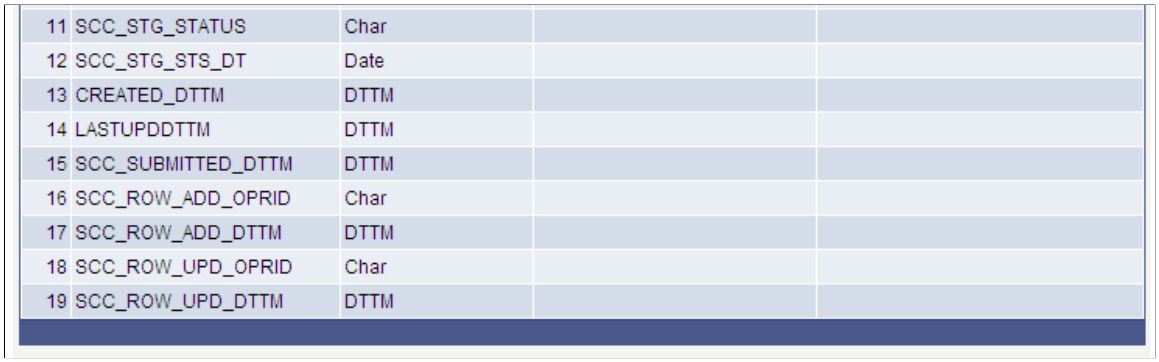


Image: Context Definition page - Counter Definition tab (2 of 2)

This example illustrates the fields and controls on the Context Definition page - Counter Definition tab (2 of 2). You can find definitions for the fields and controls later on this page.



Counter Record Name and **Counter Field** Enter the record (table) and field from which to take the next sequential number.

Staging Table Fields - Calculated Field Application Class Tab

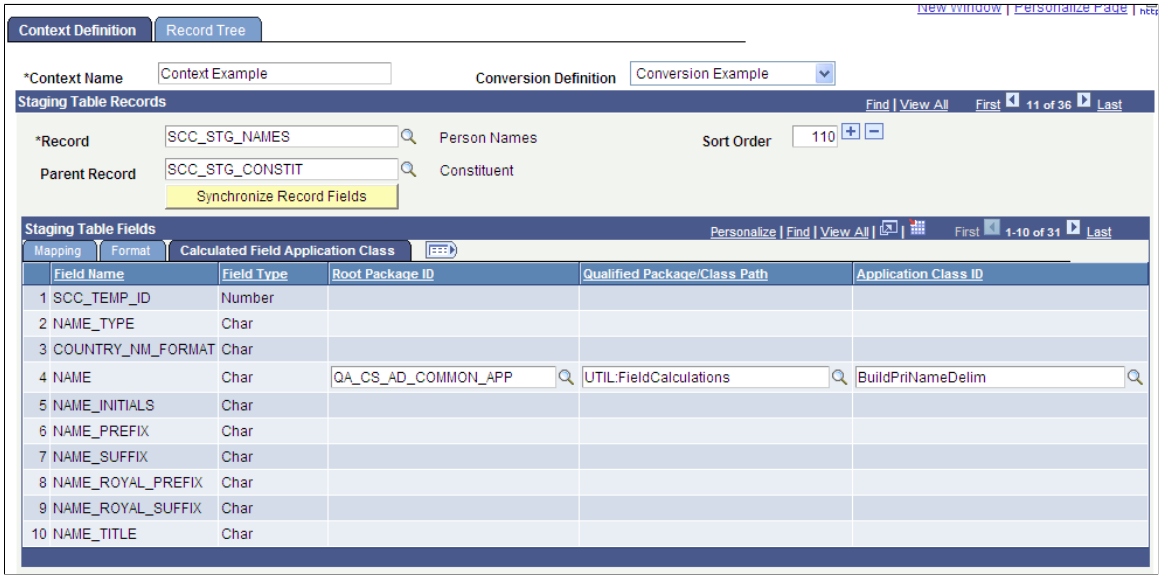
The Calculated Field Application Class tab is available only if *Calculated* is selected as the Mapping Action for one or more fields.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access the Calculated Field Application Class tab.

Image: Context Definition page - Calculated Field Application Class tab

This example illustrates the fields and controls on the Context Definition page - Calculated Field Application Class tab. You can find definitions for the fields and controls later on this page.



Supply the name of your Application Package in the Root Package ID column. Prompt values will then become available for the Qualified Package/Class Path and Application Class ID columns. Select the class path and path ID that reflect the calculation that you want to associate with the Record/Field Context Definition.

The PeopleCode application class for a calculated field should not be dependent on any other calculated field.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Viewing the Record Tree

Access the Record Tree page (Set Up SACR, System Administration, Utilities, File Parser, Context Definition, Record Tree).

Image: Record Tree page (1 of 2)

This example illustrates the fields and controls on the Record Tree page (1 of 2). You can find definitions for the fields and controls later on this page.

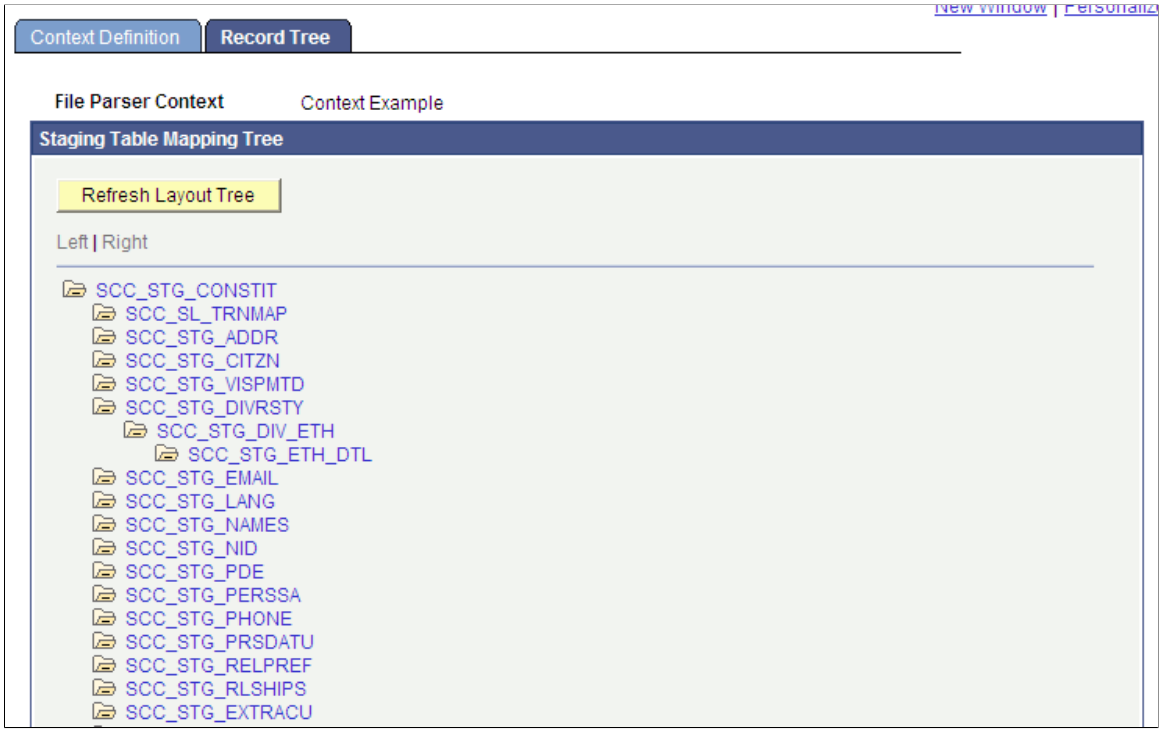


Image: Record Tree page (2 of 2)

This example illustrates the fields and controls on the Record Tree page (2 of 2). You can find definitions for the fields and controls later on this page.

Staging Table Mapping Tree

Click the Refresh Layout Tree link to view the hierarchy of staging tables for this conversion. Review the tree to verify that the records are in the correct parent-child order.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Refresh the layout tree after you add staging tables or change the sort order.

Copying a Context Definition

Access the Copy Context Definition page (Set Up SACR, System Administration, Utilities, File Parser, Copy Context Definition).

You can copy an existing context definition and save it with a different name. You can then change the new context definition as necessary.

On the search page, select the context definition to copy. Then, enter the name of the new context definition and save the page. To edit the new definition, access it from the Context Definition page.

Setting Up File Mapping Definitions

This section discusses how to:

- Create a file definition.
- Define a file layout.
- Map the file and convert the data.
- Preview the record tree.
- Preview the converted data.

Pages Used to Set Up File Mapping Definitions

Page Name	*Definition Name*	*Navigation*	*Usage*
File Definition	SCCFP_FILE_DFN	Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition	Map the external file to the context containing the destination records.
File Layout	SCCFP_FILE_LAYOUT	Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, File Layout	Identify the fields from the external file, and specify the action to use to convert data and place it on the destination staging table records.
Mapping	SCCFP_FILE_MAP	Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, Mapping	Map the fields in the external file to fields in the destination record.
Preview Layout	SCCFP_FILE_PRVW	Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, Preview Layout	View the record tree to determine if the correct data is set to be mapped and converted.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

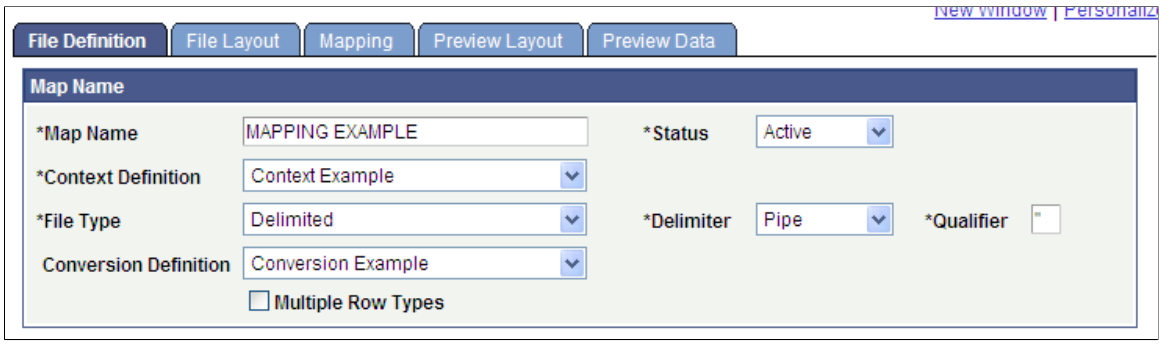
Page Name	*Definition Name*	*Navigation*	*Usage*
Preview Data	SCCFP_FILE_PRVW2	Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, Preview Data	Identify the external file and view the data as it will be converted before running the File Parser process to place the converted data into the destination records.
Copy File Map Definition	SCCFP_CPY_MAP_DFN	Set Up SACR, System Administration, Utilities, File Parser, Copy File Map Definition	Copy an existing file map definition under a new definition name.

Creating a File Definition

Access the File Definition page (Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition).

Image: File Definition page

This example illustrates the fields and controls on the File Definition page. You can find definitions for the fields and controls later on this page.

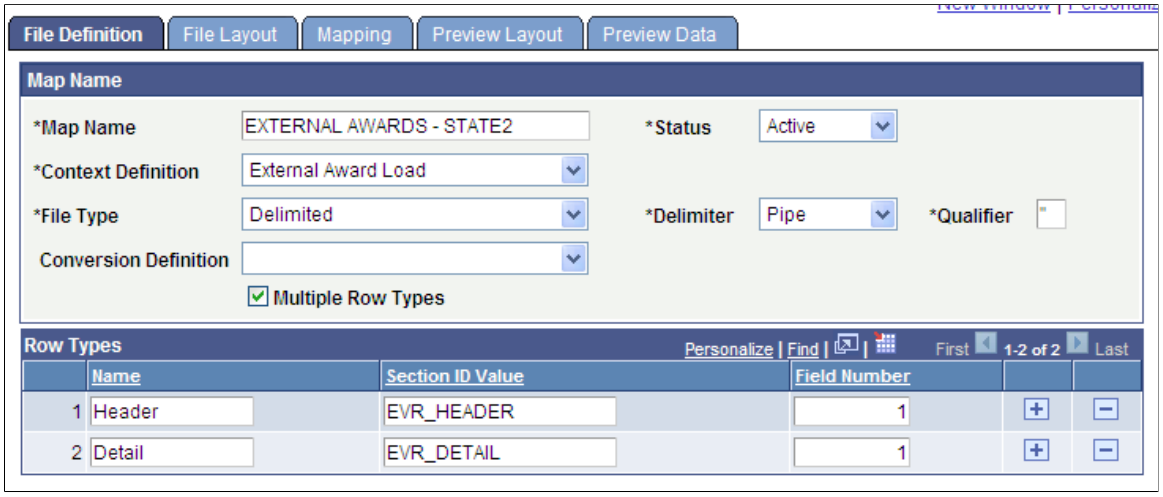


 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

This is an example of a File Definition page with multiple row types:

Image: File Definition page – Multiple Row Types

This example illustrates the fields and controls on the File Definition page – Multiple Row Types. You can find definitions for the fields and controls later on this page.



Map Name

Map Name

Enter a name for this mapping of the external file.

Status

Enter the status of file mapping definition.

The default value is *Pending* when the mapping definition is initially added. Only file mapping definitions with an *Active* status are available for selection when converting external file data.

Context Definition

Enter the context definition to use for this mapping. Context definitions are set up on the Context Definition page.

If a conversion definition is associated with the context on the Context Definition page, the system displays the name of it in the Conversion Definition field.

Important! For a saved file mapping definition, if you change this field's value, the system will automatically delete the setup in both the File Layout and Mapping pages. This is because the system assumes that the target records (context) are different. Do not change this field's value unless you want to repeat your setup.

File Type

Specify the type of the external file, either *Flat* or *Delimited.*

If you enter *Delimited,* you must also verify or enter values for the delimiter and the qualifier.

Delimiter

Appears only if the file type is *Delimited.*

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Enter the item in the external file that signifies the end of one value and the beginning of another. Values are:

Comma

Pipe

Semicolon

Tab

Qualifier

Appears only if the file type is *Delimited.*

Enter the character in the external file that qualifies the end of one field and the beginning of another. The default qualifier is ″ (quotation mark).

Note: If the external file does not contain any field qualifier, leave the default value unchanged. This will not have any impact on processing the file.

Conversion Definition

Enter the conversion definition to use. Conversion definitions are set up on the Field Conversion Definition page.

If a conversion definition is associated with this context on the Context Definition page, the system displays it as the default value. You can change the conversion definition.

Multiple Row Types

Select if the external file contains multiple row types. For example, the external file might have a header row, a detail row, and a trailer or total row.

When Multiple Row Types is selected, the File Row Type Detail group box appears, which is where you must identify each row type to which data is to be converted.

Note: The Force Insert functionality is not available when multiple row types are specified.

Row Types

This group box is available only when the Multiple Row Types check box is selected.

Name

Enter a name to describe the general purpose or type of the row.

Section ID Value

Enter the type of value that appears in the row.

Starting Position and **Field Length**

These two fields appear when the file type is *Flat*.

Enter the position in which the field value for the row begins in the Starting Position field, and enter the maximum character length of the field to read in the Field Length field.

Field Number

This field appears when the file type is *Delimited*.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Indicate the field that contains the field value for the row.

The Field Number (for delimited files), and Starting Position and Field Length (for flat files) indicate the field or column that the File Parser process should look at to determine the row type. For example, shown in the preceding exhibit, if Field Number *1* in the file has a value *EVR_HEADER* then the row type = *Header*.

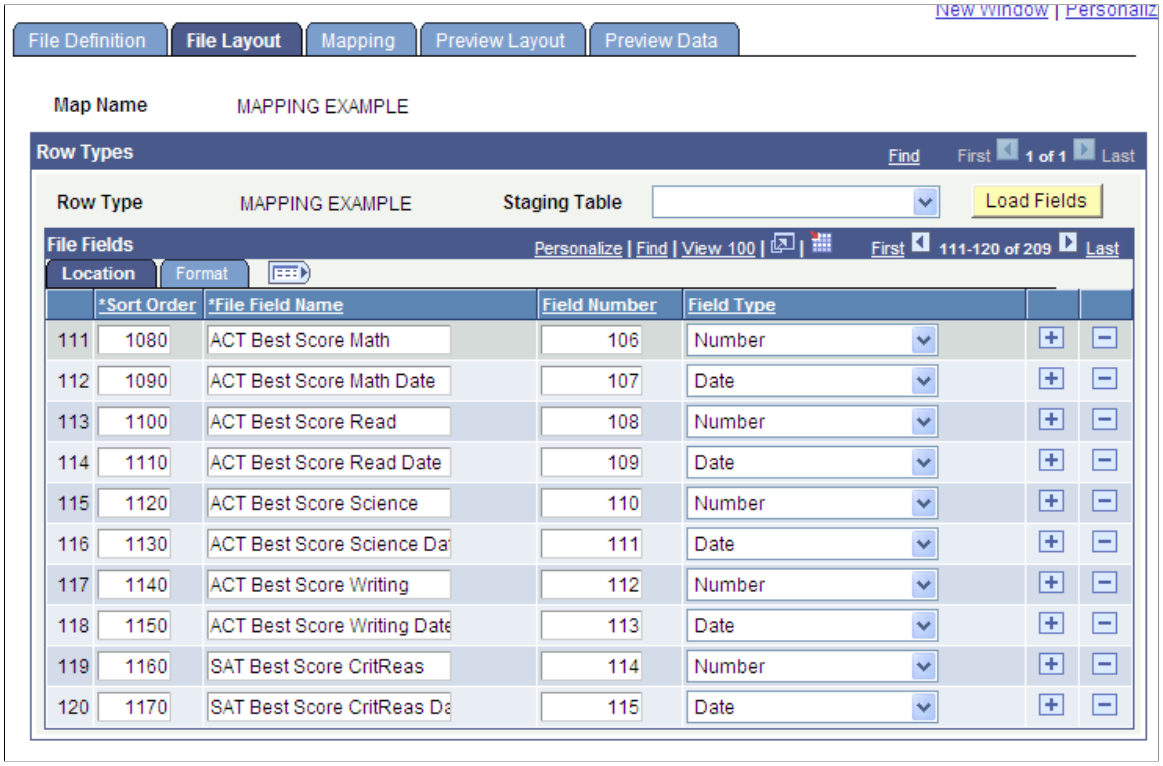
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Define a File Layout

Access the File Layout page (Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, File Layout).

Image: File Layout page - Location tab

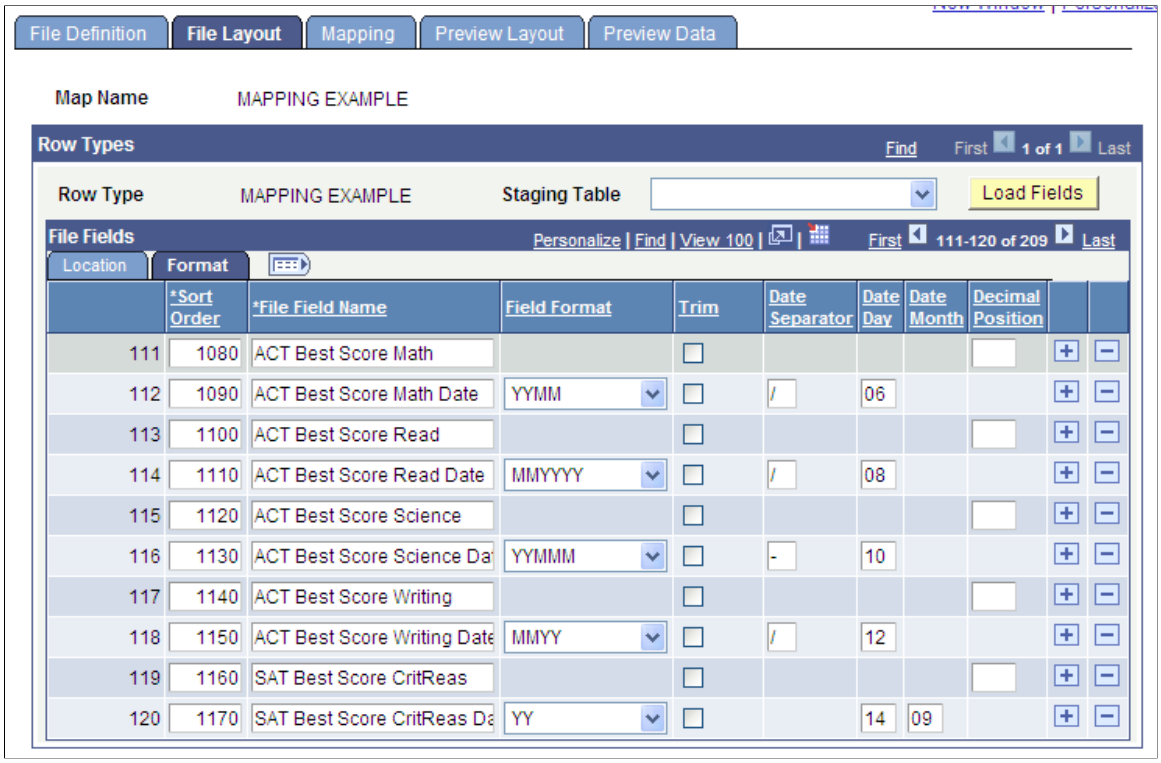
This example illustrates the fields and controls on the File Layout page - Location tab. You can find definitions for the fields and controls later on this page.



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Image: File Layout page - Format tab

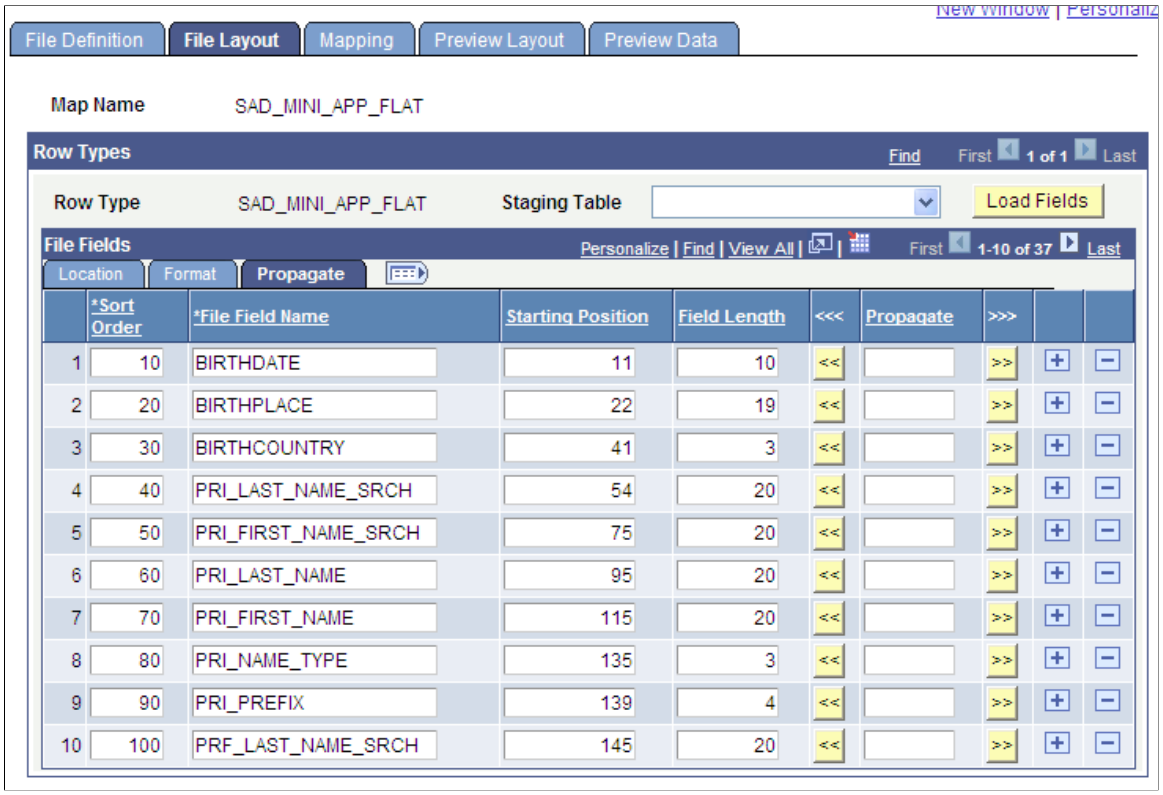
This example illustrates the fields and controls on the File Layout page - Format tab. You can find definitions for the fields and controls later on this page.



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Image: File Layout page - Propagate tab

This example illustrates the fields and controls on the File Layout page - Propagate tab. You can find definitions for the fields and controls later on this page.



You control the layout of the file by identifying the fields on the external file to be converted. You can either enter file field names from the external file or you can load and populate the fields from the staging table, view the field controls, and make changes as needed.

Row Types

The page contains a row for each row type identified on the File Definition page. Enter data for each row type.

Staging Table and **Load Fields**

(Optional) Enter the name of the staging table into which data for this row type is to be placed. Only the staging tables associated with this context are available.

Click Load Fields to load a list of the fields from the staging table into the File Fields group box.

Loading the fields can save data entry.

File Fields

The File Fields group box lists the field names to reference in the external file. Use these file names to map to fields on the staging tables on the Mapping page.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

File Fields - Location Tab

The Location tab lists the location of the field on the external field. You can view and change the sort order, the field name, and the field type.

For a flat file, the location is the starting position and field length.

For a delimited file, the location is the field number.

Field Type

Enter the type of field to use.

The values are:

Character

Date

Date Time

Number

Signed Decimal

Time

Note: The field type that you select here must be applicable to the corresponding field displayed on the Field Mapping Detail section of the Mapping page. For example, if setting up a field for ACT Math Score, the Field Type should be set as *Number* to correspond with the Field Type of the SCORE field on SCC _STG_TESTCMP. Fields defined with the Field Type of *Time* must be in the format of hh:mm:ss AM/PM, such as *12:15:42 PM.*

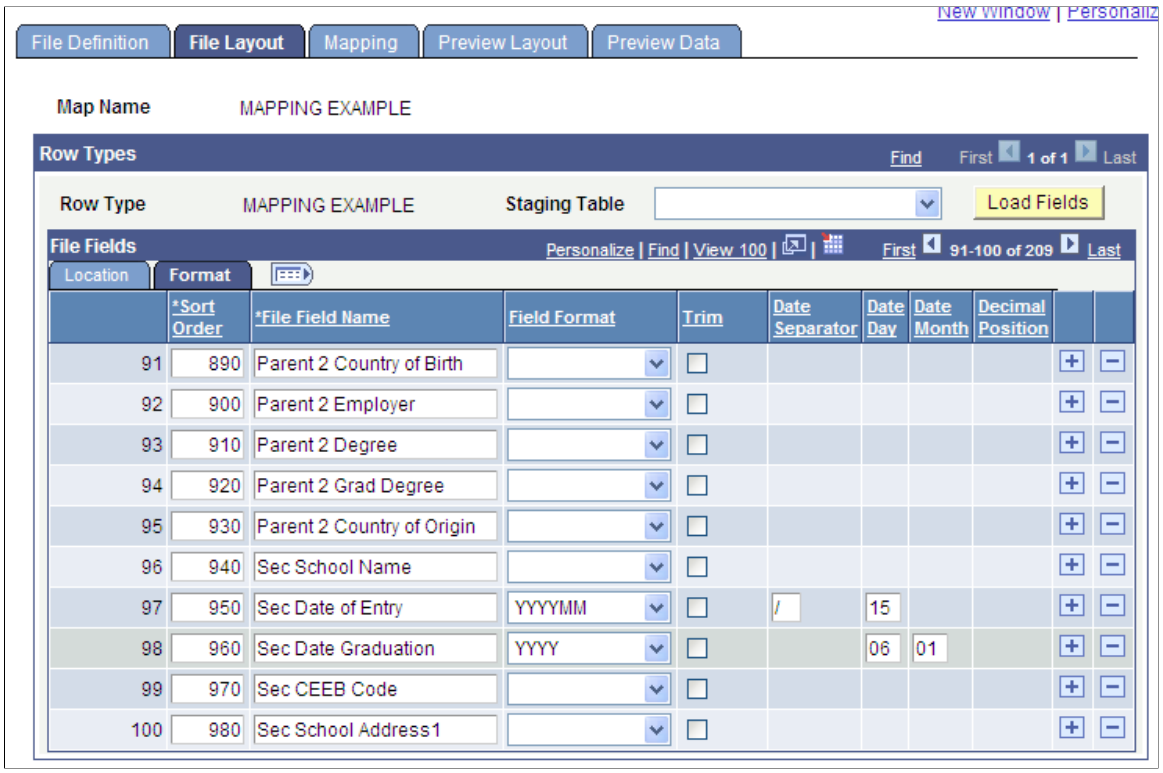
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

File Fields - Format Tab

The Format tab enables you to view and change field formats, trim leading and trailing spaces, enter date separators for date fields, and change decimal placement for currency amounts.

Image: File Layout page - Format tab

This example illustrates the fields and controls on the File Layout page - Format tab. You can find definitions for the fields and controls later on this page.



Date Day

This field appears when you define a Field Type of *Date* and select a date field format that contains only month and year (for example, *MMYYYY* or *YYMMM*). Select a value (1-28) of your choice to be assigned to the incoming date value and the subsequent mapped record/field.

Date Month

This field and the Date Day field appear when a date format contains only year (for example, *YYYY* or *YY*). For the month value, select a value (1-12) of your choice, and then select a value (1-28) to be assigned to the incoming date value and the subsequent mapped record/field.

File Fields - Propagate Tab

This tab appears for flat files only. The Propagate tab enables you to adjust the starting position and length of a field when a new field is added to an existing definition.

Click to decrease the starting position of the field.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Click to increase the starting position of the field.

Mapping the File and Converting the Data

Note that if you are using a PDF version of this document and the graphics appear unclear, we recommend that you adjust the zoom or magnification tool of Adobe Reader.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Access the Mapping page (Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, Mapping).

Image: Mapping page - Mapping tab

This example illustrates the fields and controls on the Mapping page - Mapping tab. You can find definitions for the fields and controls later on this page.

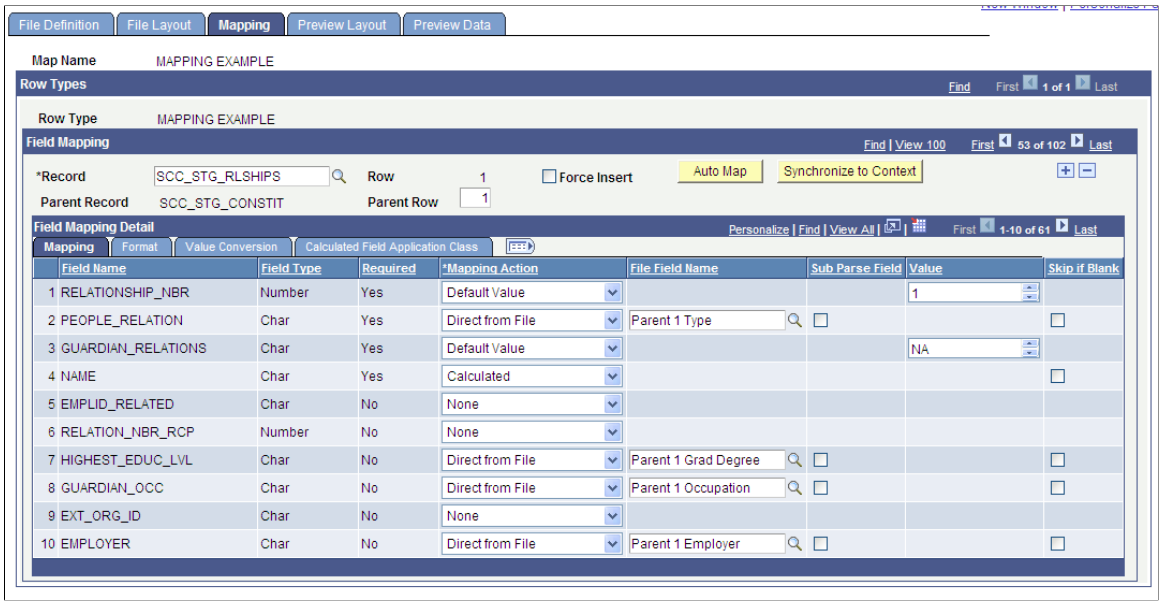
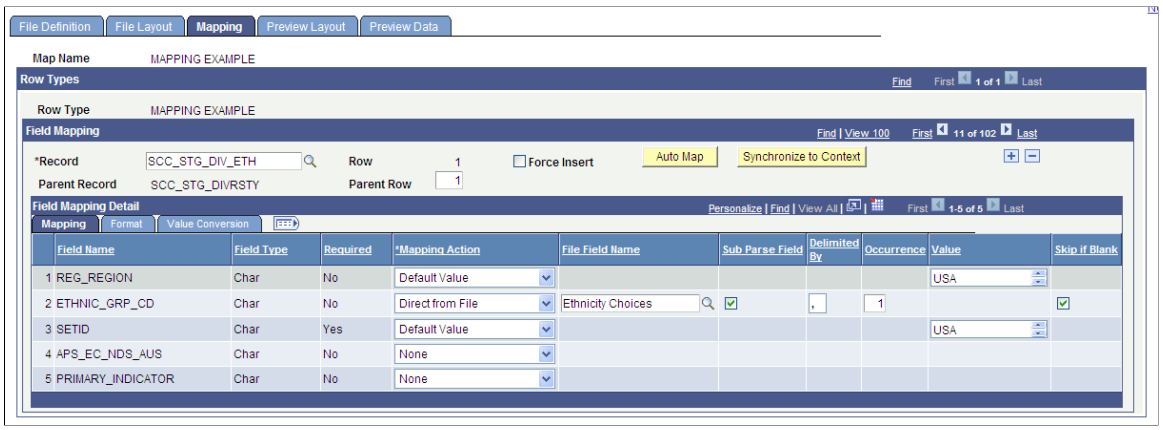


Image: Mapping page - Mapping tab with the Sub Parse Field check box selected

This example illustrates the fields and controls on the Mapping page - Mapping tab with the Sub Parse Field check box selected. You can find definitions for the fields and controls later on this page.



 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Image: Mapping page - Format tab

This example illustrates the fields and controls on the Mapping page - Format tab. You can find definitions for the fields and controls later on this page.

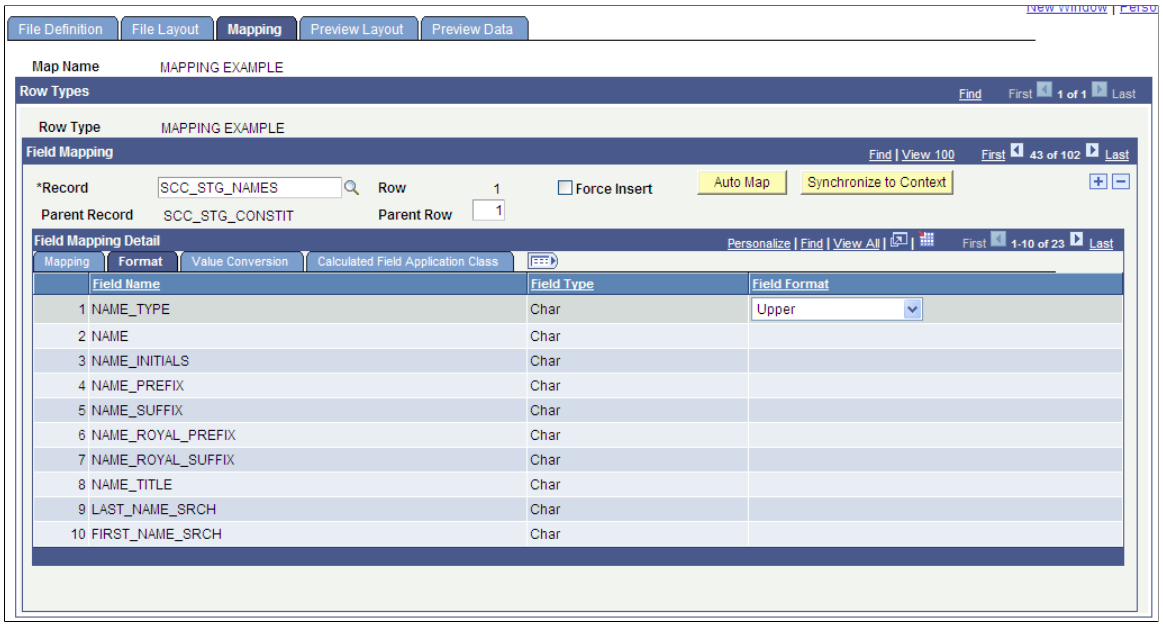
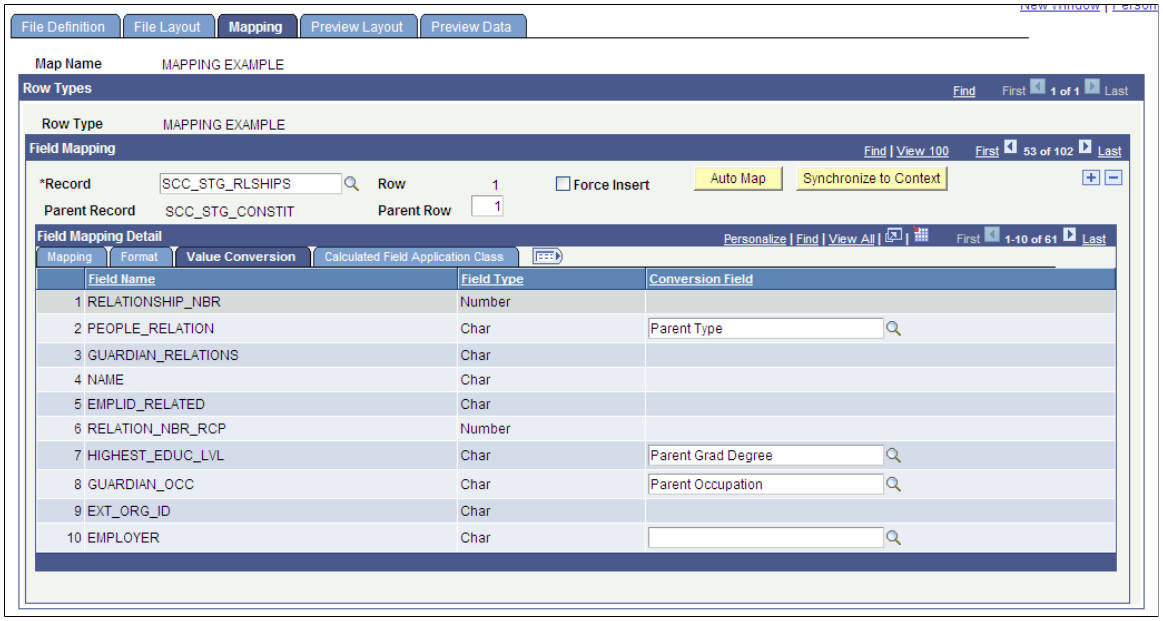


Image: Mapping page - Value Conversion tab

This example illustrates the fields and controls on the Mapping page - Value Conversion tab. You can find definitions for the fields and controls later on this page.



Enter data to define how to update the staging record when the external file is processed. Fields on the staging table may take data directly mapped from the external file or set with a default value when external file is processed.

Note: Click Save on the File Layout page before using the Mapping page. Saving the file mapping definition and layout causes the file field names to be available on the Mapping page.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Row Types

Each row type identified on the File Definition page and designed on the File Layout page is listed in this group box. View or enter mapping instructions for each field to map the field to the staging tables.

Field Mapping

Data in this group box defines how to update the staging tables and fields when the external file is processed.

Record

Enter the record to use as the staging table. Available records are from the related context definition.

When the process runs, it updates the specified staging table with the fields defined in the row type and File Layout page.

Once a staging table is selected in the Record field, based on the context definition, all visible fields from the staging table are displayed.

Note: If a staging table with mapped file fields has a parent record, you must include the parent record in the field mapping even if the parent has no mapped file fields. If the parent record does not contain any reference to a mapped value (that is, you have selected *Direct from File* Mapping Action for the parent record), you must select the Force Insert check box.

Row

Identifies the row of the staging table into which the File Parser process will insert the converted data.

Note: In most cases, the row number is *1*. However, sometimes one row in the file may need to be normalized into more than one row in the staging tables. For example, PeopleSoft Financial Aid's external awards have award files that contain more than one disbursement on a single file row. The External Award staging tables store disbursements in a normalized manner to support a large number of disbursements for each award. To map a single file row to more than one staging table record, you must add rows on the field mapping level. Each of these rows must be mapped to the correct file field. The disbursement table is keyed to a unique sequence and the keys must be correctly mapped. In most cases, set the Mapping Action to *Default Value,* and enter in the Value field a numeric sequence that equates to the number of disbursements.

Force Insert

Select to have the File Parser process insert a record if you have not selected the *Direct From File* mapping action for any field in the record.

Force Insert functionality allows the end user to write to a staging table while not relying on the existence of mapped value(s) from a specific file.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

When Force Insert is enabled, File Parser inserts a row into the specified staging table. File Parser presumes that the end user recognizes that the target table is a child to a parent table structure and that there is sufficient data either inherited from the parent and/or defined at the mapped level to insure data integrity.

Note: The Force Insert check box is not available when multiple row types are specified. A calculated field must be used to force insert a record in this case.

Auto Map

(Optional) Click for the system to compare fields on the File Layout page to fields in the staging table and place fields with the same name into the staging tables.

Synchronize to Context

(Optional) Click for the system to compare fields on the Context Definition page to fields on the Mapping page and to remove fields from the mapping that are not in the context definition. Synchronizing content is helpful when the context definition has been updated after the file mapping definition is created.

Note: If changes have been made to a record definition you are using, it is required that you synchronize content to successfully run the File Parser process.

Field Mapping Detail - Mapping Tab

The Mapping tab appears for each record and displays all of the visible fields from the staging table and their mapping actions.

Mapping Action

Enter the mapping action for the process to use when updating the target staging table.

Values are:

Default Value to enable you to hard code the value when the file is processed. The Value column appears where you can enter the default value to post to staging table. For example, the External Award staging table (SFA_EASTAGE_DTL) requires that a value posts to the external award report SFA_EA_REPORT_CD field. In most cases the external file will not have a corresponding field for this value. Select *Default Value* and provide appropriate value in the Value column.

Direct from File to update the staging table with a field defined on the File Layout page. The File Field Name column appears, and the available values are fields defined in File Layout page. You can click the Auto Map button to cause the system to populate the File Field Name fields with the available values if the File Field Name is the same as the Field Name.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Inherit from Parent to carry forward the value from the parent record.

None to take no action nor update the staging table field.

Process Instance to bring in the current process instance of the batch process.

Sequence to assign a unique sequence number, used primarily to create unique key values. Sequencing starts at 1 and increments by 1 for each row retrieved.

Unique Counter to assign a unique numeric index created from the next number on the referenced counter record and field. The Population Selection process uses the PeopleTools *GetNextNumberWithGapsCommit() function* to get the next value and update the record.

Calculated to use a PeopleCode application class that calculates the field value.

When you select *Calculated*, the Calculated Field Application Class tab appears. Specify the application class details on this tab.

File Field Name

Available when Mapping Action is *Direct from File.*

Available values are fields defined in File Layout page.

Sub Parse Field

Available when Mapping Action is *Direct from File*. This attribute is used when an incoming field may contain multiple responses or answers to a specific question. For example: Question: *What is your ethnic background*. Answer: *I am White and Native American Indian*. Each of these responses would constitute a separate row in the ethnicity tables.

When you select the Sub Parse Field check box, the Delimited By and Occurrence fields are available for entry:

Delimited By: Specify the character that is used to separate the values within the incoming mapped field.

Occurrence: Specify a numeric value corresponding to the occurrence within the field that you wish to assign to the specific staging table.

Repeat this construct for as many responses that could potentially be contained within the incoming mapped field.

Value

Available when Mapping Action is *Default Value.*

Enter the value in the correct format for field.

Skip if Blank

Select if you do not want the File Parser process to insert a blank row if the field value is blank.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

If you clear this check box, the File Parser process inserts a blank row if the field value has a blank value. For example, if you do not select this check box and a student does not provide a preferred name, the File Parser process will insert a blank name in the staging table with a name type set to *Preferred*.

Note: The Skip if Blank check box is available only for the following Mapping Actions: *Direct from File*, *Inherit from Parent*, and *Calculated*.

Field Mapping Detail - Format Tab

The Format tab appears when the mapping action is *Default Value.*

View and enter data to format the mapped fields.

Field Mapping Detail - Value Conversion Tab

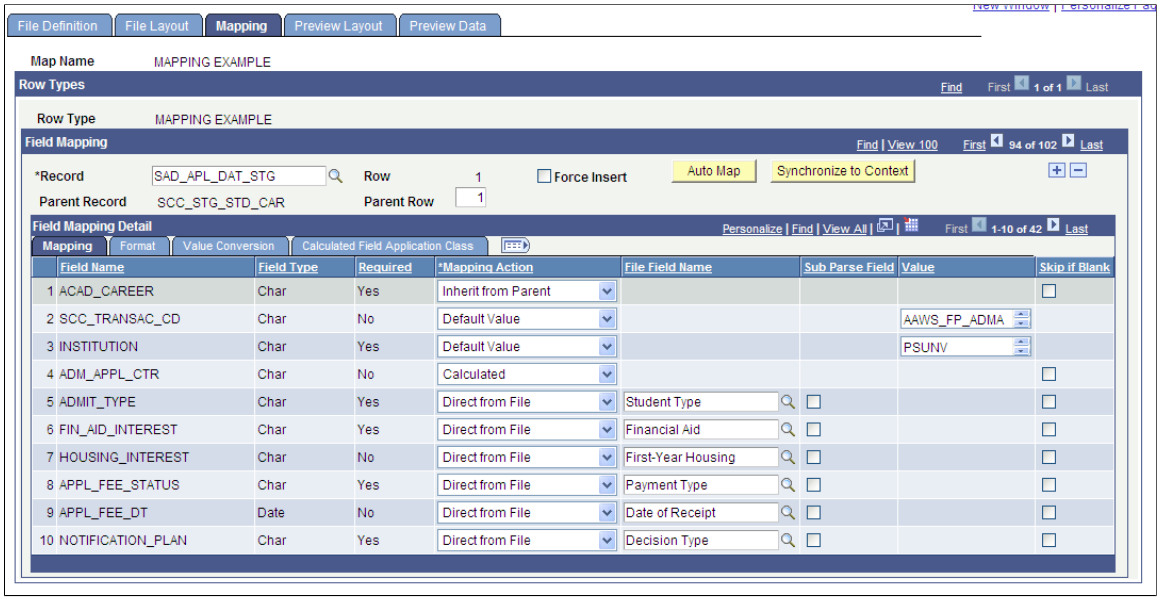
The Value Conversion tab appears when the mapping action is *Direct from File.* You can optionally assign a conversion value to fields that are mapped directly from the external file.

Field Mapping Detail - Calculated Field Application Class Tab

When you select a Mapping Action of *Calculated* on the Mapping tab, the Calculated Field Application Class tab becomes available.

Image: Mapping page - Mapping tab with a calculated Mapping Action selected

This example illustrates the fields and controls on the Mapping page - Mapping tab with a calculated Mapping Action selected. You can find definitions for the fields and controls later on this page.

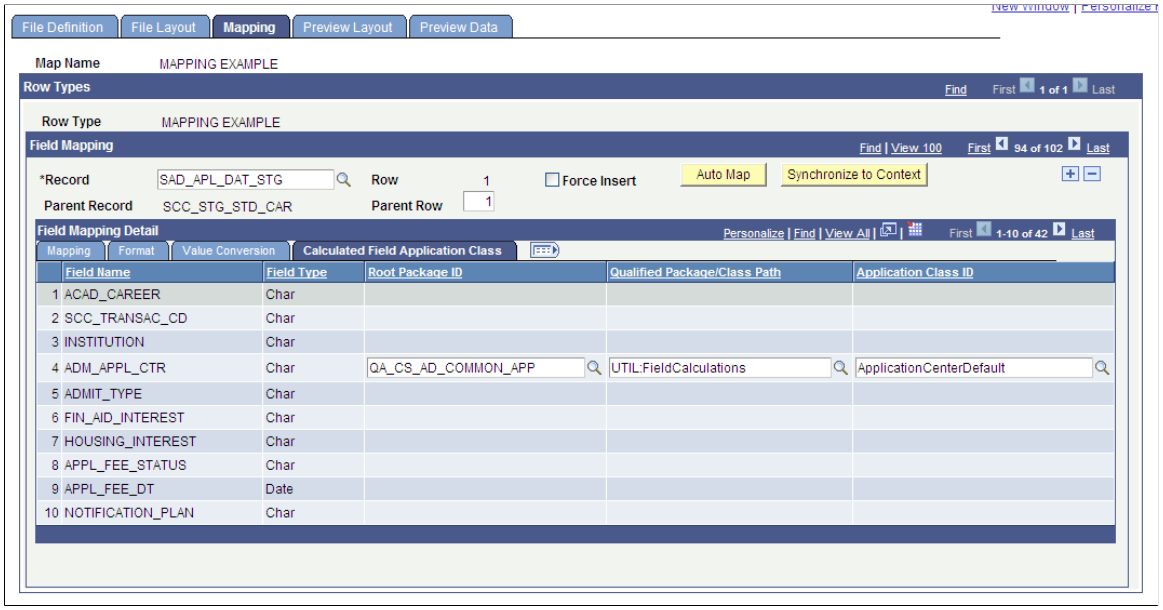


Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

On the Calculated Field Application Class tab, you then need to provide the name of the Application Package that you previously defined in Application Designer under the Root Package ID column:

Image: Mapping page - Calculated Field Application Class tab

This example illustrates the fields and controls on the Mapping page - Calculated Field Application Class tab. You can find definitions for the fields and controls later on this page.



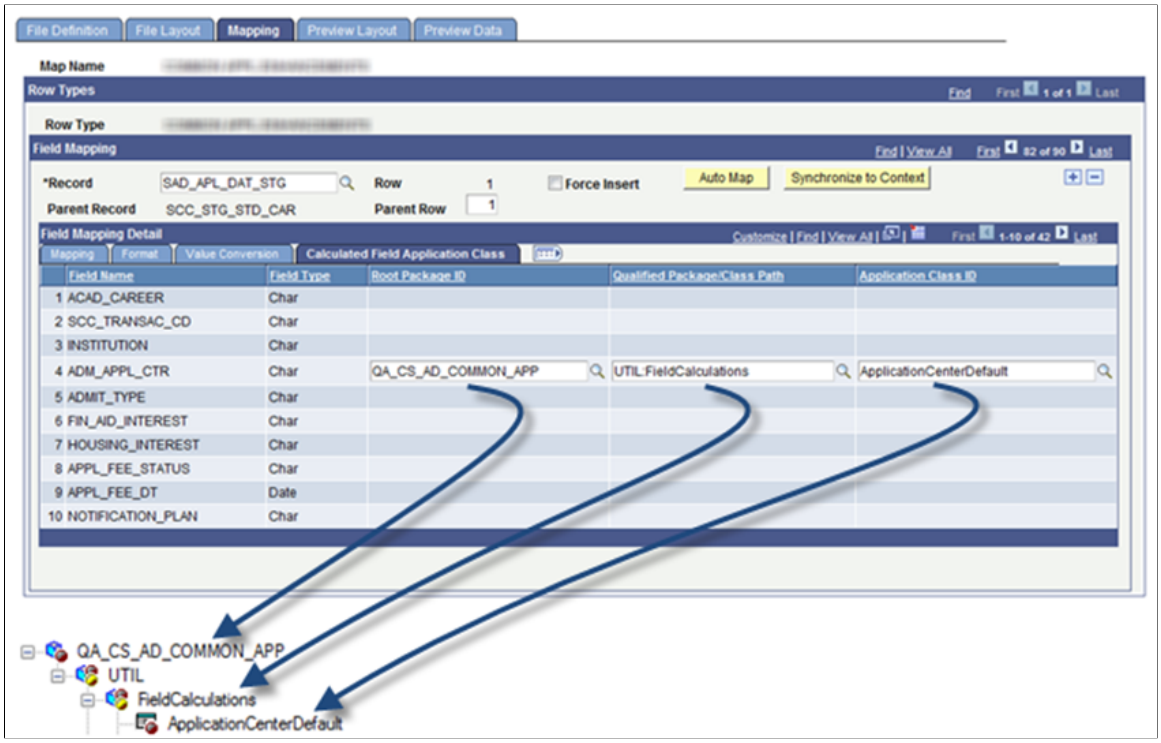
Once you select the application package that pertains to this specific calculation field, you must provide the Qualified Package/Class Path and the Application Class ID.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

The following graphic illustrates how the specified field values on the Calculated Field Application Class tab relate to the user defined application package:

Image: Calculated Field Application Class tab values relating to an application package

This example illustrates the fields and controls on the Calculated Field Application Class tab values relating to an application package. You can find definitions for the fields and controls later on this page.



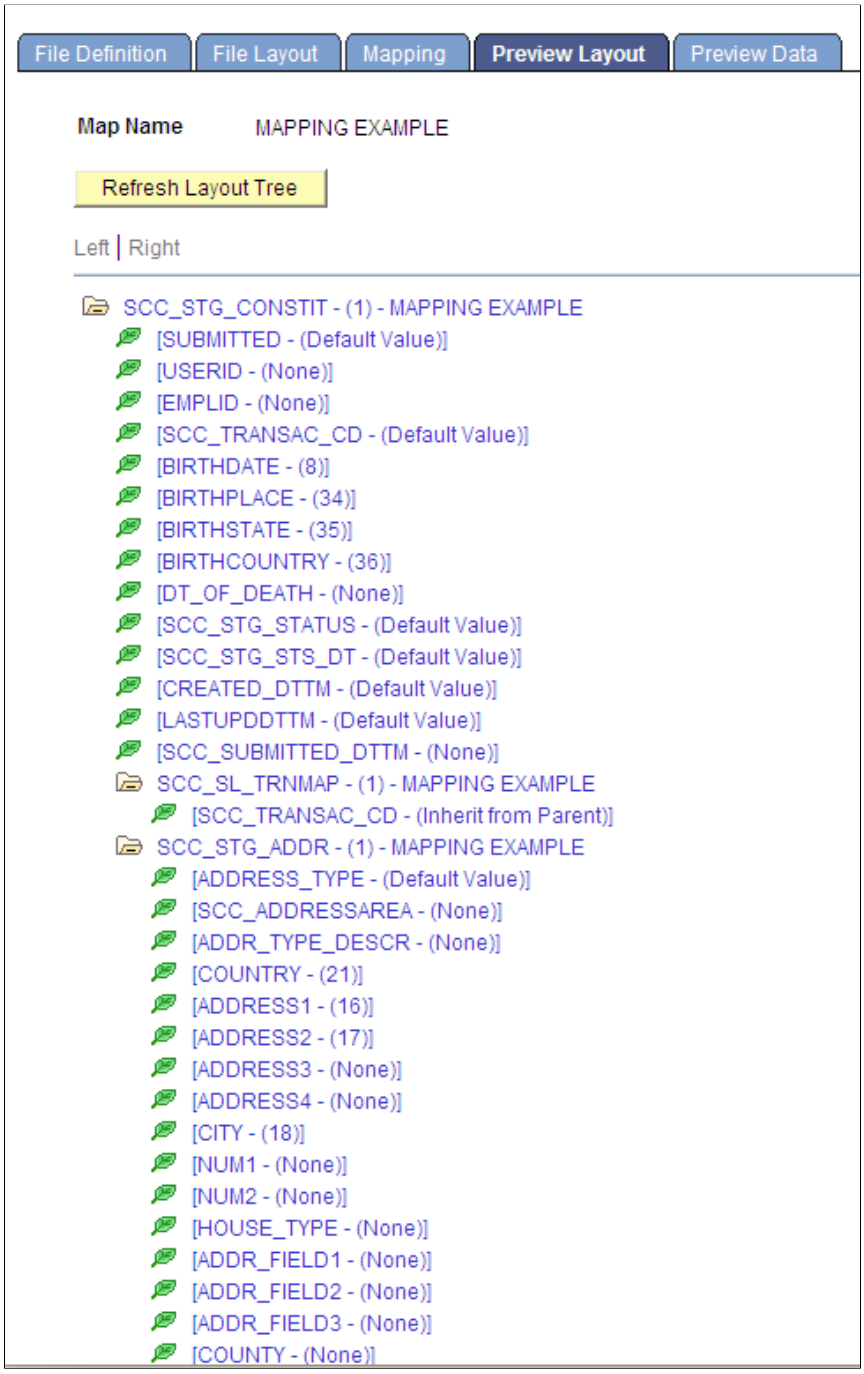
Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Previewing the Record Tree

Access the Preview Layout page (Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, Preview Layout).

Image: Preview Layout page

This example illustrates the fields and controls on the Preview Layout page . You can find definitions for the fields and controls later on this page.



After you design the layout and map the file fields to the staging tables, click the Refresh Layout Tree button to view the hierarchy of the staging tables and fields; and determine how each field will be updated

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

when external file is processed. You should preview the layout before you run the process to move the data into the target tables.

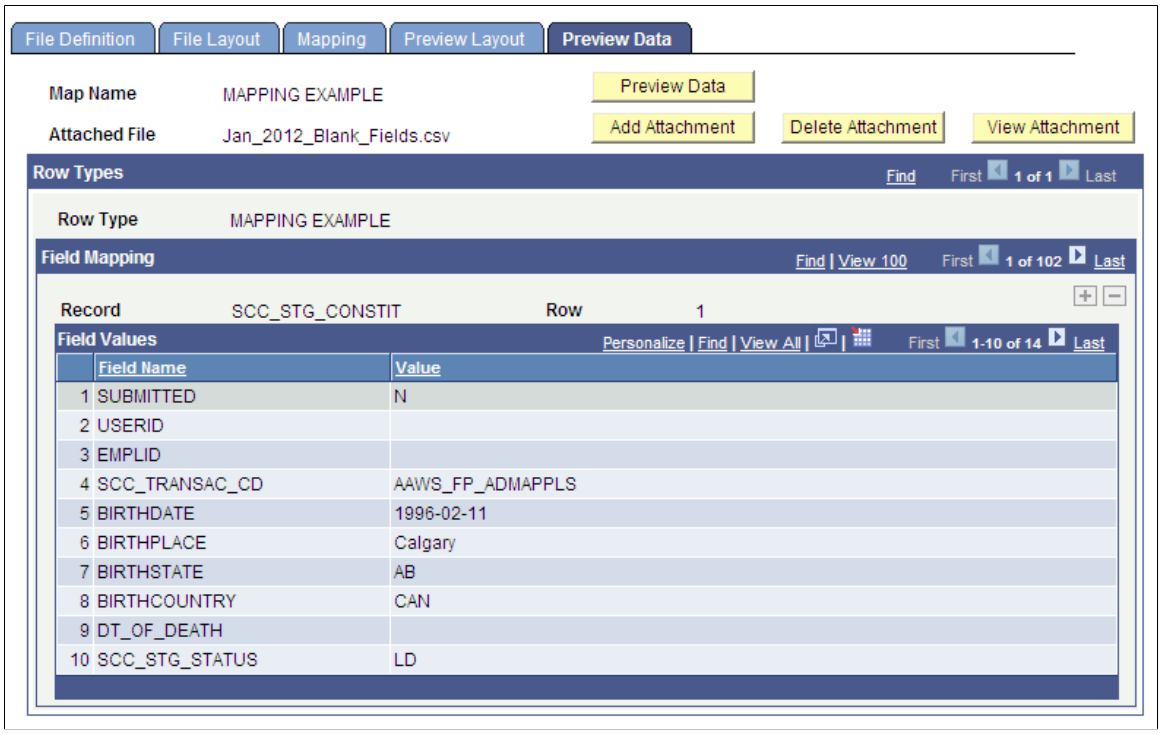
If the Preview Layout page displays a field with one or more numbers in parenthesis, this means the field is mapped with a Mapping Action of *Direct From File*. For Delimited File Types, the displayed number corresponds to the Field Number on the File Layout page. For example, BIRTHDATE - (8) means that this field is mapped to Field Number 8 on the File Layout page, which is Date of Birth. For Flat File Types, the first number corresponds to the starting position of the field and the second number corresponds to the field length. For example, BIRTHDATE – (11,10) means that this field begins in position 11 and has a length of 10.

Previewing the Converted Data

Access the Preview Data page (Set Up SACR, System Administration, Utilities, File Parser, File Mapping Definition, Preview Data).

Image: Preview Data page

This example illustrates the fields and controls on the Preview Data page . You can find definitions for the fields and controls later on this page.



After mapping is complete, you can preview the data with a test file to confirm the file field layout, mapping to staging tables, and field formats and conversions. Perform this preview before you run the process to move the data to target tables. The preview feature shows only the first row of each row type.

Note: Use a small test file. Previewing with a large file would waste system resources, be too large to upload as an attachment, and potentially degrade the application server.

Add Attachment Click and browse to attach a sample file. The file to be attached should be small but have at least one row for each file type.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Preview Data Click to display data in the attached file according to the specifications.

View Attachment Click to view the sample file with data.

Delete Attachment Click to delete the sample file.

Running the File Parser Process

This section discusses how to run the File Parser process.

Page Used to Run the File Parser Process

Page Name	*Definition Name*	*Navigation*	*Usage*
Run File Parser	SCCFP_RUNCNTL	Set Up SACR, System Administration, Utilities, File Parser, Run File Parser	Enter the path to the external file to be converted, and run the process to convert the data and place it into the staging tables.

Running the File Parser Process

Access the Run File Parser page (Set Up SACR, System Administration, Utilities, File Parser, Run File Parser).

Enter parameters, and then run the File Parser process to convert the data and place it into the staging tables.

File Path Enter the path to the external file to be converted.

The path must be accessible to Process Scheduler.

File List Indicator

Select this check box to indicate that the file specified in the File Path field points to a list of files to be converted.

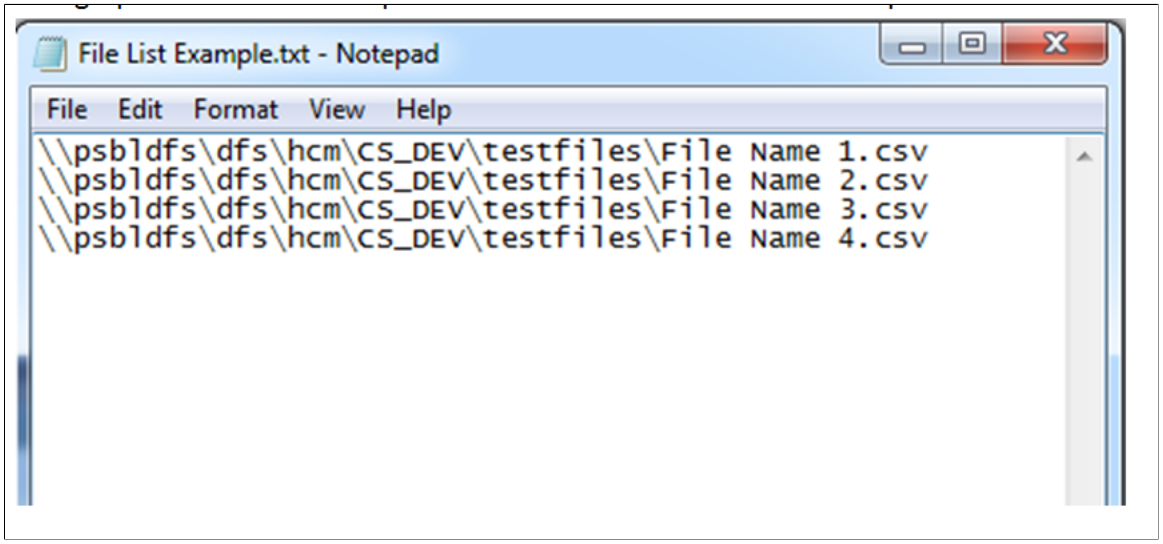
To load multiple files, you must create a document (for example .txt) that lists all of the files that you want to load. In this document, for each file, enter the entire file path on which the file resides.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

This graphic shows an example text document for the File List Indicator option:

Image: Example text document for the File List Indicator option

This example illustrates the fields and controls on the Example text document for the File List Indicator option. You can find definitions for the fields and controls later on this page.

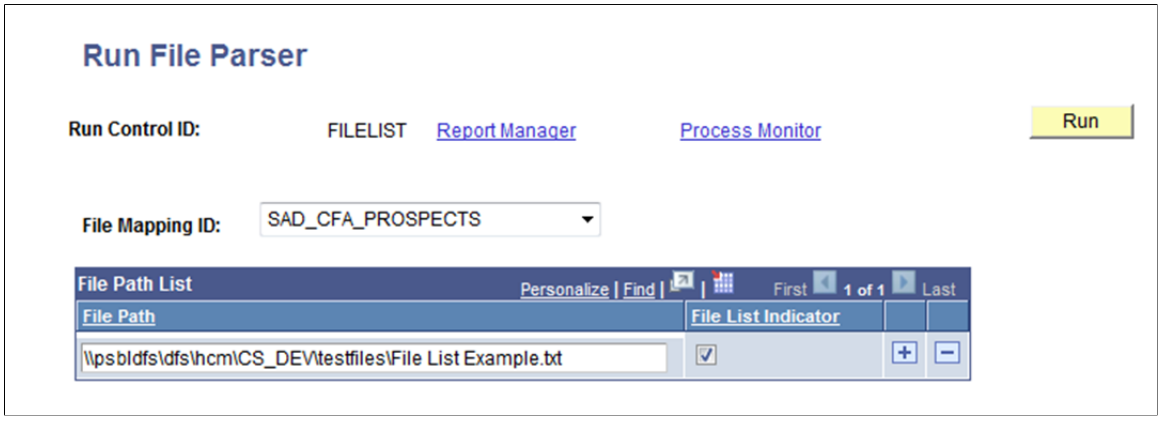


Enter the path and name of this document in the File Path field.

This is an example of the run control page with the File List Indicator selected:

Image: Example of the Run File Parser page with the File List Indicator selected

This example illustrates the fields and controls on the Example of the Run File Parser page with the File List Indicator selected. You can find definitions for the fields and controls later on this page.



Mapping the File Parser Process for Population Selection

The Population Selection process uses the File Parser process when the selection tool for identifying the population is external file. When using an external file to select IDs for a process, you must map the fields required by the process to the target table fields for the application process that is integrated with Population Selection.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

This section discusses how to map the File Parser process for Population Selection.

Related Links

Using the Population Selection Process

Page Used to Map File Parser for Population Selection

Page Name	*Definition Name*	*Navigation*	*Usage*
Population Selection File Map	SCCFP_PS_FILE	Set Up SACR, System Administration, Utilities, File Parser, Population Selection File Map Click the Create File Mapping or Edit File Mapping link in the Population Selection group box on any run control page when the Population Selection tool is an external file.	Map the File Parser process for integration with the Population Selection process.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Mapping the File Parser Process for Population Selection

Access the Population Selection File Map page (Set Up SACR, System Administration, Utilities, File Parser, Population Selection File Map).

Image: Population Selection File Map - Mapping tab

This example illustrates the fields and controls on the Population Selection File Map - Mapping tab. You can find definitions for the fields and controls later on this page.

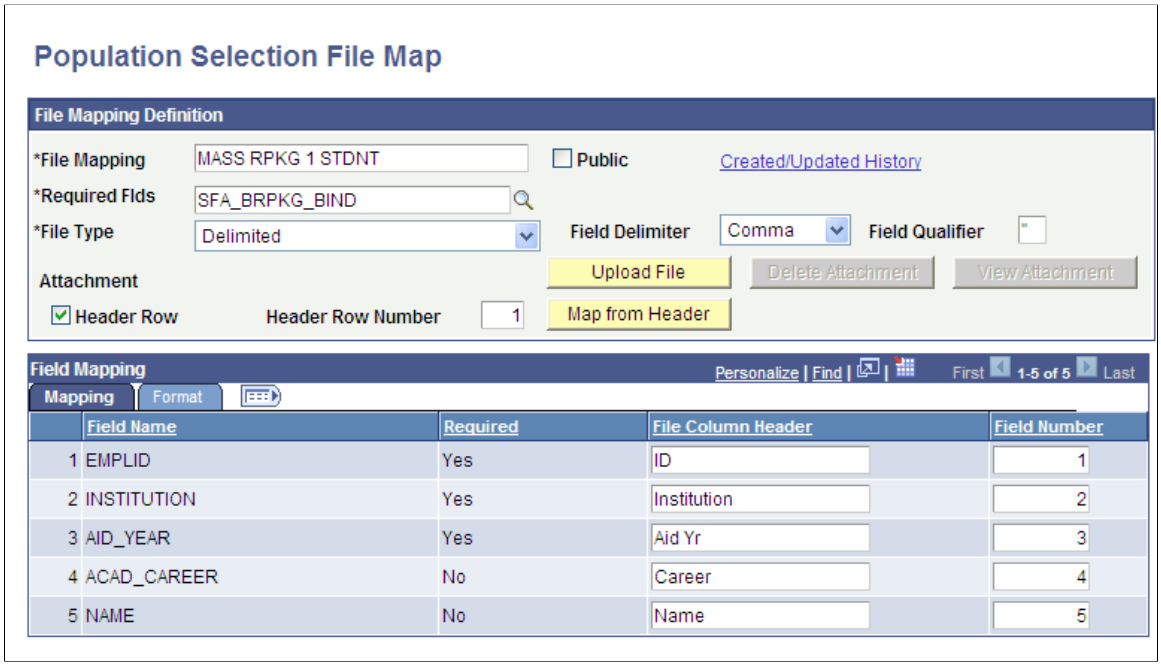
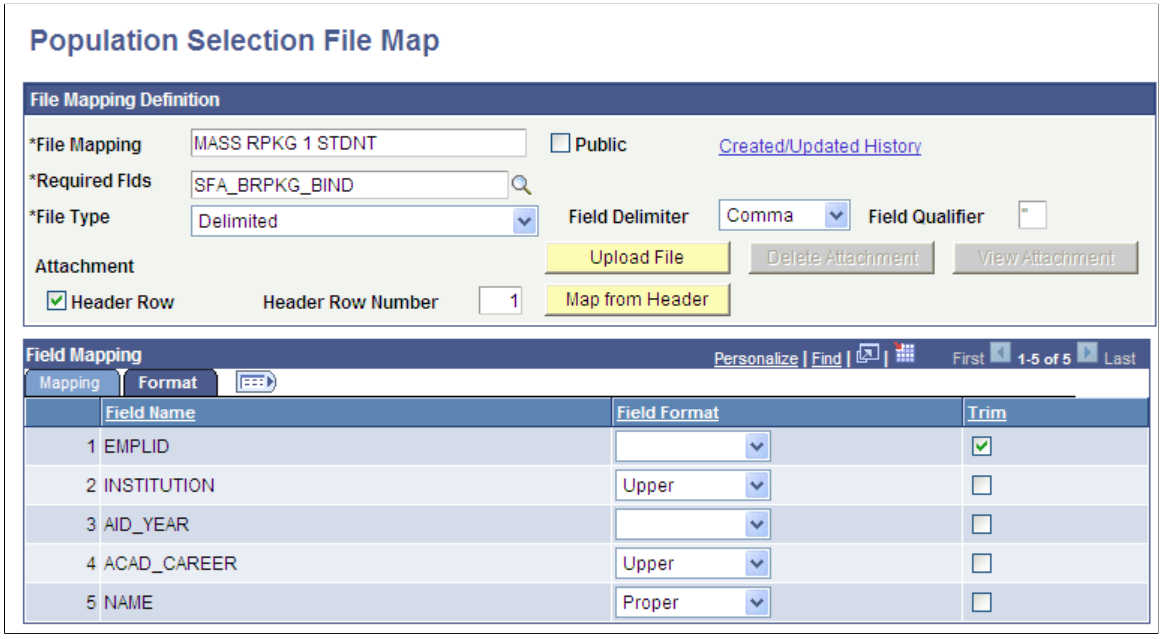


Image: Population Selection File Map - Format tab

This example illustrates the fields and controls on the Population Selection File Map - Format tab. You can find definitions for the fields and controls later on this page.



Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

If you access the Population Selection File Map page from the Population Selection group box, the system enters the required fields record from Population Selection Context for the application process, and displays the list of fields in the required fields record in the Field Mapping group box. If you manually navigated to the page, you must enter the name of the required fields records in the Required Flds field. When you exit the field, the system displays the fields from that record.

File Mapping Definition

File Mapping Enter a name for this mapping.

Public Select to make this mapping accessible for all users to use and update.

Clear to make this mapping accessible only to the user who created it.

A public mapping is useful for example, when a mapping can be reused for different files or for files that are refreshed regularly.

Created/Updated History Click to view a history of who updated the mapping and when.

Required Flds (required fields) Enter the name of the record that contains the fields required to identify the population.

Note: If you access the page directly from a run control page for a process, the Required Flds field does not appear. The bind record containing the required fields is derived from the Population Context for the process, and the field is hidden.

File Type Specify the type of external file, either *Flat* or *Delimited.*

If you enter *Delimited,* you must also verify or enter the delimiter and qualifier.

File types are translate values. Do not modify these values.

Field Delimiter Appears only for a file type of *Delimited.*

Enter the value in the external file that signifies the end of one value and the beginning of another. Values are:

Comma (default value)

Pipe

Semicolon

Tab

Field Delimiter values are translate values. Do not modify these values.

Field Qualifier Appears only for a file type of *Delimited.*

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Enter the value in the external file that qualifies the end of one field and the beginning of another. The default qualifier is ″ (quotation mark).

Header Row and **Header Row Number**

The Header Row check box and the Header Row Number field appear only if the file type is *Delimited.*

If the file that contains the IDs to process includes header rows, for example column titles or to display a title for the document, select the Header Row check box and enter the number of rows in the uploaded file used as the header. The Population Selection process evaluates values after the end of the number of header rows specified.

If you are unsure if the uploaded file has header rows or unsure of which row to use as the header, click Cancel to return to the run control page. Then, click View File to view the delimited file and validate and count the header rows if they exist. Return to the mapping and continue.

Map From Header

Appears only when Header Row is selected.

Click to map the column names in the uploaded spreadsheet to the fields required to run the process. The system reads the name of each column in the spreadsheet, compares it to the fields listed in the Field Mapping group box, and if it finds a match, enters the column number in theField Number field for that field. For example, assume that *CAREER* is the title of the second column in the spreadsheet that you uploaded. If the *ACADEMIC_CAREER* field appears in the list of fields in the File Mapping group box, then the system sets the Field Number field to *2.*

The Mapping From Header feature does not map columns in the spreadsheet that do not match any fields in the File Mapping group box. To map these, you must manually enter the column number in the Field Number column.

Note: The system uses the short name of the fields to map to the columns in the spreadsheet because PS Query uses the short name as the default value to name the column names when the query results are downloaded into Microsoft Excel.

See *PeopleTools: Query*, “Downloading Queries.”

Upload File, Delete Attachment, and **View Attachment**

These buttons appear only if you navigate to the Population Selection File Map page from the menu, you select the Header Row check box, and if the file option for the External File selection tool on the Selection Tool setup page is set to *Attachment.*

The buttons do not appear if you access the page from a run control page because the buttons are available from the standard Population Selection group box on the run control page.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Click to upload, delete, or view the external file to use.

Note: The process uses the uploaded file for automatic field mapping only. No other process uses this file.

File Path and Name

This field appears only if you navigate to the Population Selection File Map page from the menu, and if the file option for the External File selection tool on the Selection Tool setup page is set to *Physical Path.*

The field does not appear if you access the page from a run control page because the field is available from the standard Population Selection group box on the run control page.

Enter the path to the external file to use. The file path must follow the appropriate syntax for the application server that you are using:

- For an NT application server, an appropriate syntax might be: \\tl-abc \test_data\My_File_Name.csv.
- For a UNIX application server, an appropriate syntax might be: //test_data/ My_File_Name.csv. (File names are case sensitive for UNIX.)
- For a mixed environment (for example, an NT application server with a Unix Process Scheduler) consider setting the External File selection tool to *Attachments* rather than to *Physical Path.*

Warning! Consider these conditions when uploading an external file to parse for use with the Population Selection process:

---

1. The uploaded file must include at least the fields marked as required for the process that you want to run.
2. When mapping a flat file, you must include blank spaces when a value is shorter than the longest character length for the column.

   Looking at the columns below, the starting position for the ID column is 1 and the length is 10. Ten is the longest value for the ID column. All other values need to have blank spaces (represented by x in the example) to make them a length of 10. The same applies to the Name column. Mary-Anne Fletcher has the longest name, which is 18 characters long. Therefore the other name values need to end with blank spaces to make them a character length of 18. In this case the mapping Start position is 11 and the field length is 18.

   ```
   ID NAME AD1000xxxx John Smithxxxxxxx AD100xxxxx
    Mary-Ann Fletcher AD10xxxxxx Joe Pittxxxxxxxxxx
    ADxxxxxxxx Mark Smithxxxxxxxx
   ```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

3. Not adding the blank spaces will result in a preview error that says: *File row [#] contains fewer fields than file layout. (14015,577) The file row could not be parsed.*

   In the case of a delimited file, this error appears when the parser process looks for a field defined on the file mapping that is not in the file.

4. Always make changes in the original spreadsheet, not in the converted .csv file.

   The Population Selection process considers all leading zeros. If you click the Preview Selection Results link on the run control component, the leading zeros are there. However, when you save a Microsoft Excel spreadsheet into a .csv file and reopen the file either by using Microsoft Excel or by clicking the View File button in the standard Population Selection group box, values with leading zeros are truncated. If you make changes to the .csv file, save it, and re-upload it, you will save a file without the leading zeros and the values will be invalid.

5. Due to a known Microsoft error (SYLK error) as of the date of this publication, you cannot view a file if its text begins with ID. You can process and preview the file, but you cannot view it from the run control page when its text begins with ID.

For information about "SYLK: File format is not valid" error, see Microsoft web site.

Field Mapping - Mapping Tab

Field Name and **Required**

Displays the name of each field in the required fields record that is specified in the Population Selection Context, and indicates *Yes* if the uploaded file must have the same field for the process to run. Required fields also require a Field Number to map the field to the corresponding column in the uploaded spreadsheet.

Fields marked *No* are optional. They do not need to be in the spreadsheet and do not required a Field Number value unless you choose to use them, in which case you can manually enter the corresponding column number in the Field Number field.

File Column Header

Appears only when File Type is *Delimited.* and Header Rowis selected.

When the Header Row check box is selected, the system enters the short name of the field in the File Column Header column.

Field Number

Appears only when File Type is *Delimited.*

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

	Displays the number of the column in the uploaded file that corresponds to this field.
Starting Position and **Field Length**	Appear only when File Type is *Flat.* Enter the starting position and the field length for each required field.

Field Mapping - Format Tab

Field Format	Appears if formatting options exist. Enter the format to use for the field value. Each field has options relevant to the data type. For example, possible date type formats are *DDMMMYY, DDMMYYYY, DDMMYY,* and so on. Possible character type formats include the case controls *Lower, Proper,* and *Upper.*
Date Separator	Appears only for date field types. Enter the symbol to use to separate the date parts.
Decimal Position	Appears only for number field types. Enter the position to use for placing the decimal.
Trim	Appears only for fields that are not date or time field types. Select to trim leading or trailing spaces in fields.

Warning! After entering mapping information, preview the results before running the File Parser process to confirm that the column titles match the information inside the columns and that the mapping and results are otherwise correct. To preview the results, access the standard Population Selection group box in the run control component of the application process. If the Population Selection check box is available, select it. Then set the Selection Tool to *External File,* upload the file, select the mapping name you just created, and click the Preview Selection Results link. The Preview Selection Results link appears only if your institution set the external file selection tool to allow preview results at installation.

Cross-references to Application-specific File Parser Content

This section discusses the cross-references to application-specific documents that contain content related to File Parser.

Financial Aid

Topics:

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Understanding the File Parser Process

PeopleSoft Financial Aid incorporates the File Parser utility automatically in its External Awards process to convert data from an external award data file into external award staging tables. The process uses a predefined context definition delivered by the PeopleSoft system. You do not have to define the field conversion or set up the context definition. However, you will need to create file mapping definition using the External Award context definition to populate the target external award staging tables.

See:

- "Setting Up External Award Processing" (PeopleSoft Campus Solutions 9.2: Financial Aid)
- "Understanding External Award Processing" (PeopleSoft Campus Solutions 9.2: Financial Aid)

Setting Up a Context Definition

The PeopleSoft system currently delivers a predefined External Award Load context definition for the Financial Aid External Awarding process. The context definition identifies the fields in the External Award staging table that can accept converted external file data. You can modify the context definition to prevent fields from being visible (Visible for Mapping check box), but you should not make any other modifications to the predefined context definition. Changing the record or field names or their values or making any other modifications can cause the External Award process to fail.

See "Setting Up External Award Processing" (PeopleSoft Campus Solutions 9.2: Financial Aid).

Running the File Parser Process

The External Awards process in PeopleSoft Financial Aid automatically incorporates the File Parser utility to convert data from an external award data file into the External Award staging tables. Do not run the File Parser process separately, as described in this document, for the External Awards process. Instead, after you have created your specific file mapping definition for your external award file, use the External Award Staging process to load external award data into the staging tables.

See "Understanding External Award Processing" (PeopleSoft Campus Solutions 9.2: Financial Aid).

Recruiting and Admissions

See "Setup for Loading Applications Using File Parser" (PeopleSoft Campus Solutions 9.2: Recruiting and Admissions).

Student Records

The Higher Education Statistics Agency (HESA) functionality uses File Parser to import HESA Validation Kit errors.

See "Understanding HESA Returns" (PeopleSoft Campus Solutions 9.2: Student Records).

Campus Community

Constituent Transaction Management (CTM) uses File Parser for offline transactions.

See Understanding CTM.

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Creating Calculated Mapping Application Class Objects

This section discusses how to:

- Create an application package.
- Create an application class.
- Access staged field values.
- Access file field values.

It is assumed that the person who will be responsible for creating and managing the application classes is a seasoned engineer who is comfortable with working with PeopleTools Application Designer, is well versed in writing PeopleCode, and has detailed knowledge of the referential integrity associated with the tables referenced by File Parser.

Warning! The examples presented in this section have not been exposed to any rigorous testing scenarios. If you decide to use these examples in any form, it is strongly recommended that a complete and thorough set of test cycles be run before moving to production.

Creating an Application Package

To create an application package:

1. Open PeopleTools Application Designer.
2. File/New/Application Package.
3. Save the empty application package with a meaningful name.
4. At the root level of the application package, insert a new application package. Repeat as necessary to produce a meaningful application package structure.

Creating an Application Class

To create an application class:

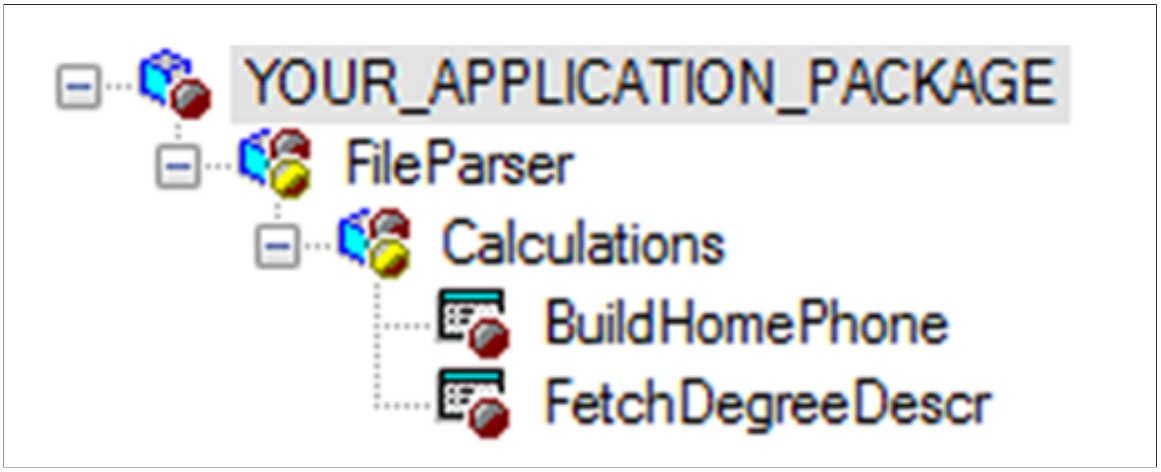
1. At the appropriate level within your application package, insert an application class.
2. Save it with a meaningful name, reflective of what the application class is doing and is intuitive for the end user to select when defining the File Parser Mapping Definitions.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

The following shows an example:

Image: Application package example

This example illustrates the fields and controls on the Application package example. You can find definitions for the fields and controls later on this page.



Accessing Staged Field Values

The following application class code illustrates how to reference data elements that reside at the *Staging* level.

Note: In the following code, only the code formatted in bold can be changed by the developer. Rest of the code is required to successfully run or reference staging field values.

```
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:MODEL:Results:ResultsRecord;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
import SCC_FILE_PARSER:UTIL:Exception:FileParserException;
/**
* class FetchDegreeDescr
*
* @version 1.0
* @author Your Institution
*
* Module: Your Application
* Description: Your Description
*/
/**
* This Application Class is designed to read the incoming Degree value
* residing on the mapped/staged Record Definition: SCC_STG_EXTDEGR, Field: DEGREE,
* and then obtain the description from the table PS_DEGREE_TBL.
*
* If found, pass back the description, else pass back the string 'NOT FOUND'.
*/
class FetchDegreeDescr extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
/* public methods */
method FetchDegreeDescr();
method calculateValue(&ResultsFieldIn As SCC_FILE_PARSER:MODEL:Results:ResultsField⇒
,
&ResultsCollectionIn As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns an⇒
y;
private
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
method GetFileData();
method GetFieldReferences();
instance string &Degree;
instance string &DegreeDescrOut;
instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
instance SCC_FILE_PARSER:MODEL:Results:ResultsField &DegreeField;
end-class;
/**
* Instantiate the App. Class.
*/
method FetchDegreeDescr
%Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();
end-method;
/**
* This method is the main driver that passes back the derived value.
*/
method calculateValue
/+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
/+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
/+ Returns Any +/
/+ Extends/implements SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue ⇒
+/
&ResultsCollection = &ResultsCollectionIn;
&ResultsField = &ResultsFieldIn;
%This.GetFileData();
/**
* Pass back the derived value to the field calling this Calculated Application Clas⇒
s.
*/
Return &DegreeDescrOut;
end-method;
/**
* This method uses the obtained degree value as the key in the criteria to obtain t⇒
he
* description from the table PS_DEGREE_TBL.
*/
method GetFileData
Local string &DegreeDescr;
/**
* Call the method GetFieldReferences to obtain the incoming degree value.
*/
If &DegreeField = Null Then
%This.GetFieldReferences();
End-If;
&Degree = "";
/**
* Test to see if we found a degree value. If so, execute the SQL statement to
* obtain the description.
*/
If &DegreeField <> Null Then
&Degree = &DegreeField.FieldValue;
SQLExec("SELECT A.DESCR FROM PS_DEGREE_TBL A WHERE A.DEGREE =:1 AND A.EFF_STATUS =
'A' AND A.EFFDT = (SELECT MAX(A1.EFFDT) FROM PS_DEGREE_TBL A1 WHERE A1.DEGREE =
A.DEGREE AND A1.EFFDT <= %DateIn(:2))", &Degree, %Date, &DegreeDescr);
/**
* Test to see if we found a degree description. If so, assign the value to be
* passed back.
* Otherwise, assign the text string 'NOT FOUND'.
*/
If &DegreeDescr = "" Then
&DegreeDescrOut = "NOT FOUND";
Else
&DegreeDescrOut = &DegreeDescr;
End-If;
End-If;
end-method;
method GetFieldReferences
Local integer &SegmentNbr;
Local integer &RecordRow;
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
Local SCC_FILE_PARSER:MODEL:Results:ResultsRecord &obj_SCC_STG_EXTDEGR;
&SegmentNbr = &ResultsField.ResultsRecord.SegmentNbr;
&RecordRow = &ResultsField.ResultsRecord.RecordRow;
/**
* Establish a reference to the staging record definition
* containing the incoming mapped field.
*/
&obj_SCC_STG_EXTDEGR = &ResultsCollection.GetResultsRecord(&SegmentNbr,
"SCC_STG_EXTDEGR", &RecordRow);
/**
* If the reference exists, establish a reference to the staging mapped field.
*/
If &obj_SCC_STG_EXTDEGR <> Null Then
&DegreeField = &obj_SCC_STG_EXTDEGR.GetResultsField("DEGREE");
End-If;
end-method;
```

Accessing File Field Values

To reference an incoming File Field value, you need to obtain two key values associated with that specific field. These key values are *Segment Number* and the *Sort Order Number*.

To obtain the required keys, follow these steps:

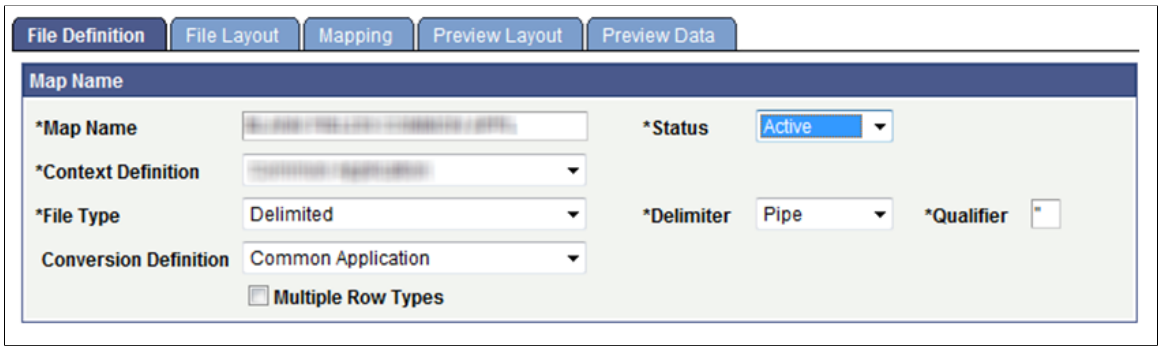
Step 1: Identify the File Fields you wish to access.

In this example, we want to access the incoming Home Phone Area Code and the Home Phone.

Step 2: Identify the Segment Number. This value is derived based upon the File Definition. If the File Definition does not contain Multiple Row Types (that is, you have not selected the Multiple Row Type check box), then the Segment Number will always be a value of 1 (one).

Image: File Definition page - Multiple Row Types is not selected

This example illustrates the fields and controls on the File Definition page - Multiple Row Types is not selected. You can find definitions for the fields and controls later on this page.

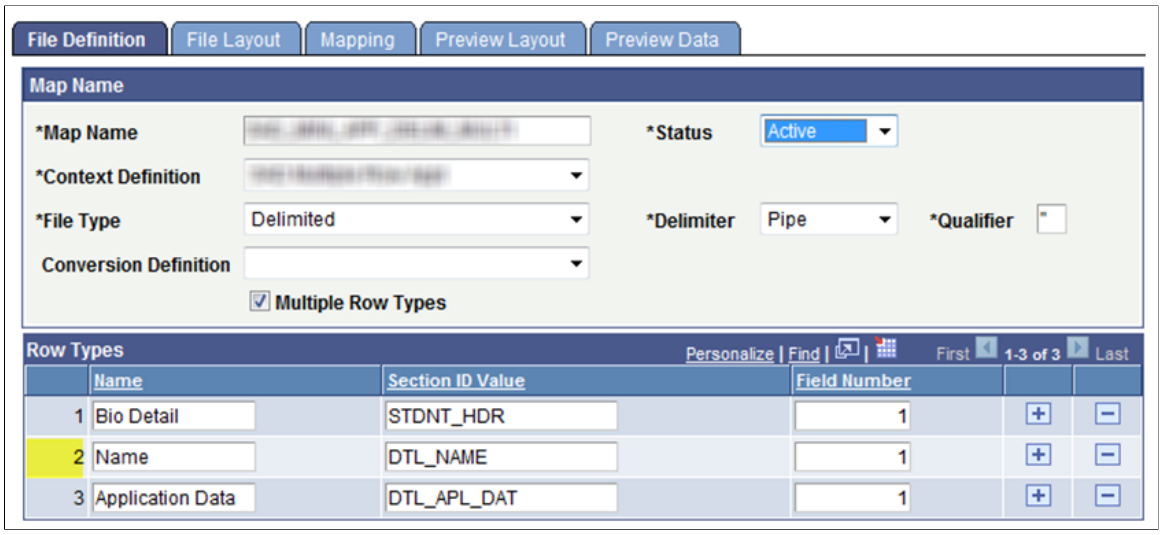


Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

In this example, a Segment Number of 2 (two) would be used if fields contained within DTL_NAME are to be referenced:

Image: File Definition page - Multiple Row Types is selected

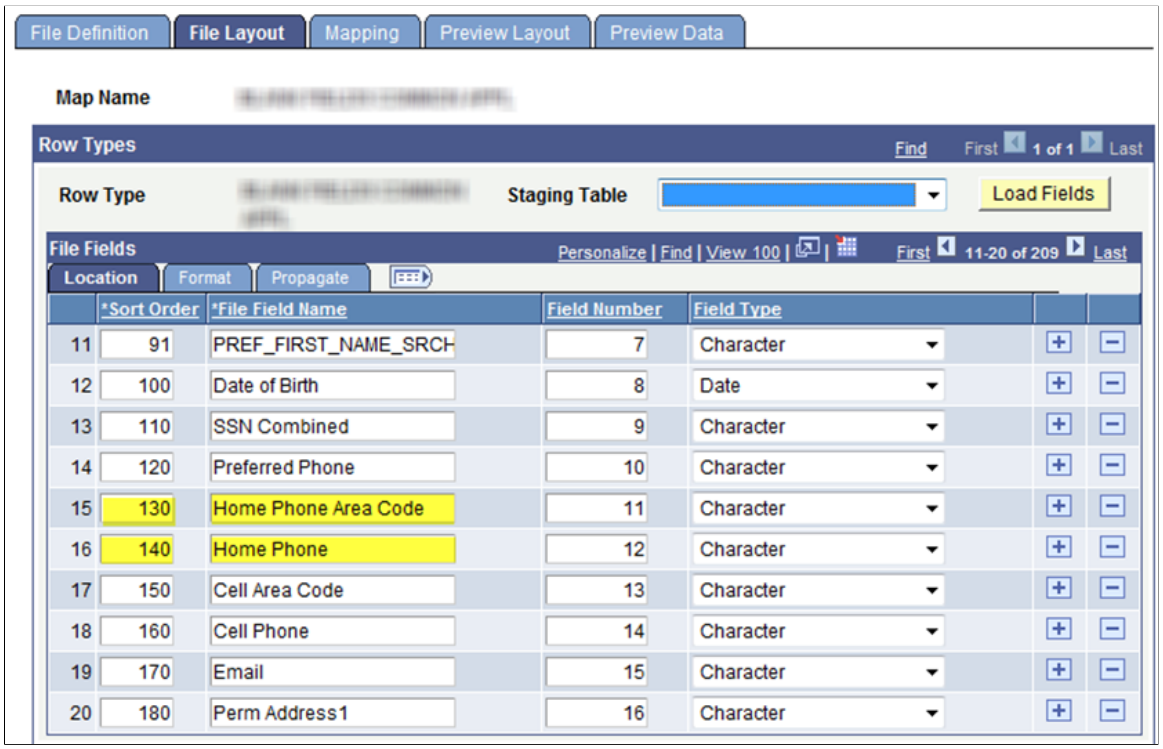
This example illustrates the fields and controls on the File Definition page - Multiple Row Types is selected. You can find definitions for the fields and controls later on this page.



Step 3: Obtain the Sort Order Number(s). The Sort Order Number can be found on the File Layout page:

Image: File Layout - Location tab

This example illustrates the fields and controls on the File Layout - Location tab. You can find definitions for the fields and controls later on this page.



 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

In the graphic we see that Home Phone Area Code has a Sort Order Number of 130, and the Home Phone Sort Order Number is 140.

Step 4: Write these values down to be referenced within your application class.

The following application class code illustrates how to reference data elements that reside at the incoming File Field level.

Note: In the following code, only the code formatted in bold can be changed by the developer. Rest of the code is required to successfully run or reference staging field values.

```
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsFileFields;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
/**
* class BuildHomePhone
*
* @version 1.0
* @author Your Institution
*
* Module: Your Application
* Description: Your Description
*/
/**
* This Application Class is designed to read two values, Home Phone Area Code
* and Home Phone from the incoming File Fields. They will then be concatenated
* into one single field and passed back to the mapped field containing the
* Calculated mapping action/reference to this Application Class.
*/
class BuildHomePhone extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
/* public methods */
method BuildHomePhone();
method calculateValue(&ResultsFieldIn As SCC_FILE_PARSER:MODEL:Results:ResultsField⇒
,
&ResultsCollectionIn As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns an⇒
y;
private
instance string &FileFldValueOut;
instance string &FieldValueString01, &FieldValueString02;
instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
instance SCC_FILE_PARSER:MODEL:Results:ResultsFileFields &FileFldValueField;
end-class;
/**
* Instantiate the App. Class.
*/
method BuildHomePhone
%Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();
end-method;
/**
* This method is the main driver that passes back the derived value.
*/
method calculateValue
/+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
/+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
/+ Returns Any +/
/+ Extends/implements SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue ⇒
+/
Local integer &filesegout, &fldnumout;
&FileFldValueField = &ResultsCollectionIn.RFF;
/**
* Assign the appropriate value for the Segment Number to be passed to
* the Method FetchFileFieldValue.
*/
&filesegout = 1; /* File Segment Number will always be 1 for non-multiple row types⇒
. */
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
/**
* Assign the appropriate value for the File Field value to be passed to
* the Method FetchFileFieldValue.
*/
/* Home Phone Area Code */
&fldnumout = 130;
&FieldValueString01 = &FileFldValueField.FetchFileFieldValue(&filesegout, &fldnumou⇒
t);
/* Home Phone */
&fldnumout = 140;
&FieldValueString02 = &FileFldValueField.FetchFileFieldValue(&filesegout, &fldnumou⇒
t);
/**
* Test that we have values to build and pass back.
*/
If (&FieldValueString01 <> " " And
&FieldValueString02 <> " ") Then
/* Build the home phone number */
&FileFldValueOut = &FieldValueString01 | " " | &FieldValueString02;
Else
&FileFldValueOut = " ";
End-If;
Return &FileFldValueOut;
end-method;
```

Commonly Used Calculated Mapping Application Class Objects

This section provides examples for how to derive:

- External Organization ID.
- Activity Description.
- Degree Description.
- Subject Description.
- Write Transaction Map.
- Remove special characters for Last Name Search.
- Remove special characters for First Name Search.

It is assumed that the person who will be responsible for creating and managing the application classes is a seasoned engineer who is comfortable with working with PeopleTools Application Designer, is well versed in writing PeopleCode, and has detailed knowledge of the referential integrity associated with the tables referenced by the File Parser context definition.

Warning! These are examples only, and have not been exposed to any rigorous testing scenarios. Should you decide to use these examples in any form, it is strongly recommended that a complete and thorough set of test cycles be exercised before moving to production.

External Organization ID

Code example:

```
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
import SCC_FILE_PARSER:MODEL:Results:ResultsRecord;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
import SCC_FILE_PARSER:UTIL:Exception:FileParserException;

/**
  *  class ExtOrgId
  *
  *  @version 1.0
  *  @author Campus Solutions
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */


class ExtOrgId extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method ExtOrgId();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField,
&ResultsCollectionIn As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns an⇒
y;

private
   method GetFileData();
   method GetFieldReferences();

   instance string &SchoolCeeb;
   instance string &ExtOrgIdOut;

   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &SchoolCeebField;

end-class;


method ExtOrgId

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/


   &ResultsCollection = &ResultsCollectionIn;
   &ResultsField = &ResultsFieldIn;

   %This.GetFileData();

   Return &ExtOrgIdOut;

end-method;


method GetFileData

   Local string &ExtOrg;

   If &SchoolCeebField = Null Then
      %This.GetFieldReferences();
   End-If;
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
   &SchoolCeeb = "";

   If &SchoolCeebField <> Null Then
      &SchoolCeeb = &SchoolCeebField.FieldValue;

      SQLExec("SELECT EXT_ORG_ID FROM PS_ORG_ADM_CUR_VW WHERE ATP_CD = :1",
&SchoolCeeb, &ExtOrg);

      &ExtOrgIdOut = &ExtOrg;
   End-If;

end-method;


method GetFieldReferences

   Local integer &SegmentNbr;
   Local integer &RecordRow;

   Local SCC_FILE_PARSER:MODEL:Results:ResultsRecord &obj_SCC_STG_ACADHST;

   &SegmentNbr = &ResultsField.ResultsRecord.SegmentNbr;
   &RecordRow = &ResultsField.ResultsRecord.RecordRow;

   &obj_SCC_STG_ACADHST = &ResultsCollection.GetResultsRecord(&SegmentNbr,
"SCC_STG_ACADHST", &RecordRow);

   If &obj_SCC_STG_ACADHST <> Null Then
      &SchoolCeebField = &obj_SCC_STG_ACADHST.GetResultsField("SAD_SCHOOL_CEEB");
   End-If;

end-method;
```

Activity Description

Code example:

```
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:MODEL:Results:ResultsRecord;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
import SCC_FILE_PARSER:UTIL:Exception:FileParserException;

/**
  *  class FetchActivityDescr
  *
  *  @version 1.0
  *  @author Campus Solutions
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */


class FetchActivityDescr extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method FetchActivityDescr();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField, &ResultsCollectionIn
As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns any;

private
   method GetFileData();
   method GetFieldReferences();

   instance string &Setid;
   instance string &Activity;
   instance string &ActivityDescrOut;
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ActivityField;

end-class;


method FetchActivityDescr

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/


   &ResultsCollection = &ResultsCollectionIn;
   &ResultsField = &ResultsFieldIn;

   %This.GetFileData();

   Return &ActivityDescrOut;

end-method;


method GetFileData

   Local string &ActivityDescr;

   If &ActivityField = Null Then
      %This.GetFieldReferences();
   End-If;

   &Setid = "PSUNV"; /* For now, the SetID is
hard coded - need to add code to fetch from system. */
   &Activity = "";

   If &ActivityField <> Null Then
      &Activity = &ActivityField.FieldValue;

      SQLExec("select a.descr from ps_extr_actvty_tbl a
where a.extra_activity =:1 and a.setid =:2 and
a.eff_status = 'A' and a.effdt = (select max(a1.effdt)
from ps_extr_actvty_tbl a1 where a1.setid = a.setid and
a1.extra_activity = a.extra_activity and
a1.effdt <= %DateIn(:3))", &Activity,
&Setid, %Date, &ActivityDescr);

      If &ActivityDescr = "" Then
         &ActivityDescrOut = "NOT FOUND";
      Else
         &ActivityDescrOut = &ActivityDescr;
      End-If;
   End-If;

end-method;


method GetFieldReferences

   Local integer &SegmentNbr;
   Local integer &RecordRow;
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
   Local SCC_FILE_PARSER:MODEL:Results:ResultsRecord &obj_SCC_STG_EXTRACU;

   &SegmentNbr = &ResultsField.ResultsRecord.SegmentNbr;
   &RecordRow = &ResultsField.ResultsRecord.RecordRow;

   &obj_SCC_STG_EXTRACU = &ResultsCollection.GetResultsRecord(&SegmentNbr,
"SCC_STG_EXTRACU", &RecordRow);

   If &obj_SCC_STG_EXTRACU <> Null Then
      &ActivityField = &obj_SCC_STG_EXTRACU.GetResultsField("EXTRA_ACTIVITY");
   End-If;

end-method;
```

Degree Description

Code example:

```
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:MODEL:Results:ResultsRecord;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
import SCC_FILE_PARSER:UTIL:Exception:FileParserException;

/**
  *  class FetchDegreeDescr
  *
  *  @version 1.0
  *  @author Campus Solutions
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */


class FetchDegreeDescr extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method FetchDegreeDescr();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField, &ResultsCollectionIn
As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns any;

private
   method GetFileData();
   method GetFieldReferences();

   instance string &Degree;
   instance string &DegreeDescrOut;

   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &DegreeField;

end-class;


method FetchDegreeDescr

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/


   &ResultsCollection = &ResultsCollectionIn;
   &ResultsField = &ResultsFieldIn;

   %This.GetFileData();

   Return &DegreeDescrOut;

end-method;


method GetFileData

   Local string &DegreeDescr;

   If &DegreeField = Null Then
      %This.GetFieldReferences();
   End-If;

   &Degree = "";

   If &DegreeField <> Null Then
      &Degree = &DegreeField.FieldValue;

      SQLExec("SELECT A.DESCR FROM PS_DEGREE_TBL A
WHERE A.DEGREE =:1 AND A.EFF_STATUS = 'A' AND
A.EFFDT = (SELECT MAX(A1.EFFDT) FROM PS_DEGREE_TBL A1
WHERE A1.DEGREE = A.DEGREE AND
A1.EFFDT <= %DateIn(:2))", &Degree, %Date, &DegreeDescr);

      If &DegreeDescr = "" Then
         &DegreeDescrOut = "NOT FOUND";
      Else
         &DegreeDescrOut = &DegreeDescr;
      End-If;
   End-If;

end-method;


method GetFieldReferences

   Local integer &SegmentNbr;
   Local integer &RecordRow;

   Local SCC_FILE_PARSER:MODEL:Results:ResultsRecord &obj_SCC_STG_EXTDEGR;

   &SegmentNbr = &ResultsField.ResultsRecord.SegmentNbr;
   &RecordRow = &ResultsField.ResultsRecord.RecordRow;

   &obj_SCC_STG_EXTDEGR = &ResultsCollection.GetResultsRecord(&SegmentNbr,
"SCC_STG_EXTDEGR", &RecordRow);

   If &obj_SCC_STG_EXTDEGR <> Null Then
      &DegreeField = &obj_SCC_STG_EXTDEGR.GetResultsField("DEGREE");
   End-If;

end-method;
```

Subject Description

Code example:

```
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:MODEL:Results:ResultsRecord;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
import SCC_FILE_PARSER:UTIL:Exception:FileParserException;

/**
  *  class FetchSubjectDescr
  *
  *  @version 1.0
  *  @author Campus Solutions
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */


class FetchSubjectDescr extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method FetchSubjectDescr();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField, &ResultsCollectionIn
As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns any;

private
   method GetFileData();
   method GetFieldReferences();

   instance string &SubjectArea;
   instance string &SubjectAreaDescrOut;

   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &SubjectAreaField;

end-class;


method FetchSubjectDescr

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/


   &ResultsCollection = &ResultsCollectionIn;
   &ResultsField = &ResultsFieldIn;

   %This.GetFileData();

   Return &SubjectAreaDescrOut;

end-method;


method GetFileData

   Local string &SubjectAreaDescr;

   If &SubjectAreaField = Null Then
      %This.GetFieldReferences();
   End-If;

   &SubjectArea = "";

   If &SubjectAreaField <> Null Then
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
      &SubjectArea = &SubjectAreaField.FieldValue;

      SQLExec("SELECT A.DESCR FROM PS_EXT_SUBJECT_TBL A
WHERE A.EXT_SUBJECT_AREA =:1 AND A.EFF_STATUS = 'A'
AND A.EFFDT = (SELECT MAX(A1.EFFDT) FROM PS_EXT_SUBJECT_TBL A1
WHERE A1.EXT_SUBJECT_AREA = A.EXT_SUBJECT_AREA AND
A1.EFFDT <= %DateIn(:2))", &SubjectArea, %Date, &SubjectAreaDescr);

      If &SubjectAreaDescr = "" Then
         &SubjectAreaDescrOut = "NOT FOUND";
      Else
         &SubjectAreaDescrOut = &SubjectAreaDescr;
      End-If;
   End-If;

end-method;


method GetFieldReferences

   Local integer &SegmentNbr;
   Local integer &RecordRow;

   Local SCC_FILE_PARSER:MODEL:Results:ResultsRecord &obj_SCC_STG_ADMINT;

   &SegmentNbr = &ResultsField.ResultsRecord.SegmentNbr;
   &RecordRow = &ResultsField.ResultsRecord.RecordRow;

   &obj_SCC_STG_ADMINT = &ResultsCollection.GetResultsRecord(&SegmentNbr,
"SCC_STG_ADMINT", &RecordRow);

   If &obj_SCC_STG_ADMINT <> Null Then
      &SubjectAreaField = &obj_SCC_STG_ADMINT.GetResultsField("EXT_SUBJECT_AREA");
   End-If;

end-method;
```

Write Transaction Map

Code example:

```
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:MODEL:Results:ResultsRecord;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;
import SCC_FILE_PARSER:UTIL:Exception:FileParserException;

/**
  *  class WriteTranMap
  *
  *  @version 1.0
  *  @author Campus Solutions
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */


class WriteTranMap extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method WriteTranMap();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField, &ResultsCollectionIn
As SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns any;

private
   method GetFileData();
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
   instance string &WriteTranMapOut;

   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &SccTempIdField;

end-class;


method WriteTranMap

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/


   &ResultsCollection = &ResultsCollectionIn;
   &ResultsField = &ResultsFieldIn;

   %This.GetFileData();

   Return &WriteTranMapOut;

end-method;


method GetFileData

   Local string &TransAcCd;
   Local integer &SccTempIdIn;

   &TransAcCd = "AAWS_FP_ADMAPPLS";

   SQLExec("SELECT SCC_TEMP_ID FROM PS_SCC_STG_CTRS", &SccTempIdIn);

   SQLExec("INSERT INTO PS_SCC_SL_TRNMAP (SCC_TEMP_ID,SCC_TRANSAC_CD)
VALUES (:1,:2)", &SccTempIdIn, &TransAcCd);

   &WriteTranMapOut = " ";

end-method;
```

Remove Special Characters for Last Name Search

Code example:

```
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsFileFields;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;

/**
  *  class LastNameSrch
  *
  *  @version 1.0
  *  @author Campus Solutions
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
class LastNameSrch extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method LastNameSrch();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField, &ResultsCollectionIn As
SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns any;

private

   instance string &LastNameIn;

   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsFileFields &FileFldValueField;

end-class;

Declare Function RemoveAccent PeopleCode FUNCLIB_CS.PREFERRED_NAME FieldFormula;


method LastNameSrch

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/

   Local integer &filesegout, &fldnumout;
   Local string &LastNameTemp, &LastNameSrch;

   &FileFldValueField = &ResultsCollectionIn.RFF;

   &filesegout = 1; /* File Segment Number will always be 1 for
non-multiple row types. */

   /* Last Name */
   &fldnumout = 60;
   &LastNameIn = &FileFldValueField.FetchFileFieldValue(&filesegout, &fldnumout);

   RemoveAccent(&LastNameIn, &LastNameTemp);
   &LastNameSrch = &LastNameTemp;

   Return &LastNameSrch;

end-method;
```

Remove Special Characters for First Name Search

Code example:

```
import SCC_FILE_PARSER:MODEL:Results:ResultsField;
import SCC_FILE_PARSER:UTIL:FieldCalculationAbstract;
import SCC_FILE_PARSER:MODEL:Results:ResultsFileFields;
import SCC_FILE_PARSER:MODEL:Results:ResultsCollection;

/**
  *  class FirstNameSrch
  *
  *  @version 1.0
  *  @author Campus Solutions
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
  *
  *  Module: Recruiting and Admissions / Common App.
  *  Description:
  */


class FirstNameSrch extends SCC_FILE_PARSER:UTIL:FieldCalculationAbstract
   /* public methods */

   method FirstNameSrch();
   method calculateValue(&ResultsFieldIn As
SCC_FILE_PARSER:MODEL:Results:ResultsField, &ResultsCollectionIn As
SCC_FILE_PARSER:MODEL:Results:ResultsCollection) Returns any;

private

   instance string &FirstNameIn;

   instance SCC_FILE_PARSER:MODEL:Results:ResultsField &ResultsField;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsCollection &ResultsCollection;
   instance SCC_FILE_PARSER:MODEL:Results:ResultsFileFields &FileFldValueField;

end-class;

Declare Function RemoveAccent PeopleCode FUNCLIB_CS.PREFERRED_NAME FieldFormula;


method FirstNameSrch

   %Super = create SCC_FILE_PARSER:UTIL:FieldCalculationAbstract();

end-method;


method calculateValue
   /+ &ResultsFieldIn as SCC_FILE_PARSER:MODEL:Results:ResultsField, +/
   /+ &ResultsCollectionIn as SCC_FILE_PARSER:MODEL:Results:ResultsCollection +/
   /+ Returns Any +/
   /+ Extends/implements
SCC_FILE_PARSER:UTIL:FieldCalculationAbstract.CalculateValue +/

   Local integer &filesegout, &fldnumout;
   Local string &FirstNameTemp, &FirstNameSrch;

   &FileFldValueField = &ResultsCollectionIn.RFF;

   &filesegout = 1; /* File Segment Number will always be 1 for
non-multiple row types. */

   /* First Name */
   &fldnumout = 40;
   &FirstNameIn = &FileFldValueField.FetchFileFieldValue(&filesegout, &fldnumout);

   RemoveAccent(&FirstNameIn, &FirstNameTemp);
   &FirstNameSrch = &FirstNameTemp;

   Return &FirstNameSrch;

end-method;
```

Troubleshooting Tips

If setting up a large file, map out everything on paper first to determine which records to populate before embarking on setting up the Context Definition and File Mapping Definition.

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Context Definitions

Tips:

- Do not deselect the Visible for Mapping check box on important fields when initially setting up and testing because it can be very confusing when trying to debug issues. It is easier to see important data being imported on File Mapping when troubleshooting. Once you have the mapping definition tested, you can go back and deselect Visible for Mapping on the field you do not want the users to change.
- Use the default sort order which increments by 10. This will allow you to insert additional staging records in the future.
- Ensure the staging tables hierarchy via the sort order. Parent table(s) will have a lower sort order than their children.

File Mapping Definition

Tips:

- If you start setting up in *Pending* status, make sure that you change the status to *Active* and save along the way to resolve setup issues as you proceed. Errors are not fired when the status is set to *Pending*.
- Use an incremental approach. Set up the required records, test, add some more records, and test. This is an easier way to troubleshoot setup issues.
- File Layout: When inserting File Fields, use the default sort order which increments by 10. This will allow you to insert additional fields later.
- Mapping page:
  - The system inserts the records according to the order listed on the Context Definition.
  - A value must be provided for each required field. The system displays an error message if a required field has a Mapping Action set to *None*.
  - If you are having issues using the Sequence Mapping Action, try setting the Mapping Action to *Default* and entering 1, 2, 3, and so on for each row of the record you are inserting to.

Porting definitions from one environment to another

Tips:

- When creating new File Parser definitions, it is important to understand that File Parser has a unique key structure for each of its definitions (that is, Conversion, Context, Pop Select, and Mapping).
- After creating these definitions in your development environment, query your current Production File Parser definitions to make sure that you will not overwrite them when porting your new definitions into your production environment.
- If you find any collision(s), rename the keys via native SQL to a unique value(s) prior to exporting the definitions.

The following are some examples of useful SQL and DataMover scripts to help you manage the export or import of File Parser definitions:

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

Note: Do not use any ID value between 1 and 999. This range is reserved for Campus Solutions.

List Current File Parser Definitions:

```
/*
Instructions:

1) Identify scope. If you are porting either Conversion definitions and/or
   Pop Select Mapping Definitions, you will need to uncomment the respective SQL
   statements - See below.

2) Supply a WHERE clause to narrow down the results.
*/

/* -- - - - - - - - - Conversion Definitions (Optional)  - - - - - - - - - - */
/*
SELECT * FROM PS_SCCFP_CNVR;
SELECT * FROM PS_SCCFP_CNVR_FLD;
SELECT * FROM PS_SCCFP_CNVR_VAL;
*/

/* - - - - - - - - Pop Select Mapping Definitions (Optional) - - - - - - - - -*/
/*
SELECT * FROM PS_SCCFP_PS_MAP;
SELECT * FROM PS_SCCFP_PS_MAPFLD;
*/

/* - - - - - - - - - - - - - Context Definitions - - - - - - - -- - - - - - - */

SELECT * FROM PS_SCCFP_CNTXT_DFN;
SELECT * FROM PS_SCCFP_CNTXT_REC;
SELECT * FROM PS_SCCFP_CNTXT_RF;

/* - - - - - - - - - - - - - Mapping Definitions - - - - - - - - - - - - - - - */

SELECT * FROM PS_SCCFP_MAP_DFN;
SELECT * FROM PS_SCCFP_MAP_REC;
SELECT * FROM PS_SCCFP_MAP_FLD;
SELECT * FROM PS_SCCFP_FILE_SGT;
SELECT * FROM PS_SCCFP_FILE_FLD;
```

Update File Parser IDs (Keys):

```
/*
Instructions:

1) Identify scope. If you are porting either Conversion definitions and/or
   Pop Select Mapping Definitions, you will need to uncomment the respective SQL
   statements.

2) Identify the next available ID value(s).

3) Edit the SQL as follows:

   a) Conversion Definitions - (If applicable) Find/Replace the
      string 'New_CNVR_ID'with the next available value assigned
      to your application.

      Find/Replace the string 'Current_CNVR_ID' with
      the current value of SCCFP_CNVR_ID.

   b) Pop Select Mapping Definitions - (If applicable) Find/Replace the
      string 'New_MAP_ID' with the next available value
      assigned to your application.

      Find/Replace the string 'Current_MAP_ID' with the current
      value of SCCFP_MAP_ID.

   c) Context Definitions - Find/Replace the string 'New_CNTXT_ID' with
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
      the next available value assigned to your application.

      Find/Replace the string 'Current_CNTXT_ID' with the current
      value of SCCFP_CNTXT_ID.

   d) Mapping Definitions - Find/Replace the string 'New_MAP_ID' with
      the next available value assigned to your application.

      Find/Replace the string 'Current_MAP_ID' with the current
      value of SCCFP_MAP_ID.

4) Save your changes and execute the SQL statements in the native
   SQL tool of your choice.

5) Repeat steps 1-4 for any additional File Parser definitions.

6) Verify/Unit test the changes within the File Parser component.
*/


/* - - - - - - - - - - Conversion Definitions (Optional) - -  - - - - - */
/*
UPDATE PS_SCCFP_CNVR
SET SCCFP_CNVR_ID = New_CNVR_ID
WHERE SCCFP_CNVR_ID = Current_CNVR_ID;

UPDATE PS_SCCFP_CNVR_FLD
SET SCCFP_CNVR_ID = New_CNVR_ID
WHERE SCCFP_CNVR_ID = Current_CNVR_ID;

UPDATE PS_SCCFP_CNVR_VAL
SET SCCFP_CNVR_ID = New_CNVR_ID
WHERE SCCFP_CNVR_ID = Current_CNVR_ID;
*/


/* - - - - - - -- Pop Select Mapping Definitions (Optional) - -  - - - - - -*/
/*
UPDATE PS_SCCFP_PS_MAP
SET SCCFP_MAP_ID = New_MAP_ID
WHERE SCCFP_MAP_ID = Current_MAP_ID;

UPDATE PS_SCCFP_PS_MAPFLD
SET SCCFP_MAP_ID = New_MAP_ID
WHERE SCCFP_MAP_ID = Current_MAP_ID;
*/


/* - - - - - - - - -- - - Context Definitions - - - - - - - - - - - - */

UPDATE PS_SCCFP_CNTXT_DFN
SET SCCFP_CNTXT_ID = New_CNTXT_ID
/*    ,SCCFP_CNVR_ID = New_CNVR_ID */
WHERE SCCFP_CNTXT_ID = Current_CNTXT_ID;

UPDATE PS_SCCFP_CNTXT_REC
SET SCCFP_CNTXT_ID = New_CNTXT_ID
WHERE SCCFP_CNTXT_ID = Current_CNTXT_ID;

UPDATE PS_SCCFP_CNTXT_RF
SET SCCFP_CNTXT_ID = New_CNTXT_ID
WHERE SCCFP_CNTXT_ID = Current_CNTXT_ID;


/* - - - - - - - - - - - - - - Mapping Definitions - - - - - - - - - - - - */

UPDATE PS_SCCFP_MAP_DFN
SET SCCFP_MAP_ID = New_MAP_ID,
    SCCFP_CNTXT_ID = New_CNTXT_ID
/*    ,SCCFP_CNVR_ID = New_CNVR_ID */
WHERE SCCFP_MAP_ID = Current_MAP_ID;
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
UPDATE PS_SCCFP_MAP_REC
SET SCCFP_MAP_ID = New_MAP_ID
WHERE SCCFP_MAP_ID = Current_MAP_ID;

UPDATE PS_SCCFP_MAP_FLD
SET SCCFP_MAP_ID = New_MAP_ID
WHERE SCCFP_MAP_ID = Current_MAP_ID;

UPDATE PS_SCCFP_FILE_SGT
SET SCCFP_MAP_ID = New_MAP_ID
WHERE SCCFP_MAP_ID = Current_MAP_ID;

UPDATE PS_SCCFP_FILE_FLD
SET SCCFP_MAP_ID = New_MAP_ID
WHERE SCCFP_MAP_ID = Current_MAP_ID;
```

Export File Parser Definitions Using DataMover:

```
-- Use this script to extract the File Parser definitions into a data (.dat) format⇒

-- to be used to import into another database.
--
-- Step 1) Save this script with a meaningful name; something reflective
-- of the File Parser definitions being extracted.
--
-- Step 2) Change the string 'xxxx' to a meaningful name for the export
-- log. Add/prefix a path if you do not wish to use the default path
-- defined in Configuration Manager.
--
-- Step 3) Change the string 'yyyy' to a meaningful name for the data
-- file. Add/prefix a path if you do not wish to use the default path
-- defined in Configuration Manager.
--
-- Step 4) If you are extracting any conversion definitions, change the
-- string 'aaaa' to the value of the conversion ID.
--              Else, delete the EXPORT statements containing the string 'aaaa'.   ⇒

--
-- Step 5) Change the string 'bbbb' to the context ID.
--
-- Step 6) If you are extracting any Pop Select definitions, change the
-- string 'cccc' to the value of the map ID.
--              Else, delete the EXPORT statements containing the string 'cccc'.   ⇒

--
-- Step 7) Change the string 'dddd' to the value of the map ID.
--
-- Step 8) Save your changes and run the DataMover script against the
-- database containing the File Parser definitions you wish to extract.

SET LOG xxxx_Export.LOG;
SET OUTPUT yyyy_EXPORT.DAT;

EXPORT SCCFP_CNVR WHERE SCCFP_CNVR_ID = aaaa;
EXPORT SCCFP_CNVR_FLD WHERE SCCFP_CNVR_ID = aaaa;
EXPORT SCCFP_CNVR_VAL WHERE SCCFP_CNVR_ID = aaaa;

EXPORT SCCFP_CNTXT_DFN WHERE SCCFP_CNTXT_ID = bbbb;
EXPORT SCCFP_CNTXT_REC WHERE SCCFP_CNTXT_ID = bbbb;
EXPORT SCCFP_CNTXT_RF WHERE SCCFP_CNTXT_ID = bbbb;

EXPORT SCCFP_PS_MAP WHERE SCCFP_MAP_ID = cccc;
EXPORT SCCFP_PS_MAPFLD WHERE SCCFP_MAP_ID = cccc;

EXPORT SCCFP_MAP_DFN WHERE SCCFP_MAP_ID = dddd;
EXPORT SCCFP_MAP_REC WHERE SCCFP_MAP_ID = dddd;
EXPORT SCCFP_MAP_FLD WHERE SCCFP_MAP_ID = dddd;
EXPORT SCCFP_FILE_SGT WHERE SCCFP_MAP_ID = dddd;
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
EXPORT SCCFP_FILE_FLD WHERE SCCFP_MAP_ID = dddd;
```

Import File Parser Definitions Using DataMover:

```
-- Use this script to import the File Parser definition data (.dat) file
-- into another database.
--
-- Step 1) Save this script with a meaningful name; something reflective of
-- the File Parser definitions  being imported.
--
-- Step 2) Change the string 'xxxx' to a meaningful name for the import
-- log. Add/prefix a path if you do not wish to use the default path defined
-- in Configuration Manager.
--
-- Step 3) Change the string 'yyyy' to the name of the .dat file being
-- imported. This is the name of the file exported
-- by FILE_PARSER_DEFN_EXPORT.dms. Add/prefix a path if you do not
-- wish to use the default path defined in Configuration Manager.
--
-- Step 4) If you are importing any conversion definitions, change the
-- string 'aaaa' to the value of the conversion ID.
--              Else, delete the DELETE statements containing the string 'aaaa'.
--              Delete the IMPORT statements containing the Record Definition
--              beginning with 'SCCFP_CNVR'.
--
-- Step 5) Change the string 'bbbb' to the context ID.
--
-- Step 6) If you are importing any Pop Select definitions, change the
-- string 'cccc' to the value of the map ID.
--              Else, delete the DELETE statements containing the string 'cccc'.
--              Delete the IMPORT statements containing the Record Definition
--              beginning with 'SCCFP_PS_MAP'.
--
-- Step 7) Change the string 'dddd' to the value of the map ID.
--
-- Step 8) Save your changes and run the DataMover script against the database
-- containing the File Parser definitions you wish to import.
--

SET LOG xxxx_Import.LOG;
SET INPUT yyyy_EXPORT.DAT;

DELETE FROM PS_SCCFP_CNVR WHERE SCCFP_CNVR_ID = aaaa;
DELETE FROM PS_SCCFP_CNVR_FLD WHERE SCCFP_CNVR_ID = aaaa;
DELETE FROM PS_SCCFP_CNVR_VAL WHERE SCCFP_CNVR_ID = aaaa;

DELETE FROM PS_SCCFP_CNTXT_DFN WHERE SCCFP_CNTXT_ID = bbbb;
DELETE FROM PS_SCCFP_CNTXT_REC WHERE SCCFP_CNTXT_ID = bbbb;
DELETE FROM PS_SCCFP_CNTXT_RF WHERE SCCFP_CNTXT_ID = bbbb;

DELETE FROM PS_SCCFP_PS_MAP WHERE SCCFP_MAP_ID = cccc;
DELETE FROM PS_SCCFP_PS_MAPFLD WHERE SCCFP_MAP_ID = cccc;

DELETE FROM PS_SCCFP_MAP_DFN WHERE SCCFP_MAP_ID = dddd;
DELETE FROM PS_SCCFP_MAP_REC WHERE SCCFP_MAP_ID = dddd;
DELETE FROM PS_SCCFP_MAP_FLD WHERE SCCFP_MAP_ID = dddd;
DELETE FROM PS_SCCFP_FILE_SGT WHERE SCCFP_MAP_ID = dddd;
DELETE FROM PS_SCCFP_FILE_FLD WHERE SCCFP_MAP_ID = dddd;

COMMIT;

IMPORT SCCFP_CNVR;
IMPORT SCCFP_CNVR_FLD;
IMPORT SCCFP_CNVR_VAL;

IMPORT SCCFP_CNTXT_DFN;
IMPORT SCCFP_CNTXT_REC;
IMPORT SCCFP_CNTXT_RF;

IMPORT SCCFP_PS_MAP;
```

Copyright © 2017, Oracle and/or its affiliates. All rights reserved.

```
IMPORT SCCFP_PS_MAPFLD;

IMPORT SCCFP_MAP_DFN;
IMPORT SCCFP_MAP_REC;
IMPORT SCCFP_MAP_FLD;
IMPORT SCCFP_FILE_SGT;
IMPORT SCCFP_FILE_FLD;
```

Bundle 45. Documentation update only.

Update File Parser Definitions to ensure Max Content, Mapping and Field Value Counter values are set correctly.

```
-- Step 1)Select the one row of data, three field values prior to making the update
SELECT * FROM PS_SCCFP_COUNTER

-- Step 2)Update the counter record with the max context, mapping, and field value ⇒
sequences
UPDATE PS_SCCFP_COUNTER
SET SCCFP_CNVR_ID = (SELECT MAX(SCCFP_CNVR_ID) FROM PS_SCCFP_CNVR)
,SCCFP_CNTXT_ID = (SELECT MAX(SCCFP_CNTXT_ID) FROM PS_SCCFP_CNTXT_DFN)
,SCCFP_MAP_ID = (SELECT MAX(SCCFP_MAP_ID) FROM PS_SCCFP_MAP_DFN)
```

 Copyright © 2017, Oracle and/or its affiliates. All rights reserved.