STE Data Loading Process for Contractors – v2 (Mar. 31, 2011, CPC)

# Submission of new data in a Contractor Package

The TEIS Environment Contractor Package contains the following folders:

* Documentation
* Maps
* Operational\_Data.gdb
* Scratch
* Scripts
* Tools\_and\_Templates.gdb

Of these, only the contents of the **Operational\_Data.gdb** file geodatabase will be modified and submitted to MoE by the contractor. This is a file geodatabase that will contain the new (or updated) ecosystem mapping project data, formatted to MoE standards. The other five folders contain tools, reference tables and documentation to be used in processing the new data into the format matching that of the **Operational\_Data.gdb** geodatabase.

The following is a detailed description of the recommended methods for data processing. The text assumes that the contractor has a working knowledge of ESRI’s ArcGIS Desktop applications, tools and data formats (version 9.3.1 and up).

# Stage 1 - Convert data to single FGDB

1. If the new data is kept in file geodatabase (FGDB) format, make a copy of that at the same folder level as the six folders listed above, and rename it **Stage\_1** (which will create a file folder named **Stage1.gdb**). (The data will be processed in a series of Stages, which are titled **Stage\_1** through **Stage\_5** for consistency within the scripting processes.) If the new data is not already in FGDB format, create a new empty file geodatabase named **Stage\_1**. If the new data is in coverage or shapefile format, set the tolerance of each one to 0.001 (in ArcCatalog, right-click the layer, select Properties, and select the Tolerance tab), and export each coverage or shapefile to the new **Stage\_1** FGDB by using the basic conversion tools in ArcGIS Desktop. If the attributes for a data layer are kept separately in a table (such as an Excel spreadsheet), export each of these to new tables in the **Stage\_1** FGDB as well.
2. Start a “Loading Log”, a simple text file or document with notes about each operation that is performed, and the results. See the Loading Log Files section at the end of this document.
3. Ensure that a BAPID number has been assigned to each project dataset. Keep track of which BAPID number belongs to which dataset by including the BAPID number in the names of the layers and tables in the FGDB or folder that contains the data.
4. In each project dataset, if there is no polygon ID tag field present, such as ECP\_TAG or TER\_TAG (not POLY\_NUM or similar with a simple integer value), then such a field should be created now (using the field name PROJPOLYID, type Text, length 50) and populated with unique values. Use ArcMap’s Calculate Field function to populate the new field as a concatenation of the BAPID number and POLY\_NUM (if it exists and is populated uniquely) or OBJECTID, separated by a hyphen or underscore character. Example: enter CStr([BAPID]) & “\_“ & CStr([POLY\_NUM]) (a VBScript expression) or str(!BAPID!) + “\_“ + str(!POLY\_NUM!) (a Python expression) in the Calculate Field form.
5. Each project data layer should have its attribute table permanently joined. If the attributes for a project dataset are kept in a separate table from the polygon layer, use ArcMap to create a join between the layer’s attribute table and the full attribute table, and then export the layer as a new layer. If the project dataset has no polygon layer, only tabular attributes, these records can still be imported into the TEI\_Long\_Tbl with “null geometry” in Stage 2.
6. Browse through the fields in each project dataset and note anything that does not have a correlation to any field in the TEI\_Long\_Tbl template. If there are any such fields, then load this data into the TEI\_Usr\_Dfn\_Data table template. Right-click the empty TEI\_Usr\_Dfn\_Data table and select Load, then select Load Data. Load your attribute table into the template. Note that the template only contains three fields (TEIS\_ID, BAPID and PROJPOLYID). Ensure that the unique identifying attribute in your attribute table is mapped to the PROJPOLYID field in the TEI\_Usr\_Dfn\_Data template. Next, use the Join Field tool (in the ArcToolbox, Data Management toolbox, Joins toolset) to attach your user-defined fields to the TEI\_Usr\_Dfn\_Data table. Finally, add information about your user-defined fields to the TEI\_Usr\_Dfn\_Fields template (see the TEI Submission Standards document for details).
7. The following 3-character text fields for texture and surface expression are no longer in the new TEI\_Long\_Tbl (they have been retired):
   * TTEX\_1, TTEX\_2, TTEX\_3
   * STTEX\_1, STTEX\_2, STTEX\_3
   * TTTEX\_1, TTTEX\_2, TTTEX\_3
   * SURF\_E1, SURF\_E2, SURF\_E3
   * SSURF\_E1, SSURF\_E2, SSURF\_E3
   * TSURF\_E1, TSURF\_E2, TSURF\_E3

Each one of these eighteen fields has been replaced with three single-character text fields (TTEX\_1C, TTEX\_1B, TTEX\_1A and so on – a total of 54 new fields). If any of the 3-character fields exists and is populated in the source data, it needs to be “parsed out” before the table can be loaded to the new TEI\_Long\_Tbl template. As well, the geomorphic process subclass fields (such as GEOP\_SCM1) need to be parsed out to the appropriate single character fields, although the original 3-character field still exists in the TEI\_Long\_Tbl.

The best way to begin this parsing process is to add the 54 new single-character fields to the original table for the feature class to be loaded. This can be done by running the following commands in the ArcMap Python window:

import arcgisscripting  
gp = arcgisscripting.create(9.3) ## (these two commands are only needed once per ArcMap or ArcCatalog session – they should work in both Arc 9.3.1 and Arc 10)

newfieldlist = ['TTEX\_1C','TTEX\_1B','TTEX\_1A','TTEX\_2C','TTEX\_2B','TTEX\_2A','TTEX\_3C','TTEX\_3B','TTEX\_3A','STTEX\_1C','STTEX\_1B','STTEX\_1A','STTEX\_2C','STTEX\_2B','STTEX\_2A','STTEX\_3C','STTEX\_3B','STTEX\_3A','TTTEX\_1C','TTTEX\_1B','TTTEX\_1A','TTTEX\_2C','TTTEX\_2B','TTTEX\_2A','TTTEX\_3C','TTTEX\_3B','TTTEX\_3A','SURF\_E1A','SURF\_E1B','SURF\_E1C','SURF\_E2A','SURF\_E2B','SURF\_E2C','SURF\_E3A','SURF\_E3B','SURF\_E3C','SSURF\_E1A','SSURF\_E1B','SSURF\_E1C','SSURF\_E2A','SSURF\_E2B','SSURF\_E2C','SSURF\_E3A','SSURF\_E3B','SSURF\_E3C','TSURF\_E1A','TSURF\_E1B','TSURF\_E1C','TSURF\_E2A','TSURF\_E2B','TSURF\_E2C','TSURF\_E3A','TSURF\_E3B','TSURF\_E3C']  
for newfield in newfieldlist:   
 gp.AddField\_management("data\_layer\_name",newfield,"TEXT","","",1)

Then populate these new fields as in the following examples.

**TEXTURAL FIELDS**

Texture codes are parsed in reverse order where the final character represents the dominant characteristic and therefore and belongs in the “A” field. For this reason the texture ABC fields are listed in reverse order in the TEI\_Long\_Tbl; for example, TTEX\_1C, TTEX\_1B, TTEX\_1A.

Examples:  
TTEX\_1 = dsm: TTEX\_1C = d TTEX\_1B = s TTEX\_1A = m  
TTEX\_1 = xs: TTEX\_1B = x TTEX\_1A = s  
TTEX\_1 = a: TTEX\_1A= a

To achieve this, you can run commands in the ArcMap Python window such as the following:

gp.CalculateField\_management("*long\_tbl\_name*", "TTEX\_1A", "!TTEX\_1![-1:]", "PYTHON")

gp.CalculateField\_management("*long\_tbl\_name*", "TTEX\_1B", "!TTEX\_1![-2:-1]", "PYTHON")

gp.CalculateField\_management("*long\_tbl\_name*", "TTEX\_1C", "!TTEX\_1![-3:-2]", "PYTHON")

**SURFACE EXPRESSION FIELDS**

SURF\_E{1-3} parsed to SURF\_E{1C, 1B, 1A, 2C, 2B, 2A, 3C, 3B, 3A} and the same for SSURF\_E{1-3} and TSURF\_E{1-3}

Characters are parsed in order into the A, B and C fields.

Examples:  
SURF\_E2 = cfk: SURF\_E2A = c SURF\_E2B = f SURF\_E2C = k  
SURF\_E2 = vb: SURF\_E2A = v SURF\_E2B = b  
SURF\_E3 = p: SURF\_E3A = p

To achieve this, you can run commands in the ArcMap Python window such as the following:

gp.CalculateField\_management("*long\_tbl\_name*", "SURF\_E1A", "!SURF\_E1![0:1]", "PYTHON")

gp.CalculateField\_management("*long\_tbl\_name*", "SURF\_E1B", "!SURF\_E1![1:2]", "PYTHON")

gp.CalculateField\_management("*long\_tbl\_name*", "SURF\_E1C", "!SURF\_E1![2:3]", "PYTHON")

**GEOMORPHIC SUBCLASS FIELDS**

GEOP\_SCM{1-3} to GEOP\_INZ{1-3}\*\* and GEOP\_SCM{1A, 1B, 1C-3A, 3B, 3C}

\*\* only when the first character is “ (double prime). The double prime character is copied the GEOP\_INZ field and the remainder of the characters are parsed in order into GEOP\_SCM1A, B and C. (Note that the original 3-character fields GEOP\_SCM1, GEOP\_SCM2 and GEOP\_SCM3 still exist in the TEI\_Long\_Tbl, along with the “parsed out” fields.)

Examples:  
GEOP\_SCM1 = “aw: GEOP\_INZ1 = “ GEOP\_SCM1A = a GEOP\_SCM1B = w  
GEOP\_SCM2 = sdb: GEOP\_SCM2A = s GEOP\_SCM2B = d GEOP\_SCM3C = d  
GEOP\_SCM3 = pt: GEOP\_SCM2A = p GEOP\_SCM2B = t  
GEOP\_SCM2 = u: GEOP\_SCM2A = u

Parsing out the GEOP\_SCM fields should be done on two separate selection sets: one set where GEOP\_SCM value does not begin with a double prime character, and the other set where GEOP\_SCM does begin with a double prime character. To select the first set, start ArcMap and load the layer, then open the layer’s table and Select by Attributes. Use the following WHERE clause for the selection:

"GEOP\_SCM1" IS NOT NULL AND "GEOP\_SCM1" <> '' AND "GEOP\_SCM1" <> ' ' AND "GEOP\_SCM1" NOT LIKE '"%'

If any records are selected, the following commands can then be run in the ArcMap Python window:

gp.CalculateField\_management(layer\_name, "GEOP\_SCM1A", "Mid([GEOP\_SCM1],1,1)", "VB")

gp.CalculateField\_management(layer\_name, "GEOP\_SCM1B", "Mid([GEOP\_SCM1],2,1)", "VB")

gp.CalculateField\_management(layer\_name, "GEOP\_SCM1C", "Mid([GEOP\_SCM1],3,1)", "VB")

(Note: it is preferable here to use the VB “Mid” substring expression to parse out the values rather than the Python “slice notation”, because the latter will result in an error anytime the GEOP\_SCM1 field contains a double-quote character.)

Next, we Select by Attributes to find records where GEOP\_SCM1 has a non-null value that does not begin with a double-quote. Use the following WHERE clause for the selection:

"GEOP\_SCM1" LIKE '"%'

Then use the following Python commands in ArcMap to parse out the values for the selected records:

gp.CalculateField\_management(layer\_name, "GEOP\_INZ1", "Mid([GEOP\_SCM1],1,1)", "VB")

gp.CalculateField\_management(layer\_name, "GEOP\_SCM1A", "Mid([GEOP\_SCM1],2,1)", "VB")

gp.CalculateField\_management(layer\_name, "GEOP\_SCM1B", "Mid([GEOP\_SCM1],3,1)", "VB")

gp.CalculateField\_management(layer\_name, "GEOP\_SCM1C", "Mid([GEOP\_SCM1],4,1)", "VB")

Note that the fourth command is redundant because the GEOP\_SCM1 field can only contain 3 or fewer characters. (This is an issue that may be addressed by changing the GEOP\_SCM fields to four characters in future versions of the TEIS Environment.)

Repeat all of the above to parse out GEOP\_SCM2 and GEOP\_SCM3.

# Stage 2 – Standardize Attributes (By Data Set)

1. Within the TEIS Environment contractor package, in the Operational\_Data FGDB, there should be two empty feature classes:
   1. TEI\_Long\_Tbl (a polygon feature class)
   2. TEI\_Project\_Details (a table)
2. Create a new FGDB called Stage 2 and copy those two templates here. Make as many copies of the TEI\_Long\_Tbl feature class as the number of project datasets you will be submitting, and name each copy “TEI\_Long\_Tbl\_*BAPID*”. There need only be a single copy of the TEI\_Project\_Details table.
3. Check to see if the feature classes have the correct fields relative to the templates with the following processes:
   1. Start the CompareFieldsNoAlias tool in the TEI\_Toolbox. Enter the original dataset and the empty template in the dialog box that follows.
   2. If any fields are an incorrect type, create a new field of the correct type in the attribute table of the project dataset and use the Field Calculator in ArcMap to populate the new field.
4. All project datasets must be in the Albers projection to match the BC Government map data standard. For any datasets that are not in Albers, run the Project tool (in ArcToolbox under Data Management Tools, Projections and Transformations, Feature). Use the “NAD\_1983\_BC\_Environment\_Albers” projection definition file, found in the Scripts folder of the TEIS Environment contractor package.
5. Load each project feature class from Stage 1 into the corresponding empty template in Stage 2 by right-clicking on the empty template in ArcCatalog and selecting ‘Load’, then ‘Load Data…’. In the dialog box that follows, browse for and select the project dataset, and click the “Add” button. In the next stage of the wizard, scroll through to make sure each field of the TEI\_Long\_Tbl is linked to the correct field from the project dataset. Address the following items individually:
6. PROJPOLYID – This is a new item that must be created from the contents of the project identifier field (ECP\_TAG or TER\_TAG). If those tag fields are empty or don’t exist, then use the PROJPOLYID field, which you created and populated in step 3 of Stage 1.
7. TEIS\_ID will be unspecified (blank) for now.
8. Run the StandardizeNoData tool in the TEI\_Toolbox for each of the TEI\_Long\_Tbl feature classes that you have just created and populated. This will ensure that all text fields that contain no data set to “” (empty string) rather than <Null>, and that all numeric fields where 0 is an invalid value (such as TDEC\_1) are set to <Null> rather than 0.
9. Verify that all polygons have been loaded (input record count = export record count) and record it in the loading log (the text file that you created in step 2 of Stage 1, remember?).
10. Verify that all fields that should be populated are populated (e.g. SEI fields, texture fields, etc.)
11. The TEI\_Project\_Details table contains the assessment information about the BAPID. For example: the project name, the completion date, the organization name, etc. These fields are populated manually from the project boundary feature class or from using EcoCat reports and other sources. In Project Details table, make sure that there is exactly one record per project and that the following fields for each record are populated:
    1. BAPID – and double-check that the BAPID is unique in the TEI\_Project\_Details table. If not, assign a new BAPID. (Also, check BAPID\_TRACKER)
    2. PROJ\_TYPE – and that the PROJ\_TYPE is in the Domain\_Project\_Types table. If the PROJ\_TYPE is TER, recalculate it to TIM, and if it is TERSOI recalculate to TIMSOI. This field has also been extended to 9 characters from 6.
    3. PROJ\_SCALE
    4. PROJ\_STAT – should be set to “Boundary Only”, “Complete”, or another status based on the data set being loaded. See the Tools\_and\_Templates.gdb\Domain\_Project\_Status for allowable codes.
    5. Boudary\_Only\_Flag (BDRY\_ONLY) – should be set to “Y” if project boundary is boundary only.
12. Ensure that the following fields in each TEI\_Long\_Tbl are populated with the same values as the corresponding record in the Project Details table: BAPID, FCODE, PROJ\_TYPE, PROJ\_SCALE.
13. Copy any User\_Defined\_Fields tables from the Stage\_1 FGDB into Stage\_2.

# Stage 3 – Combine Projects

1. Make a copy of the Operational\_Data FGDB (containing only an empty TEI\_Long\_Tbl feature class and an empty TEI\_Project\_Details table), and rename the copied FGDB “Stage\_3”.
2. Append all of the Long Table feature classes from Stage\_2 into the empty TEI\_Long\_Tbl feature class template in Stage\_3, so there will be one amalgamated TEI\_Long\_Tbl. This can be achieved using the ArcGIS Append tool (in the Toolbox under Data Management, General). Copy the Project Details table from Stage\_2 into Stage\_3 as well. Also, copy any User\_Defined\_Fields tables from Stage\_2 into Stage\_3.
3. Populate the TEIS\_ID field in the new Stage\_3 TEI\_Long\_Tbl by running the SetTEIS\_ID tool in the TEI\_Toolbox.
4. Run each of the error-checking tools (ValidateDomains, ValidateGeometry, ValidateMandatoryFields, ValidateRows, followed by CreateErrorSummary, in that order) in the TEI\_Toolbox. For each tool, specify the TEI\_Long\_Tbl layer in Stage\_3 as the input.
5. After all four of these tools have been run, there should be five new tables in the Stage\_3 FGDB:

* TEI\_Long\_Tbl\_DomainErrors
* TEI\_Long\_Tbl\_GeometryErrors
* TEI\_Long\_Tbl\_MandatoryFieldErrors
* TEI\_Long\_Tbl\_RangeErrors
* TEI\_Long\_Tbl\_RowErrors
* TEI\_Long\_Tbl\_SummaryErrors

Scan through each of these new tables to get an idea of what kinds of errors are present in the data.

1. Copy any User\_Defined\_Fields tables from the Stage\_2 FGDB into Stage\_3.

# Stage 4 – Fix Geometry and Attribute Issues

1. Copy the Stage\_3 FGDB to a new FGDB and name it “Stage\_4”.
2. Fix any standard/systemic issues:
   1. Run the Repair Geometry tool (in the ArcToolbox under Data Management, Features) if needed. If the data contains valid features that have no geometry, be sure to specify that the tool should not delete features with null geometry.
3. Use the StandardizeNoData tool in the TEI\_Toolbox (if needed and not run previously) to fix systemic issues, such as the TDEC and SDEC fields being populated with zerosView each of the error tables listed in step 23 of Stage 3, and edit the TEI\_Long\_Tbl’s attribute table in ArcMap to eliminate as many errors as possible. Note that each error table has a field called FEATURE\_ID that can be “joined” to the TEI\_Long\_Tbl’s TEIS\_ID field. Every edit should be noted in the Loading Log file, and it may also be a good idea to add a short comment in the POLY\_COM field for the record(s) you edited, describing the error and what was done to fix it. After each round of error fixes, re-run all of the error-checking tools as in step 22), check the new error tables and make more fixes as needed.
4. For your own checking purposes, you can use the CreateProjectBoundaries tool in the TEI\_Toolbox (using the TEI\_Long\_Tbl as input). This will produce a new polygon feature class called TEI\_Project\_Boundaries, which you can view to ensure that the project boundaries appear as expected. This boundary layer will not be included in the data being submitted.
5. Similary, you can optionally run the CreateShortTable tool in the TEI\_Toolbox, using the TEI\_Long\_Tbl as input. This will create a new polygon feature class called TEI\_Short\_Tbl. It should have the same number of records as the long table, but the attribute table will have far fewer fields with values from the long table concatenated together into longer “map label” strings, which may be easier to read and to spot errors in.

# Stage 5 – Load to Submission File Geodatabase

1. Copy the Stage\_4 FGDB and name it Stage\_5. Delete all layers and tables except for the TEI\_Long\_Tbl layer, the latest version of the five error tables (as listed in step 23), the TEI\_Project\_Details table, and any User\_Defined\_Fields tables.
2. In ArcCatalog, view the contents of the Operational\_Data FGDB, which should still contain only empty templates of the TEI\_Long\_Tbl layer and the TEI\_Project\_Details table. Using the same Load Data procedure as in step 12 of Stage 2, right-click the TEI\_Long\_Tbl layer and load into it the data from TEI\_Long\_Tbl in the Stage\_5 FGDB. Do the same for the TEI\_Project\_Details table. Finally, copy the five error tables and any User\_Defined\_Fields tables from the Stage\_5 FGDB into the Operational\_Data FGDB.

# Submitting the data

At this stage, the Operational\_Data FGDB should include:

* **TEI\_Long\_Tbl** (polygon feature class)
* **TEI\_Long\_Tbl\_DomainErrors** (table)
* **TEI\_Long\_Tbl\_GeometryErrors** (table)
* **TEI\_Long\_Tbl\_RangeErrors** (table)
* **TEI\_Long\_Tbl\_RowErrors** (table)
* **TEI\_Long\_Tbl\_SummaryErrors** (table)
* **TEI\_Project\_Details** (table)
* **TEI\_Usr\_Dfn\_Data** (table) (only if created in step 6, Stage 1)
* **TEI\_Usr\_Dfn\_Fields** (table) (only if created in step 6, Stage 1)

If all of the above are present and populated, then the FGDB is ready for submission to MoE for inclusion into the production database. (Note that the error tables should be included even if they include zero records.) The Loading Log text file, separate from the FGDB, should also be submitted. More details of the data transfer procedure will be described here as a new web-based tool develops.

# Loading Log Files

In order to keep track of the data loading process for each project dataset, an associated log file is created. This log file is a simple text document that records every procedure, edit, tool execution, issue or comment that was added or performed so that any change made to the data can be traced back to its origin. It is a document without any standardized format, usually containing the date, file locations, name of the feature classes, tools run, and descriptions of and reasoning behind any edits made. In case there are any issues that arise in the future, these logfiles will be the reference point. Here is an example of a project log file:

------------------------------------------

2010-01-11

- Copied the data from the BAPID cleanup (SR2)

- Created Stage\_1 from that. (141285 features)

- 141285 features

- Moved data to the latest template as Stage 2

- Selected all PROJ\_TYPE = 'TER' and set to TIM.

- There are 30 records without a project type. These are generally sliver polygons, and a hand full of errors.

- Set all PROJ\_TYPE to 'TIM'

- Calculated all PROJ\_SCALE = 50000

- PROJPOLYID set to TER\_TAG

- Filtered the Project\_Details by project type (took away the SOILs) entries.

QC Round 1:

- 141285 Features.

- Created Stage\_3 from Stage\_2 (new template)

\* Moved local to GISSERV1 for speed

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\ValidateDomains.py D:\Data\Processing\SR4\CPC\Stage\_3.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- Codes B and F are in SURFM\_Q#s in many cases, also with: S and 1.

- Codes 1,2,3,4 in GEOP\_1 in some records.

- l in TTEX in many records.

- THKCLS\_1 often has a 1.

\*\* No fixes required. Comments added to POLY\_COM for problem records.

- Usual TDEC=0 issues. (Processing aborted)

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\ValidateRows.py D:\Data\Processing\SR4\CPC\Stage\_3.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- No issues identified

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\ValidateGeometry.py D:\Data\Processing\SR4\CPC\Stage\_3.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- No issues identified

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\CreateErrorSummary.py D:\Data\Processing\SR4\CPC\Stage\_3.gdb\TEIS\_Master\_Long\_Tbl\_CPC

Stage\_4 Created

\* Moved Stage\_3 to P:

\* Copied features from Stage\_3 to Stage\_4 (141285 features)

P:\Projects\09036\_MoE\_TEIMS\Working\Staging\_Area\CPC\Standard\_Fixes.py D:\Data\Processing\SR4\CPC\Stage\_4.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- Completed Sucessfully.

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\ValidateDomains.py D:\Data\Processing\SR4\CPC\Stage\_4.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- No new issues/No fixes required (2985 errors)

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\ValidateRows.py D:\Data\Processing\SR4\CPC\Stage\_4.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- No issues

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\ValidateGeometry.py D:\Data\Processing\SR4\CPC\Stage\_4.gdb\TEIS\_Master\_Long\_Tbl\_CPC

- No issues

P:\Projects\09036\_MoE\_TEIMS\Working\TEIS\_Environment\Scripts\CreateErrorSummary.py D:\Data\Processing\SR4\CPC\Stage\_4.gdb\TEIS\_Master\_Long\_Tbl\_CPC

------------------------------------------

2010-01-12

- Copied Stage\_4 to Stage\_5 (141285 features)

\* Moved all stages back to P: