Getting started with Civilian Topographic Map (CTM) Distributed Generalization Workflows

**Introduction**

Civilian Topographic Map (CTM) Distributed Generalization Workflows are designed to execute the CTM Generalization Models by partitioning the input data and distributing it to multiple machines for processing. These workflows are built using ArcGIS Workflow Manager (WMX). In addition to Workflow Manager Workflows, the distributed generalization requires that services be activated on all the machines doing the processing. This document describes the process of setting up the CTM Workflows and services in your environment.

These workflows are provided as an example and can be modified to meet your unique generalization processes.

Prerequisites:

* ArcGIS Desktop 10.5 or higher
* ArcGIS Workflow Manager Desktop 10.5 or higher
* ArcGIS Production Mapping Desktop 10.5 or higher
* CTM 3.0 or higher
* Manually test the CTM Generalization Models according to the Getting Started with CTM Generalization document on each machine. This will ensure that the CTM files and software are configured correctly on each machine. Failure to setup CTM correctly on each machine will likely cause the workflows to fail.

Contents

[Getting started 2](#_Toc471824145)

[About the CTM Distributed Generalization Workflows 4](#_Toc471824146)

[CTM 50K Generalization - Parent job type 4](#_Toc471824147)

[CTM 50K Generalization - Child job type 4](#_Toc471824148)

[Extended properties 5](#_Toc471824149)

[Import Distributed Generalization Workflows 7](#_Toc471824150)

[Loading workflows into a new Workflow Manager Repository 7](#_Toc471824151)

[Loading workflows into an existing Workflow Manager Repository 8](#_Toc471824152)

[Setup Dependencies 9](#_Toc471824153)

[Create a Shared Directory 9](#_Toc471824154)

[Create a Local Directory on all Processing Machines 10](#_Toc471824155)

[Copy the CTM Generalization Models to the Shared Directory 10](#_Toc471824156)

[Import the Input Database into SDE 10](#_Toc471824157)

[Shared Databases 11](#_Toc471824158)

[Input Database 11](#_Toc471824159)

[Output Database 11](#_Toc471824160)

[Product Library Database 11](#_Toc471824161)

[Create Partition Feature Class 12](#_Toc471824162)

[Update Workflow Configurations 12](#_Toc471824163)

[Users and Security 13](#_Toc471824164)

[Data Workspaces 14](#_Toc471824165)

[Job Types 14](#_Toc471824166)

[Emails 16](#_Toc471824167)

[Manually Test Workflows 16](#_Toc471824168)

[Manually Test the Workflows 16](#_Toc471824169)

[Configuring Distributed Processing Machines 21](#_Toc471824170)

[Ensure users running background process have permissions 21](#_Toc471824171)

[Access to Workflow Manager 21](#_Toc471824172)

[Access to Directories 21](#_Toc471824173)

[Configuring Distributed Processes 21](#_Toc471824174)

[Configure a Parent Process 21](#_Toc471824175)

[Configure Child Processes 22](#_Toc471824176)

[Running Jobs in the Distributed Environment 23](#_Toc471824177)

[Deleting Test Jobs 24](#_Toc471824178)

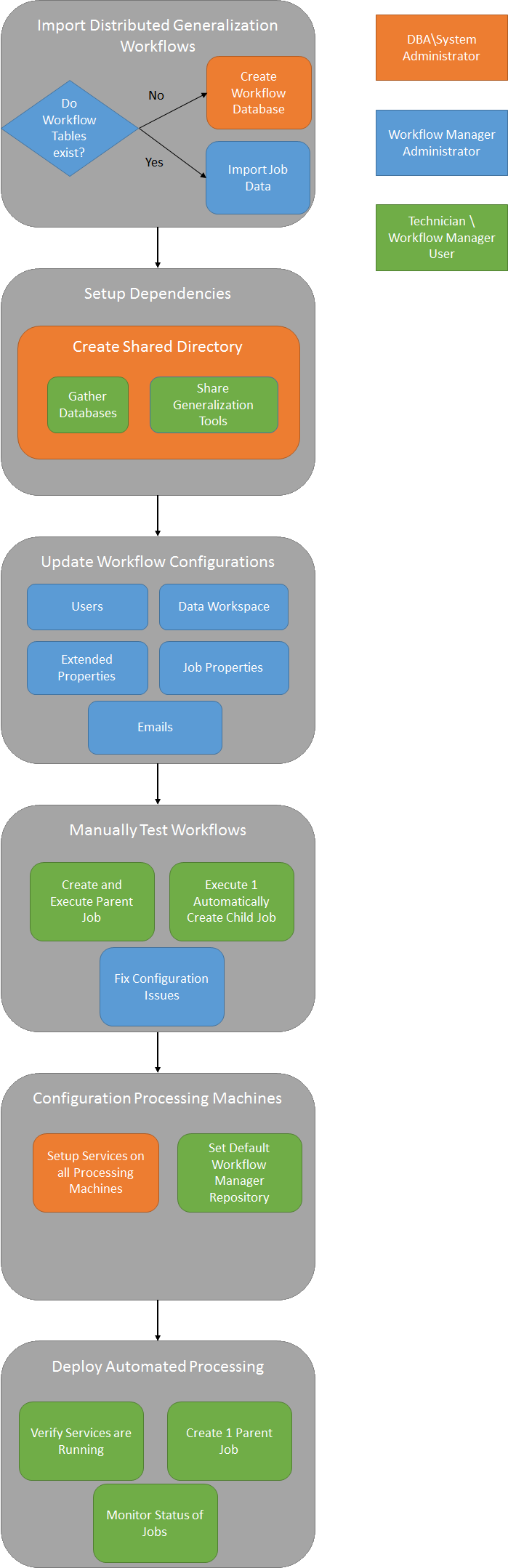
[Starting the Distributed Processes 24](#_Toc471824179)

[Create a Parent Job 24](#_Toc471824180)

[Stopping the Distributed Processes 25](#_Toc471824181)

# Getting started

This document walks you through the steps of setting up the CTM Workflows in your environment. The diagram below shows the overall process at a higher level. You may need to involve a few people in setting up the processes. The color of the steps in the workflow indicates the role of the people involved in the process. Listed below are the roles and a brief description of each role.

**DBA/System Administrator**: The orange steps will be executed by a Database Administrator or a Systems Administrator. These are individuals who have permissions to create new tables in an SDE geodatabase and run programs as an administrator on the machines you will use for processing.

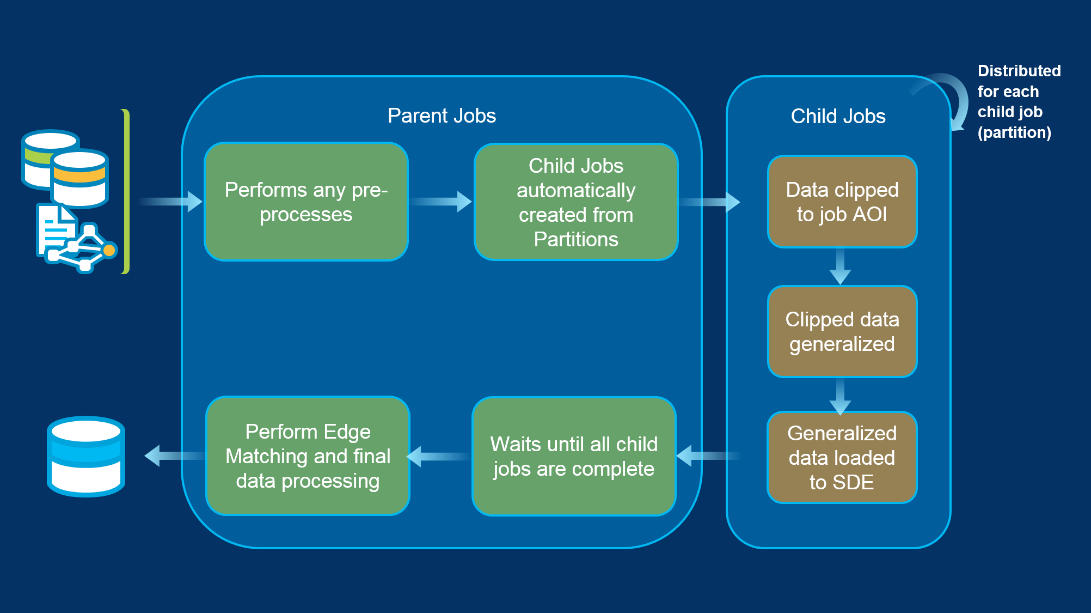
**Workflow Manager Administrator**: The blue steps will be executed by a Workflow Manager Administrator. This is someone who has been granted the Administrator role in Workflow Manager. This is typically a senior technician or technical lead who understands how to configure Workflow Manager and can make decisions about how to setup the environment.

**Technician/Workflow Manager User**: The green steps can be executed by any Workflow Manager user. This can be the same individual that executes the Workflow Manager Administrator setup steps but does not have to be. This individual must be a user in the Workflow Manager system but does not have to have Administrator access. This individual will be running the generalization workflows and monitoring the status of the jobs.

# About the CTM Distributed Generalization Workflows

The CTM Distributed Generalization Workflows illustrate how to use Distributed Generalization Tools provided in Production Mapping with the Generalization Models created for CTM. The CTM Distributed Generalization Workflows provided include two job types and an extended property table for storing information used while executing the workflows.

Distributed Generalization Workflows use parent and child workflows in Workflow Manager. When executing these workflows, a single parent job will be created to manage the work over the entire extent of the data being processed. One child job will be created for each partition that the data will be divided into.



## CTM 50K Generalization - Parent job type

With the CTM 50K Generalization - Parent job type, you will define the default values that will be used when running the generalization, information such as where the Generalization Models, partitions, and input database are located. The parent job will perform any work that needs to happen in a single process, such as dynamically creating the partitions or performing edge matching on the final database. The parent job creates a child job for each partition and waits for all the children to execute before moving on to complete its work.

## CTM 50K Generalization - Child job type

The CTM 50K Generalization - Child job type defines the work that will be distributed to run across multiple machines. One child job is created for each partition. The child job type shows how to extract data from the input database and run the Generalization Models. At the end of the child job the generalized data will be added to the job as an attachment, this makes the data available to the parent job to load into the output database.

The child job also includes a loop that allows you to restart the job if something goes wrong. One of the extended properties is the maximum retry count. If a failure happens, the job will rerun from the beginning until it reaches the maximum number of retries. If the job does not successfully complete after reaching the maximum retry count, then an email will be sent to the job owner.

## Extended properties

Extended properties are used to store information used by the Generalization Workflows. The table below describes the properties that are tracked, how they are used, and whether or not you should update the default value for this property in the parent job type.

|  |  |  |
| --- | --- | --- |
| Extended Property | Use | Update Default |
| Dependency Directory | Stores the location of a shared directory where the latest version of all the custom files and models (dependencies) used by the process are stored. The files in this location will be copied locally. | Yes |
| Input Database | The full path to the input database or database connection file that contains the data that will be used as an input to the generalization. | Yes |
| Output Database | The full path to the output database or database connection file that will be populated with the results from the generalization. This database should be empty when the workflows are started. | Yes |
| Local Directory | A folder path that exists on all machines that will do the processing. Any files copied locally to the machine will be copied into a directory within this local directory. Make sure the directory exists on all machines. | Yes |
| Job Directory | A unique directory that will be created for each job. The directory will be created within the specified Local Directory path. The value for this property will be automatically populated once the directory is created. | No |
| Buffer Distance | When data is extracted from the input database, the area of interest or partition, will be buffered by the specified distance to bring in additional data for context. Value should be populated with number and distance units, i.e. 100 Meters or 2000 Feet. | Yes |
| Partition Feature Class Name (qualified) | Just the name, not the path to a polygon feature class in the Input Database containing the extents that will be used divide the data. One child job will be created for each polygon in this feature class.  NOTE: If the partition feature class is not in the Input Database, the parent job will not run correctly without changing the functionality of some of the steps.  NOTE: If you are not connecting to the input database as the owner of the partition feature class, you must specify the fully qualified name. | Yes |
| Extract Database | File geodatabase created with the data extracted from the input workspace over the job extent. This value is automatically populated when the database is created and will be located in the job directory. | No |
| Generalized Database | File geodatabase created with the data resulting from running the generalization processes. This value is automatically populated when the database is created and will be located in the job directory. | No |
| Machine Name | The name of the machine that is running or has run the job. Automatically populated when a machine begins to execute the job. | No |
| Process ID | ID of the process on the machine that is running the job. Automatically populated when a machine begins to execute the job. | No |
| Failure Count | The number of times the job has failed to execute. This number is automatically updated whenever a failure is detected. | No |
| Maximum Retry Count | The maximum number of times the job should be restarted if a failure occurs. As long as the failure count value is less than this value, the job will try to run again if a failure was detected. | Yes |
| Child Job Count | The number of child jobs that are created for the parent. | No |
| Loaded Job Count | The number of child jobs that have had their resulting generalized data loaded into the final database. | No |
| Child Job Type | The name of the job type that will be created as child jobs. | Yes |
| Wait Time | How long the parent job will wait to check if any new child jobs are complete. | Yes |
| Job Duration | The time taken to execute the job. Automatically populated when the job is completed. | No |
| Fully Qualified Prefix | Is the fully qualified database and user that will be used within certain steps to ensure that the distributed process can see the appropriate data. An example would be database.USER as an entry. This is optional and should be used if the person is not the schema owner | Yes |
| Dependency Extract | Is the location that the dependencies will be extracted on each machine. For example, C:\DistributedGeneralization | No |

# Import Distributed Generalization Workflows

CTM Distributed Generalization Workflows are built using ArcGIS Workflow Manager. To use these workflows, you must first ensure that ArcGIS Workflow Manager for Desktop and Esri Production Mapping for Desktop has been installed on the machines of all technicians that will interact with the workflows and all the machines that will be used for automated processing of the generalization.

There are many steps involved in setting up the CTM Distributed Generalization Workflows and the Workflow Manager repository. It is recommended setting up, test the workflows manually on one machine before deploying to all machines and automating the execution of the workflows.

Once Workflow Manager has been installed, you must setup a Workflow Manager repository. The Workflow Manager repository must be an SDE geodatabase. When setting up the Workflow Manager, guidebooks are available to help you appropriately configure the SDE repository.

* + For more information about administering Workflow Manager in **SQL Server** see: <http://desktop.arcgis.com/en/arcmap/latest/extensions/workflow-manager/introduction-to-storing-workflow-manager-workspace-in-sql-server.htm>
  + For more information about administering Workflow Manager in **Oracle** see: <http://desktop.arcgis.com/en/arcmap/latest/extensions/workflow-manager/introduction-to-storing-the-workflow-manager-workspace-in-a-geodatabase-in-oracle.htm>

Once your Workflow Manager repository has been created, you are ready to load the workflows provided.

## Loading workflows into a new Workflow Manager Repository

If you have an SDE geodatabase that does not have the Workflow Manager tables, follow the steps in this section to create the tables and load the CTM Distributed Generalization Workflows. If you already use Workflow Manager or someone else setup the Workflow Manager repository using the steps laid out in the SQL Server or Oracle guide books referenced above, follow the steps in the section below for loading the workflows into an [existing repository.](#_Loading_workflows_into)

*NOTE: Before beginning the process, make sure that you have the connection properties (username and password) for the user that will own the workflow manager tables in the database. This user must have permissions to create tables in the database.*

1. Start ArcMap or ArcCatalog.
2. If necessary, enable the **Workflow Manager Extension** window by clicking **Customize > Extensions… > Workflow Manager** on the main menu.
3. Browse to or search for the **Create Workflow Database** Geoprocessing tool. If browsing the tool is located in System Toolboxes > Workflow Manager Tools > Configuration.
4. Browse to an SDE connection file to the Workflow Manager Repository for the **Input Database Connection**. Make sure that the connection file contains the username and password of a user with create table privileges to the database.
5. Choose the **Spatial Reference** for the Area of Interest feature class.

The Spatial Reference will typically match the spatial reference of the data you are editing, however, it does not have to match. Ensure that the spatial reference is appropriate to the geographic area over which you will be creating jobs. For CTM, you can use **WGS 1984** as the spatial reference to ensure that you can create AOIs anywhere in the world.

*NOTE:* *The spatial reference of the Workflow Manager Area of Interest must match the spatial reference of the partition feature class you will use to create jobs.*

1. For the Import Configuration option, choose **Custom Configuration**.
2. For the Input Custom Configuration, select the Database Configuration > DistributedGeneralizationWorkflows > **CTM\_DistributedGeneralization.jxl** file provided with the CTM configurations.
3. Set the value for User Store if desired. If no value is set, Workflow Manager will create the repository using Traditional user management which grants users permissions based on their windows user name. You can change the value to Portal to grant user permissions based on their portal user.
4. Once all the parameters are set, click **OK**.
5. Close ArcMap or ArcCatalog.

## Loading workflows into an existing Workflow Manager Repository

If you already use Workflow Manager or someone else setup the Workflow Manager repository using the steps laid out in the SQL Server or Oracle guide books referenced above, you will use the steps in this section of the document to load the workflows into your repository. If you have an empty SDE geodatabase that you will use for Workflow Manager, follow the steps below for loading the workflows into a [new repository](#_Loading_workflows_into_1).

*NOTE: Before beginning the process, make sure that you have the connection properties (username and password) for the user that is the schema owner of the workflow manager tables in the existing repository.*

1. Start ArcMap or ArcCatalog.
2. If necessary, enable the **Workflow Manager Extension** window by clicking **Customize > Extensions… > Workflow Manager** on the main menu.
3. If necessary, create a connection to the Workflow Manager repository.
   * In the Catalog window, expand the **Workflow Manager Databases** node.
   * Click **Add** **Workflow Manager Database**.
   * Enter the connection properties for the user that is the schema owner of the Workflow manager tables. Click **OK**.
   * Right-click on the newly created connection. Choose **Set as default database**.
4. Browse to or search for the **Import Job Data** Geoprocessing tool. If browsing the tool is located in System Toolboxes > Workflow Manager Tools > Configuration.
5. For the Input JXL/Acknowledgement, select the WorkflowManager> DistributedGeneralizationWorkflow> **CTM\_DistributedGeneralization.jxl** file provided with the CTM configurations.
6. Enable the Merge option, if desired. It is recommended that you enable this option to combine the contents of the Distributed Generalization Workflows with the content already existing in your Workflow Manager database. If the option is disabled, all of your existing content will be deleted.
7. If necessary, browse to your .jtc connection file to your Workflow Manager Repository. If you have already connected to your repository and set it as your default Workflow Manager Database, you do not need to populate this parameter.
8. Click **OK** to import the configurations.

for loading the workflows into an [existing repository.](#_Loading_workflows_into)

# Setup Dependencies

Dependencies are the data, files, and Generalization Models used by the workflows. The dependencies must be gathered and shared in a single location in order for the workflows to successfully run. For performing the CTM 50K generalization the dependencies are:

* 25K production data
* Partitions
* Product Library
* Generalization Hierarchy file
* Output Database
* CTM Generalization toolbox and scripts

## Create a Shared Directory

In order for everyone in the organization to be able to run the workflows, all of the Generalization Models and databases must be available for all of the machines to access. For ease of use, we recommend that you create a shared folder somewhere within your organization where the latest version of the files will be kept.

*NOTE: Your shared directory can be any location you wish. The location of the dependency directory will be an extended property in the job and can change as necessary. However, in order for the workflows to execute correctly, you MUST name and organize all subfolders and files as described below. The location and name of the files are configured in the workflows to be relative to this shared directory.*

1. Identify a shared folder path.
2. Ensure that you have write access to that path.
3. Create a folder named “**DistributedGeneralizationWorkflows**” in the identified path.
4. Grant read access to the DistributedGeneralizationWorkflows directory of the users and machines that will be running the workflows.

## Create a Local Directory on all Processing Machines

The CTM Distributed Generalization Workflows create various databases and files as the workflows are executed. For performance it is recommend that you create a folder locally on each processing machine that will be the root directory under which all of these files will be created. Here are a few tips for creating this local directory.

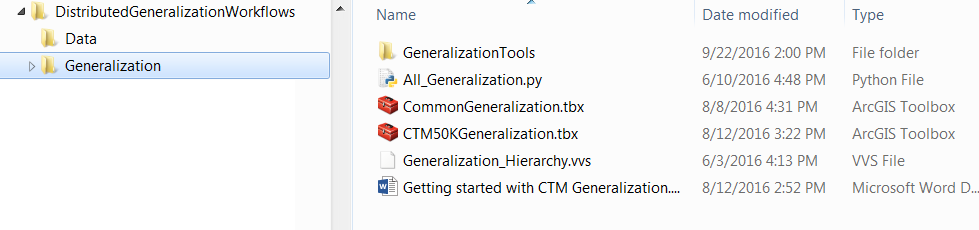
* The folder should be named the same for ALL machines that will run jobs.
* The user that will be running the jobs will need write access to this folder.
* The folder will be used only by the processes running on that machine so it does not need to be shared.

## Copy the CTM Generalization Models to the Shared Directory

The Generalization Models will be executed during the child workflows. All of the Generalization Tools, toolboxes, and scripts provided in the CTM download must be copied to the shared directory.

1. Navigate to the location where you downloaded the CTM content.
2. Select the Generalization folder.
3. Copy the **Generalization** folder and all its content to your ‘**DistributedGeneralizationWorkflow**’ shared directory.

Once copied, the shared location should have a generalization subdirectory with all of the provided generalization files.



## Import the Input Database into SDE

For generalization, the Input Database is the database that contains the data that you will use as the starting point for generalization. The Distributed Generalization Workflows are designed to use an SDE geodatabase as the input database. The child jobs created, start by extracting data from the input database. There will be multiple machines and processes extracting data at the same time so the data must be in an SDE geodatabase to prevent database locking issues.

If you do not have your SDE input database configured yet, you can use the SaltLakeCity sample data. The SaltLakeCity.zip in Fixed25K\SampleData contains a file geodatabase with data near Salt Lake City, UT. This database can be unzipped and copied to an SDE database.

When setting up the production database in SDE, best practice documents are available to help you appropriately configure the SDE repository.

* + For more information about administering your production database in **SQL Server** see:

<http://desktop.arcgis.com/en/desktop/latest/guide-books/extensions/production-mapping/introduction-to-storing-the-production-mapping-workspace-in-sql-server.htm>

* + For more information about administering your production database in **Oracle** see: <http://desktop.arcgis.com/en/desktop/latest/guide-books/extensions/production-mapping/introduction-to-storing-a-production-mapping-geodatabase-in-oracle.htm>

## Shared Databases

In order for the workflows to know what databases to use for the input and output, the databases or connection files must be included in the shared directory.

### Input Database

The input database is the SDE connection file to your Production Database.

1. Create a folder named “**Data**” within your DistributedGeneralizationWorkflows shared directory
2. Add a connection file to the SDE Production Database you configured in the section above.
3. Name your connection file **CTM\_25K.sde**.

### Output Database

The output database can be either a file geodatabase or SDE geodatabase that will contain the final results of the generalization.

*NOTE: The output database should be an empty database when you begin running the generalization processes. The tools used in the CTM Workflows append data into the output database. Data may be duplicated in the output database if data is not deleted between runs of the Generalization Workflows.*

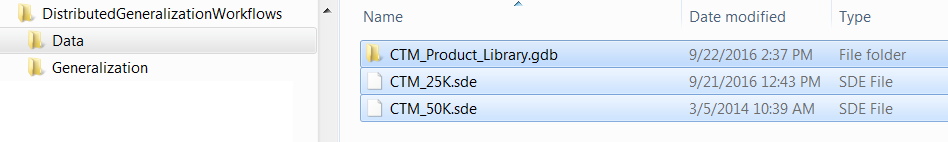
1. Navigate to the **Data** folder in the DistributedGeneralizationWorkflows shared directory
2. Add an empty file geodatabase or a connection file to the empty SDE repository.
3. The name of the database should be **CTM\_50K.gdb** or **CTM\_50K.sde** as appropriate.

### Product Library Database

The Product Library is used by the CTM Generalization Models to apply symbology to the generalized features. The CTM Workflows are setup to use the Product Library file geodatabase provided in the CTM content. The Product Library also needs to be copied to the shared directory.

1. Navigate to the location where you downloaded the CTM content.
2. Open the Product Library folder.
3. If necessary, unzip the CTM\_Product\_Library.zip.
4. Copy the CTM\_Product\_Library.gdb to the Data folder in the DistributedGeneralizationWorkflows shared directory.

When all of the required databases are copied to the shared directory, the Data folder should contain the following content.



## Create Partition Feature Class

The partition feature class is a polygon feature class where each feature is the extent of an area that will be extracted and processed during generalization. The partition feature class can be created using whatever means you wish. However, the Generalization Workflows expect the feature class to exist within the input SDE database. The workflows also require that the input database be registered as a data workspace with Workflow Manager, which you will do later.

If you do not already have a partition feature class, the steps below describe how to use the Create Cartographic Partitions tool to create partitions for the SaltLakeCity sample data.

*NOTE: If you choose not to follow these steps, make sure you add your partition feature class to the Input Database.*

1. Open ArcMap.
2. Browse to or search for the **Create Cartographic Partitions** Geoprocessing tool. If browsing the tool is located in System Toolboxes > Cartography Tools > Generalization.
3. For the Input Features, browse to the **TransportationGroundCrv** feature class in your Input Database.
4. Click the folder for the Output Features parameter.
5. Browse to and open your **SDE Production Database**.
6. Enter the name “**Partitions**”.
7. Click **Save**.
8. Enter 3000 for the Feature Count.
9. Click OK.

The Create Cartographic Partitions tool will create polygons that contain no more than the specified number of features. The parameter above should produce 10 polygons when using the Salt Lake City sample data.

# Update Workflow Configurations

Once you have added the CTM Distributed Generalization workflow configurations to your Workflow Manager repository and gathered all of your dependencies in a shared directory, there are a number of items which must be updated to match your environment. Once these items have been configured, you will be ready to deploy to all the machines requiring access to Workflow Manager.

1. Start ArcGIS **Workflow Manager Administrator.**
2. The list on the left of the Workflow Manager Administrator will contain a list of your available Workflow Manager repositories. Find the Workflow Manager Repository you created or updated in the Loading workflows section of the document.
   * If your repository is not in the list, click the Add Connection button.
   * Enter in the connection properties to the Workflow Manager database.
   * Click OK.
3. Ensure that your Distributed Generalization Workflow Manager repository has (default) at the end of its name.
   * If the repository is not the default, right-click and choose **Set As Default**.
4. **Double-click** the name of the Workflow Manager Repository to connect to the repository.

## Users and Security

Workflow Manager uses Groups and Privileges to control what functions an individual can do within the Workflow Manager application. The steps below describe how to add privileges to the user who will be configuring and testing the workflows. These steps can also be used when assigning privileges to other users, however, if you are maintaining a large number of users, you may also want to consider managing the user with Active Directory. See <http://desktop.arcgis.com/en/arcmap/latest/extensions/workflow-manager/configuring-users-and-groups.htm> for more information.

The Distributed Generalization configurations come with two key groups that are used in the workflows and during automation.

* **Generalization Technician** – Any person within the organization who will create generalization jobs, monitor job status’ or execute the generalization jobs should be added to this group.
* **Automated Processing** – No users should be assigned to this group. All jobs will be assigned to this group by default. The automated processes that will execute jobs in the background look for jobs to be assigned to this group to know that no other process or person is already working on the job.

1. Expand your Workflow Manager repository in the tree**.**
2. Expand the **Security** node in the tree.
3. Expand the **Groups** node.

Under Groups you will see a number of different groups, including the Generalization Technician and Automated Processing groups. Groups define the different roles individuals play within Workflow Manager. Individuals need the ability to do different tasks in the system based on what their roles are. For example, a Manager would need the ability to create new jobs or reassign jobs where as a Technician should only be able to work on items that have been assigned to them. When adding new users to Workflow Manager, you will want to ensure that they are added to the Group or Groups that are most appropriate for the type of work they are doing.

*NOTE: Until a user has been added to a Group, they will not have the ability to do any tasks in Workflow Manager.*

*NOTE: The Administrator Group is different from other Groups. If you wish for a user to be able to modify the Workflow Manager configurations, they must be added to the Administrator group.*

1. Expand the **Users** node.
2. Double-click your user.

If you are using an existing Workflow Manager repository, your name will likely appear in the list of Users. However, if you create a new repository you will likely see a user name “PostInstall User”. By default, when Workflow Manager is setup, the user name of the person who runs the Create Workflow Database Geoprocessing tool is added to Workflow Manager as “PostInstall User” and is added to the Administrator Group.

1. If necessary, update the First Name and Last Name values to reflect your name rather than PostInstall User.
2. If you wish to receive emails from Workflow Manager, update the email value to be a valid email address.
3. Turn to the **Groups** tab.

You will see that the Administrator group appears in the list. Other groups may appear if you are using an existing repository. To use the CTM Distributed Generalization Workflows, you will need to add additional groups.

1. Click the **Add** button.
2. Choose the appropriate groups from the list.

In order for you to test the workflows, you will need to add groups to your user account. At a minimum, add the **Generalization Technician** group. You can add more groups as appropriate.

1. Click OK to close the Choose user group dialog.
2. Click OK to close the User Properties dialog.
3. You can repeat steps 5-11 for other users that need to be manually added to Workflow Manager.

## Data Workspaces

The CTM Distributed Generalization Workflows require that the data being used as input to the Generalization Models be stored in an SDE database. See [Import the Input Database into SDE](#_Import_the_Production) section above.

Once your Production Database has been setup in SDE, you will need to register it with Workflow Manager and the appropriate Job Types.

*NOTE: Make sure that you have granted users the necessary permissions to the production database tables.*

1. Expand your Workflow Manager repository in the tree**.**
2. Expand the **GeoDatabases** node in the tree.
3. **Right-click** the Data Workspaces node in the Workflow Manager tree.
4. Choose **Add Data Workspace.**
5. Enter **CTM 25K Data** as the Database Alias.
6. Enter the connection information for the production database.
7. Click OK at add the database.

## Job Types

The extended properties on the Parent job type allow you to define where data and content are located in your environment as well as configure some properties about how the jobs will run.

1. If necessary, expand your Workflow Manager repository in the tree**.**
2. Expand the **Job Components** node.
3. Expand the **Job Types** node.
4. Double-click the **CTM 50K Generalization – Parent** job type.
5. Turn to the **Default Properties** tab.
6. From the **Data Workspace** drop-down, choose **CTM 25K Data**. This is the data workspace you made a connection to above.
7. Disable the Allow Data Workspace To Be Changed option.
8. Turn to the **Extended Properties** tab.
9. Expand the **WMX\_DISTRIBUTED\_GENERALIZATION** node.
10. Select each of the properties listed in the table below and set the value to the Default Value text box to the appropriate value. The table describes what each of the properties is used for as well as tips for what value you should enter.

|  |  |  |
| --- | --- | --- |
| Extended Property | Use | Default Value |
| Dependency Directory | Stores the location of a shared directory where the latest version of all the custom files and models (dependencies) used by the process are stored. The files in this location will be copied locally. | Enter the UNC path to the DistributedGeneralizationWorkflows shared directory you created in the [Create a Shared Directory](#_Create_a_Shared) section above. |
| Input Database | The full path to the input database or database connection file that contains the data that will be used as an input to the generalization. | Enter the full path to the CTM\_25K.sde file you shared in the [Input Database](#_Input_Database) section above. |
| Output Database | The full path to the output database or database connection file that will be populated with the results from the generalization. This database should be empty when the workflows are started. | Enter the full path to the CTM\_50K.sde or CTM\_50K.gdb file you shared in the [Output Database](#_Output_Database) section above. |
| Local Root Directory | A folder path that exists on all machines that will do the processing. Any files copied locally to the machine will be copied into a directory within this local directory. Make sure the directory exists on all machines. | The local path to the folder you created in the [Create a Local Directory on all Processing Machines](#_Create_a_Local) section above. |
| Partition Feature Class | The name of a polygon feature class containing the extents that will be used divide the data. The polygon feature class must reside in the database you specify as the Input Database. One child job will be created for each polygon in this feature class. | The name of the partition feature class you created in the [Create Partition Feature Class](#_Create_Partition_Feature) section above. The polygon feature class must reside in the database you specify as the Input Database. |
| Buffer Distance | When data is extracted from the input database, the area of interest or partition, will be buffered by the specified distance to bring in additional data for context. Value should be populated with number and distance units, i.e. 100 Meters or 2000 Feet. | Enter the value you wish to use to buffer each partition by when extracting the data from the input database. 250 Meters is recommended when using the CTM Salt Lake City sample data. |
| Maximum Retry Count | The maximum number of times the job should be restarted if a failure occurs. As long as the failure count value is less than this value, the job will try to run again if a failure was detected. | Enter the number of times you would like to retry running a job if it fails the first time. Any number is appropriate for this parameter. |
| Child Job Type | The name of the job type that will be created as child jobs. | Enter the name of the child job type that will be used to create child jobs for each partition. |
| Wait Time | Once the parent job has created the children, it will begin looking for completed child jobs to load into a single output database. This parameter specifies how long the parent job will wait between each time it checks if any child jobs are complete. | Enter the time **in seconds**. |
| Fully Qualified Prefix | If you are not the schema owner, you must enter the Fully Qualified Prefix i.e. database.USER. This is optional and should be used if the person is not the schema owner. | C:\DistributedGeneralizationWorkflows |

## Emails

Workflow Manager is designed to automatically send email notifications when certain events happen in the lifecycle of a job. For example, when a job is created or a job is assigned to a new person. Some email notifications are pre-configured in the CTM Workflows, however, in order for emails to be sent, you need to provide information about your email server.

1. Select the Workflow Manager repository in the Workflow Administrator tree.
2. Click the **Settings** button on the Main Menu, Choose **Workflow Manager System Settings**.
3. Turn to the **Notifications** tab.
4. Populate the Notification Settings with information about your email server.
5. Click OK.

# Manually Test Workflows

Once you have updated the Workflow Manager configurations provided, you will want to test on a single machine before deploying to all machines and users.

## Manually Test the Workflows

Everything should now be configured so you can begin testing the workflows. This section describes how to test the parent and child job types. When running the workflows, you will only need to manually create a parent job, the child jobs will automatically be created by the parent.

*NOTE: Before launching Workflow Manager, make sure your Workflow Manager Repository with the CTM Distributed Generalization Workflows is set as the Default. See steps 1 through 4 of* [*Update Workflow Configurations*](#_Updating_Workflow_Manager) *for more information.*

1. Open Workflow Manager.
2. Click **Create new jobs(s)**.
3. Choose the **CTM 50K Generalization - Parent** Job Type.
4. Change the Assignment to your user name.
5. Verify the Data Workspace. It should be populated with CTM 25K Data.
6. Click OK to create the job.
7. Turn to the **Job Workflow** Tab.
8. Click the Run Current Step.

The job should be executed through the **Populate Merge Database Property** step. The Merge Child Data step will not do anything until some of the child jobs have completed. The steps in the workflow are designed to be run by automated processes. So you will not see any dialogs pop up. The workflow is likely still executing and you will see the highlighted step change.

The table below explains how you can verify the steps that have run successfully.

|  |  |
| --- | --- |
| Step Name | Items to verify |
| Create Folder | Creates a folder in your local root directory with the same name as the parent job (…\CTM\_Parent\_Job\_#). The full path to this folder will be updated in the Job Directory extended property. |
| Attach Dependencies | Creates a zipped folder of anything necessary for the process to run such as the initial data, output database, Generalization Models, etc. |
| Does Job have AOI? | Checks to see if the job already has an area of interest. If not, the Create Job AOI from Partitions step will run to create the Area of Interest. |
| Create Job AOI from Partitions | Sets the Area of Interest for the job. You will be able to see if there is an area of interest for the job by turning to the LOI (location of interest) tab. The Area of Interest will be the extent of the partition feature class you created earlier. |
| Does Job have Children? | Checks to see if the job has children. For example, it will create one child job for one partition. |
| Create Partition Jobs | Will create a child job for each feature in your partitions feature class. To verify jobs were created, click the All Jobs query in Workflow Manager. You should see multiple jobs of Job Type CTM 50K Generalization – Child. |
| Get Child Job Count | Will populate the Child Job Count extended property with the number of child jobs that were created by the Create Partition Jobs step. |
| Does Merge Database Exist? | Checks if Merge database exists, if it does not one will be created in the Create Merge Database step. |
| Create Merge Database | Will create an empty file geodatabase named “MergedData.gdb” in the job directory for the Parent job. |
| Populate Merge Database Property | Populates the full path to the Merge Database created by the Create Merge Database step in the Merge Database extended property. |

1. Run the **All Jobs** query.
2. Choose one of the **CTM 50K Generalization - Child** Jobs.
3. If necessary, turn to the **Properties** tab.
4. Assign the job to yourself.

By default, all child jobs will be assigned to the Automated Processing group. This is so the services looking for jobs to run will know these jobs are available for processing.

1. Turn to the **Extended Properties** tab.

Set the Dependency Directory, Input Database and Output Database.

1. Turn to the **Job** **Workflow** tab.
2. Click the Run button.

If the job is configured correctly, it will execute straight through the main path and stop at the Wait for Jobs to Complete step. The steps in the workflow are designed to be run by automated processes. So you will not see any dialogs pop up. The workflow is likely still executing and you will see the highlighted step change.

*NOTE: When verifying extended properties, you may need to turn from the Extended Properties tab to another and back to see the values in the properties refresh.*

The table below explains how you can verify the steps have run successfully.

|  |  |
| --- | --- |
| Step Name | Items to verify |
| Populate Job Properties from Parent | Copies the values for some of the extended properties from the parent job to the child job. |
| Extract Dependencies | This unzips Dependencies onto the child machine to be utilized and will go to the Dependency Directory extended property folder. |
| Populate Machine Name | Sets the name of the computer you are running the workflow from in the Machine Name extended property. |
| Create Folder | Creates a folder for the child job. If a folder for the parent job does not exist, it creates a folder in your local root directory with the same name as the parent job (…\CTM\_Parent\_Job\_#). Then creates a folder for the child job within that Parent job folder. The full path to this folder will be updated in the Job Directory extended property. |
| Extract Data | Creates a file geodatabase in the Job Directory path named the same as the job name. When the step is completed the file geodatabase will contain all the data from the input workspace that is within the specified buffer distance of the job AOI. |
| Populate Extract Database Property | Updates the Extract Database extended property with the full path to the database created by the Extract Data step. |
| Run Generalization | Runs the Generalization Models against the extract database. For CTM, as the Generalization Models are run, you will see databases being created in the Job Directory folder. The <job name>\_generalize.gdb is the data being generalized. As various Generalization Models complete running, backup databases will be created in the job directory as well. |
| Populate Generalization DB Property | Updates the Generalized Database extended property with the full path to the database containing the final results from the Run Generalization step. |
| Clip to Job AOI | Splits the data in the database specified in the Generalized Database extended property. All data outside of the job Area of Interest will be removed from the database. This is used to remove the data in the buffer area that was included during the extract. |
| Attach Generalized Data | Zips the database specified in the Generalized Database and adds it as an attachment to the job. The zip file will be named GeneralizedData.zip |
| Populate Job Duration (Minutes) | Updates the Job Duration extended property with the time in minutes that it took from the Job Started Date to the time when the step is run. |
| Wait for Jobs to Complete | This step is a placeholder that is used to ensure all processing nodes have completed their work before moving on to merge the results to a final output. |

1. If something goes wrong while running key steps in a parent or child workflow, the workflow will enter the job failure loop.

The table below explains how you can verify the steps in the failure loop have run successfully.

|  |  |
| --- | --- |
| Step Name | Items to verify |
| Increment Failure Count | Add 1 to the current value in the Failure Count extended property. |
| Clear Process ID | Used by the automated processing scripts. After the step is run the value in the Process ID extended property should be blank. |
| Clear Machine Name | Used by the automated processing scripts. After the step is run the value in the Machine Name extended property should be blank. |
| Exceeded Maximum Retry Count | Compares the value in the Failure Count extended property to the Maximum Retry Count extended property. If the Failure Count is less than the Maximum Retry value, the job will move to the Reassign to Automation Group step. Otherwise the job will move to the Send Failure Notification step. |
| Reassign to Automation Group | Changes the assignment of the job from your user to the Automated Processing Group. The status of the job should be set to Ready to Work. Based on these criteria the job is now in a state where the automated processes will begin to execute the job again from the start. |
| Send Failure Notification | Sends an email to the Job Owner indicating that the job has failed and will not be automatically run again. |

1. Once the child job(s) have completed, you can run the final steps in the parent workflow.

If necessary, Run the All Jobs query and select the parent job.

1. Click the Run button.

The parent job should execute through to the end.

The table below explains how you can verify the remaining steps in the parent job have run successfully.

|  |  |
| --- | --- |
| Step Name | Items to verify |
| Merge Child Data | Loops through all the child jobs in the “Done Working” state. Looks for an attachment on the job named GeneralizedData<jobid>.zip. If the file exists, it is extracted from the job and the data is loaded into the Merge Database. When the data from a child job is loaded the Loaded Job Count extended property is increased by 1. When completed, the merge database should have data from all partitions (child job) loaded. |
| All Child Jobs Loaded? | Compares the Child Job Count property (which contains the number of all child jobs) to the Loaded Job Count property (which contains the number of child jobs whose generalized data has been loaded into the merge database). If the two values are equal, the job will move on. If the numbers are not equal, then not all of the child data has been loaded and the job will loop back to the Merge Child Data step. |
| Wait | If all of the child jobs are not loaded, the job will wait for the number of seconds specified in the Wait Time extended property before moving back to the Merge Child Data Step. |
| Edge Match | This will connect the features back together along artificial divisions or edges. The edge match process will do things such as snapping features together or removing duplicates but it’s primary action is to dissolve features back together across the partition edges. |
| Extract Data to XML | If all the child jobs have been loaded, extracts the content (schema and data) from the merge database into a workspace xml. The XML file is named merged\_data.xml and is created in the parent job directory. |
| Load XML to Output Database | Loads the workspace xml created from the Extract Data to XML step into the database specified in the Output Database extended property. |
| Populate Job | Will write how many hours the job took to complete. |
| Job is complete | Job will close when completed. |

If the job enters the failure loop, the History and Attachments can be used to help determine what went wrong.

* + Turn to the **History** tab and identify the last step that ran before Increment Failure Count. This is the step that is most likely to have caused the issue.
  + Turn to the **Attachments** tab. Look for the log file for the step you identified.
    1. If the step ran with issues, the log file will indicate what the issues were.
    2. If there is no log file, then the step failed to run. This could be because the step is not configured to run correctly or an extended property being used as an input is either blank or contains an incorrect value.

When running the steps automatically, the Notes tab may also provide information about what went wrong and what step was active when the failure occurred. This information will not be populated when running the job manually.

# Configuring Distributed Processing Machines

Once you have tested the workflows and ensured that they work, you will deploy the Workflow Manager configurations to the processing machines. Then you will configure the machines to automatically execute the jobs in Workflow Manager using the Distributed Process Manager.

## Ensure users running background process have permissions

### Access to Workflow Manager

In order for an individual to be able to use the Workflow Manager application or to execute steps in a workflow, they must be added as Users in the Workflow Manager system and granted privileges. Before inviting others to use Workflow Manager, ensure that you have assigned them to the correct groups in Workflow Manager. For more information about adding users see the [Users and Security](#_Users_and_Security) section of this document.

The processes that will automatically execute the jobs will be run by a user. You will need to ensure that the owner of the process is a user in Workflow Manager.

### Access to Directories

When executing the workflows, the user running the process will also need to have access to the files and databases in the shared directory. Make sure that all required users have access to the shared directory created in the [Setting up Dependencies](#_Setting_up_Dependencies) section of this document.

Ensure that all machines have a folder in the location specified default value for the Local Root Directory extended property. See the [Create a Local Directory on all Processing Machines](#_Create_a_Local) section if you have not set up the Local Root Directory already.

# Configuring Distributed Processes

For each machine you will use the Distributed Process Manager to run jobs, you will need to configure a few properties so the machine knows exactly what types of jobs it can run and how many. The configurations are saved, so as long as none of the properties change between runs, you only need to configure the processes once.

## Configure a Parent Process

As described above, a single parent job will be created each time you want to run generalization. Since there is only one parent job, you will want to configure only one parent job process. This does not mean one parent process on each machine but rather one process on one machine. The parent machine will be the location where all of the child data is brought together into a single database so be sure there is enough space on the machine to contain all of the generalized data.

1. Connect to the machines that will be used to execute the parent job
2. Navigate to **C:\ProgramData\ESRI\Production\Desktop\DistributedGeneralizationWorkflows**
3. Open the **DistributedProcessManager.exe**
4. Click the **Add New Process** button

The Add New Process dialog is where you will specify what type of jobs you will run as well as how many processes you will run on the machine.

1. For Job Type, choose CTM 50K Generalization – Parent
2. For Role, choose **Parent**

Choosing Parent for the role will automatically set the number of processes to 1.

1. For Default Status, choose **ReadyToWork**

This is the status that the job is automatically set to when it is first created. The processes that run the jobs only look for jobs in this status. The processes assume that if a job is not in this status it is currently being run by another process.

1. For Default User, choose the user who will create the Parent Job

This is the user that the job will be assigned to by default. The processes that run the jobs, only look for jobs in assigned to this user. The processes assume that if a job is not assigned to this user it is currently being run by another process.

1. For Fail Step, choose **Increment Failure Count**

This is the first step of the failure loop. Should the job fail to execute for some reason, the background process will automatically set the job to this step and begin executing the steps in the failure loop.

1. For Extended Property, choose **WMX\_DISTRIBUTED\_GENERALIZATION**

This is the name of the extended property table that will store information about the process executing the job.

1. For **Run Service As**, choose a Workflow Manager user who will run the process on the machine.

When the Run Process button is clicked later, the tool will spin up a windows service that will run in the background on the machine. The user chosen here, will be the user that will run the service.

1. The Number of Process leave at 1

By default, when you choose the Role as parent, the value is automatically set to 1 and the text box is disabled so you cannot change the value. Because you will only create 1 parent job, there is no need to create multiple processes to run parent jobs.

1. Click OK.

## Configure Child Processes

As described above, many child jobs will be created each time you want to run generalization. Because there are many child jobs, these can be run across multiple machines as well as multiple cores on the machines. When determining how many processes to run on each machine, keep in mind how the machine will be used and do not specify more processes than the machine can run concurrently. It is recommended that if the machine has multiple cores that you run 1 less process than you have cores for example, 3 processes on a 4 core machine.

1. Connect to a machine(s) that will be used to execute the child jobs
2. If necessary, navigate to **C:\ProgramData\ESRI\Production\Desktop\DistributedGeneralizationWorkflows**
3. If necessary, open the **DistributedProcessManager.exe**
4. Click the **Add New Process** button

The Add New Process dialog is where you will specify what type of jobs you will run as well as how many processes you will run on the machine.

1. For Job Type, choose **CTM 50K Generalization – Child**
2. For Role, choose **Child**
3. For Default Status, choose **ReadyToWork**

This is the status that the job is automatically set to when it is first created. The processes that run the jobs only look for jobs in this status. The processes assume that if a job is not in this status it is currently being run by another process.

1. For Default User, choose **Automated Processing**

This is the group that the job will be assigned to by default. The processes that run the jobs, only look for jobs assigned to this group. The processes assume that if a job is not assigned to this group it is currently being run by another process.

1. For Fail Step, choose **Increment Failure Count**

This is the first step of the failure loop. Should the job fail to execute for some reason, the background process will automatically set the job to this step and begin executing the steps in the failure loop.

1. For Extended Property, choose **WMX\_DISTRIBUTED\_GENERALIZATION**.

This is the name of the extended property table that will store information about the process executing the job.

1. For **Run Service As**, choose a Workflow Manager user who will run the process on the machine when Run Process button is clicked later; the tool will spin up a windows service that will run in the background on the machine. This user you choose here, will be the user that will run the service.
2. For **Number of Process**, enter a number less than or equal to the number of processors on the machine.

Machines with multiple processors, it is recommended to use 1 less than the number or processors if the machine will only be used for running jobs. If the machine will be used to run the parent job or other operations while the child jobs are being processed it is recommended that you use fewer processes.

1. Click OK.
2. Repeat steps 1-13 in this section for all computers which will run child jobs.

# Running Jobs in the Distributed Environment

Now that you have ensured that your workflows are correct and you have configured your machines to run a specified number of jobs, you are ready to run the jobs in the distributed environment.

It is recommended that before running the distributed processes, that you delete all test job from the Workflow Manager Repository.

### Deleting Test Jobs

To delete existing jobs from Workflow Manager:

1. Run the **All Jobs** Query in Workflow Manager
2. Select the jobs in the job grid that you wish to delete
3. Right-click on the selected jobs
4. Choose **Delete**…
5. Ensure the job(s) you wish to delete are enabled
6. Click OK

## Starting the Distributed Processes

Once you have configured the processes that will be run on each machine, you are ready to run the processes. If you are using the machines for other purposes, it is recommended that you wait to start the processes until you are ready to create jobs and run them.

1. Connect to a machine that will be used to execute the jobs
2. If necessary, navigate to **C:\ProgramData\ESRI\Production\Desktop\DistributedGeneralizationWorkflows**
3. If necessary, open the **DistributedProcessManager.exe**
4. Ensure that the **Enabled** option is checked on for all of the Types of jobs you want to run on the machine
5. Click the **Run Process** button

This will fire off the configured processes on the machine.

1. Repeat steps 1-4 in this section for all computers that will run the jobs

## Create a Parent Job

When running the distributed processes, you need to create a parent job in Workflow Manager, the processes will take care of running the jobs. You can use Workflow Manager to monitor the status of these jobs.

1. Open Workflow Manager
2. Click **Create New Jobs(s)**
3. Choose the **CTM 50K Generalization - Parent** Job Type
4. Change the Assignment to the default user specified when you configured the parent job
5. Verify the Data Workspace. It should be populated with CTM 25K Data
6. Click OK to create the job

At this point, the machine configured to run the parent job should identify that the parent job exists and begin to execute it. After a few minutes, the parent job should create the child jobs and the child job processes should begin automatically running. You can check the status and process of the job execution by running and refreshing queries in Workflow Manager.

If something goes wrong and the parent or child job(s) do not automatically begin running, the Distributed Process Manager can be used to view errors and change configurations.

1. Connect to the machine(s) that should be executing the jobs.
2. If necessary, navigate to **C:\ProgramData\ESRI\Production\Desktop\DistributedGeneralizationWorkflows**.
3. If necessary, open the **DistributedProcessManager.exe**.
4. Click the Details button at the bottom of the dialog.

Information messages from the processes will now be returned to the tool. Look for any errors that seemed to be preventing the process from running.

## Stopping the Distributed Processes

Once the jobs have completed, you can stop the distributed processes on the machine. In order for changes to the configurations to take effect, you must stop and restart the processes.

1. Connect to a machine that will be used to execute the jobs
2. If necessary, navigate to **C:\ProgramData\ESRI\Production\Desktop\DistributedGeneralizationWorkflows**.
3. If necessary, open the **DistributedProcessManager.exe**
4. Select the jobs in the **Enabled** column which are to be stopped
5. Click the **Stop Process** button

This will stop the configured processes on the machine.

1. Repeat steps 1-4 in this section for all computers that will run the jobs