



# **Job Tracking for ArcGIS™ (JTX) Workflow Management Solution**

*An ESRI® White Paper • April 2003*

Copyright © 2003 ESRI  
All rights reserved.  
Printed in the United States of America.

The information contained in this document is the exclusive property of ESRI. This work is protected under United States copyright law and other international copyright treaties and conventions. No part of this work may be reproduced or transmitted in any form or by any means, electronic or mechanical, including photocopying and recording, or by any information storage or retrieval system, except as expressly permitted in writing by ESRI. All requests should be sent to Attention: Contracts Manager, ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

The information contained in this document is subject to change without notice.

#### **U.S. GOVERNMENT RESTRICTED/LIMITED RIGHTS**

Any software, documentation, and/or data delivered hereunder is subject to the terms of the License Agreement. In no event shall the U.S. Government acquire greater than RESTRICTED/LIMITED RIGHTS. At a minimum, use, duplication, or disclosure by the U.S. Government is subject to restrictions as set forth in FAR §52.227-14 Alternates I, II, and III (JUN 1987); FAR §52.227-19 (JUN 1987) and/or FAR §12.211/12.212 (Commercial Technical Data/Computer Software); and DFARS §252.227-7015 (NOV 1995) (Technical Data) and/or DFARS §227.7202 (Computer Software), as applicable. Contractor/Manufacturer is ESRI, 380 New York Street, Redlands, CA 92373-8100, USA.

ESRI, the ESRI globe logo, ArcGIS, ArcMap, ArcSDE, ArcCatalog, ArcObjects, ArcInfo, www.esri.com, and @esri.com are trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other trademarks, registered trademarks, or service marks of ESRI in the United States, the European Community, or certain other jurisdictions. Other companies and products mentioned herein are trademarks or registered trademarks of their respective trademark owners.

# Job Tracking for ArcGIS (JTX) Workflow Management Solution

## An ESRI White Paper

<b>Contents</b>	<b>Page</b>
Introduction.....	1
Key Features .....	2
Workflow Management .....	2
Version Management .....	2
Job Tracking.....	3
History Tracking/Change Detection .....	3
Feature-Level Metadata .....	3
Transactional Updates.....	4
Anatomy of a Job .....	5
What Is a Job?.....	5
Job Information.....	5
Users .....	6
Job Types .....	6
Steps.....	7
Layers.....	8
Version.....	8
Area of Interest .....	9
Map Document.....	9
Framework .....	10
Geodatabase Implementation.....	10
Integrated With ArcGIS.....	10
JTX Application.....	11
JTX Application Programming Interface.....	12
JTX Configuration Manager.....	12

<b>Contents</b>	<b>Page</b>
Working With JTX .....	12
Overview of the JTX Application.....	13
Sample Workflow for an Editing Job .....	14
Security .....	23
Summary .....	24

# Job Tracking for ArcGIS (JTX) Workflow Management Solution

## Introduction

Data production and maintenance projects, both large and small, have a vital need to have their project information (e.g., staff allocation, progress and status, changes to data) tracked as well as to have the work completed in the most efficient and timely manner possible. Being able to monitor these efforts in a way that does not encumber the production process and is tightly integrated with the software has traditionally been the challenge when it comes to job tracking.

By looking at an example of a typical data maintenance project, you can get a better understanding of the business challenge that ESRI® Job Tracking for ArcGIS™ (also referred to as JTX) directly addresses and the job tracking and management functionality such projects commonly require.

For example, a central organization hosts and maintains a database of the coastal regions of California. New source information (e.g., satellite imagery and field reports) becomes available for a particular geographic area. This new imagery shows the latest conditions of the shoreline after a large storm. The central database contains all of the thematic layers for the region (e.g., buildings, roads and other infrastructure, physical features). This data now needs to be updated per this new source information to produce up-to-date maps of the area. Typically a manager or supervisor would divide this work up by geographic areas (e.g., half-degree grid cells or discrete georeferenced image extents) and distribute the work among production staff. Editing staff will need to apply the changes to the data. Quality control (QC) staff will need to review these edits, identify any errors, and send them back to editors for correction. Once errors have been fixed and verified, the data is ready to be integrated into the master database. This process could take several days, weeks, or months and could involve editing the same layers, sometimes in the same area, as work being done by other staff. Therefore, long transactions must be leveraged, providing the ability to obtain a "snapshot" of the data to work with that is only integrated into the master database when all changes are complete. The manager will want to keep track of those who are working in each area, on which unit of work, how far along they are in their edits, any problems or issues encountered, how many QC cycles are necessary and, most important, when the work is completed.

Job Tracking for ArcGIS is an enterprise workflow management extension that provides an integration framework for ArcGIS multiuser geodatabase environments. It simplifies many aspects of job management and tracking and streamlines the workflow, resulting in significant time savings for any implementation. Job Tracking for ArcGIS provides tools for allocating resources and tracking the status and progress of jobs. A detailed history of job actions is automatically recorded for each job to give managers a play-by-play of how the job was completed. This information can be supplemented with comments and notes

to provide even richer job documentation. Job Tracking for ArcGIS handles complex geodatabase tasks behind the scenes by creating and managing versions. A comprehensive transaction management infrastructure is provided to support historical tracking of feature edits and feature metadata. And developers can utilize JTX integration points and the JTX application programming interface (API) to leverage existing or off-the-shelf solutions or embed JTX functionality in new applications.

This paper introduces the key features of Job Tracking for ArcGIS. It discusses what is meant by the term "job" in the context of JTX and the intricacies of a job. It presents the JTX framework and explores the key integration points with ArcGIS as well as topics of configuration and customization. An example workflow scenario illustrates the main features covered in this paper and ties together key concepts. For a complete discussion of how to configure and use Job Tracking for ArcGIS, refer to the documentation and help system provided with the extension.

## Key Features

### Workflow Management

One of the primary goals of Job Tracking for ArcGIS is to help data maintenance and mapping engineers do their jobs more efficiently. One way to accomplish this is by providing an easy way to search through all existing jobs to find the job (or jobs) they are working on. Job Tracking for ArcGIS also provides a list of tasks involved in completing the unit of work at hand. This takes the guesswork out of what the job entails and keeps track of completed tasks, which is particularly useful for engineers who have several jobs assigned to them and frequently switch between jobs over the duration of their workday. Furthermore, Job Tracking for ArcGIS automates these steps as much as possible, effectively reducing the number of repetitive setup tasks the user must perform to begin working such as launching the program required to accomplish a task, loading the layers, assigning symbology, and zooming to the area of interest (AOI) for a job.

Another chief goal of Job Tracking for ArcGIS is to help managers assign these units of work in an efficient and orderly manner. They have the tools necessary to provide quick updates of who is working on what as well as monitor the progress of the work by user or by unit of work.

### Version Management

A geodatabase version can be created for, and associated with, a job for the duration of that job. Not all jobs require a new version to be created. However, for any job that requires edits be made to the database, a version *must* be created. Any edits are then performed on the version, not on the base data. This allows engineers to take advantage of long transactions to avoid interfering with the work of other users who may be editing the same layers in the same geographic area. They can also use the ArcGIS out-of-the-box tools for detecting and resolving conflicts with the master database (as well as edits from other jobs) and for integrating their edits with that base data. The geodatabase versioning model is also the primary mechanism JTX uses to perform detailed tracking of feature edits and historical archiving of those modifications.

For a more detailed discussion on versioning and the geodatabase, refer to the book *Building a Geodatabase*.

## Job Tracking

Job Tracking for ArcGIS provides the ability to track the user's interaction with a job throughout the existence of the job. Any action that is performed on the job (e.g., modifying any of the job's properties, changing the state of the job, executing a step) is captured and logged in the JTX administration database. This allows a manager or user to see a successive listing of what has been done to the job since its creation. A time-stamped list of comments can also be logged for each job to record corollary information (e.g., issues, clarifications, notes for QC team). This chronological registry of job-related information allows management to track back to a particular period of time and help identify problems or justify certain decisions.

Another key aspect of job tracking is the use of an AOI to delineate the working extent of a job. This helps managers track who is working in which area and determine the status and distribution of those jobs with a quick visual inspection.

## History Tracking/Change Detection

Job Tracking for ArcGIS delivers rich support for the capture and storage of feature-level transactions. These transactions are ascertained when the job version is reconciled and changes are posted back to the master database. Thus, only permanent edits to the data are captured and recorded, rather than intermediate edits that may be discarded over the course of job or iterative edits that may be part of several edits performed to achieve some end result.

The JTX transaction record stores the type of change made to the feature (whether it was added, modified, or deleted) as well as which individual attributes were edited along with their before and after values. In the case of spatial edits, both the before and after geometries are stored. A date/time stamp, the identifier of the geodatabase table the feature resides in, and the Global Feature Identifier (GFID) are also captured as part of the transaction information. The GFID links transactions back to features in the database. The transactions are recorded in an Open GIS Consortium (OGC)-based Extensible Markup Language (XML) format that leverages industry-standard models for features.

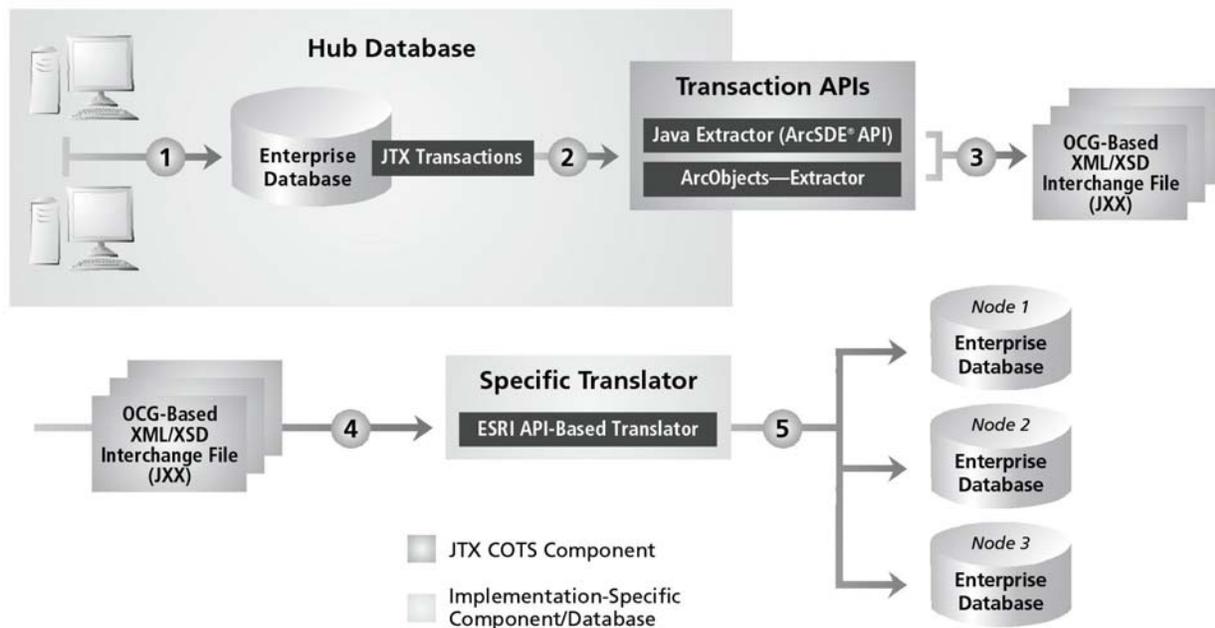
A JTX transaction table exists in the database for the sole purpose of warehousing transactions, and it is there that the properties and XML of each transaction are stored. It can then be queried later by GFID, Job ID, date, or date range to retrieve transactions meeting the desired criteria. These transactions can then be used to apply transactional updates to remote databases (see Transactional Updates) or for historical interrogation by some of the transaction inspection tools provided by JTX.

## Feature-Level Metadata

JTX also supports feature-level metadata tracking. JTX provides, through its API and the JTX ArcMap™ extension, the ability to store additional information about the feature at large or about particular field-level updates as edits occur in the client application. Some examples of such information may be the source of the edit or the accuracy of the geometry. The content of the additional metadata information is entirely determined and supplied by the user. JTX provides integration points for this information to be tracked as edits are made and, by storing this information in the ArcMap Map Document (MXD) of the job, this information is associated with the job for the life of the job. This metadata is embedded in the transaction XML when transactions are stored for that job (at the time its edits are posted to the database).

## Transactional Updates

Feature level transactions (see History Tracking/Change Detection) can be used for replicating data edits from one database, in implementations where databases must be kept in sync. The purpose of this process is to enable the generation and transmission of all editing transactions performed against the centralized database in a platform independent, nonproprietary data format. Once the target environment and database model are established, any programming language and database API can then be used to integrate the transactions.



As discussed earlier, feature-level transactions are recorded each time the user reconciles and posts the changes made to the job version (1) as detailed in History Tracking/Change Detection. The administrator would then request that an extraction file be built (2). The extraction request can be predicated by either a time interval (e.g., create a file containing all transactions since my last update) or a specific subset of layers (e.g., create a file containing all transactions for the Roads and Trails layers since my last update). The result of this request is an XML-based interchange file (.jxx) containing the XML for all of the transaction records that satisfied the query. This XML file is based on the OGC recommendations for XML storage of geographic feature data sets and provides a vendor-neutral format for the interchange of information. Depending on the desired transport mechanism, file or direct network, the .jxx file would either be written or streamed to the client (3). The XML-based interchange file can then be sent either across a network or any other suitable media to the target environment. For each target database, it would be necessary to develop a custom translator for that specific relational database management system (RDBMS) implementation, data model, and technology.

This custom translator would be able to read the XML interchange file (4) and perform any necessary data attribute mappings to match to the target database. The features would then be written into the target database (5) using the most appropriate means. These custom translators could be written in any language that has the ability to read the XML interchange file and write the features into the database.

## Anatomy of a Job

### What Is a Job?

A job is a unit of work to be performed by one or more people on a particular set of data. One example of a job might be a set of updates that needs to be made to data in a particular geographic area based on some source information (as described in Introduction). A job may be a data extraction effort that creates new data in the database. Or it may be a series of maps that needs to be created for a particular area, or set of thematic layers, at a specific scale. Each one of these jobs most likely involves a different set of tasks that must be accomplished to complete the unit of work. Likewise, different applications and tools may be required for each, and the set of data that the user works with (i.e., layers and tables) may be unique for each type of job.

The following sections will describe the different characteristics that define a job.

### Job Information

Each job that is created maintains a set of properties that describes the job and tracks the state of a job at any time in its life span.

### *Properties*

- **Job ID**—The unique identifier for the job. This is automatically generated when a job is created and provides a mechanism for the job to be distinguished from other jobs and tracked by managers. (Required)
- **Job Name**—The full name of the job, which is composed of the Job ID, Job Type, and Due Date. (Required)
- **Job Status**—Describes the current state of the job. Valid statuses are Created, Open, Assigned, Closed, and Queued. Typically a job will be assigned a status of Queued when it has been assigned to a particular user group (see Users) but has not been assigned to a particular user. In this scenario, upon completing one job, a user can go to the group's queue and begin work on the next available job. This can save the manager from having to assign every job to a specific user when any engineer with edit privileges can work that job. (Required)
- **Job Type**—Please see Job Types section on page 6.
- **Due Date**—The date by which the job must be completed. (Required)
- **Priority**—A ranking of 1 to 5 indicating a job's importance relative to other jobs.
- **Project Name**—Name of the project to which jobs can be grouped for organizational and tracking purposes. This allows the user to specify the project to which the job belongs.

***Tracking and Documentation***

- **Description**—A phrase that describes the job in plain terms. Job names are often kept short for display purposes and can be cryptic; the description can help clarify the intent of the job.
- **Assigned to**—The user or user group to which the job is currently assigned.
- **Created by**—Identifies the user who created the job.
- **History**—Tracks user interaction with the job from a workflow perspective. Each time someone interacts with the job, the action is recorded in the JTX administration database with a time stamp and user identifier.
- **Comments**—Time-stamped pieces of information or correspondence that can be added at any time to supplement the automatic history tracking. These comments are maintained by JTX for the life of the job.
- **Notes**—Maintains a single repository of notes and miscellaneous communications for each job.

**Users** At any given time in the life cycle of the job, the job is assigned to a user. This identifies the user who has access to work that job. When the job is assigned to an individual user, only that user can interact with that job (e.g., execute steps, update properties, reassign the job).

Each user belongs to a user group. Jobs can initially be assigned to a user group rather than to an individual user. Assignment to a user group places the job in a queue. This provides a pool of jobs that users belonging to that user group can pull from when they are ready to begin work on a new job. For example, there are 12 data engineers who all have the same editor privileges. The manager can divide a geographic area into chunks by defining a set of jobs and specifying the area of interest for each job. The manager would then assign each job to the editor group. When the users of that user group are finished with their job (or are waiting for another job to be reviewed or approved), they can simply search for the jobs assigned to their group, reassign a job to themselves, and begin working. Jobs cannot be modified when assigned to a queue. Only jobs assigned to individual users may be modified or processed.

User groups also define what application privileges that user has. A typical implementation might have a manager group, an editor group, and a QC group. Each group would be assigned privileges based on the role it plays in the maintenance and execution of the job. The manager may have the ability to create jobs and close jobs, but an editor or QC user would not.

**Job Types** Part of configuring Job Tracking for ArcGIS for a particular implementation involves the definition of job types. A job type is a logical grouping of steps and layers required to achieve some common goal.

For example, you may define a job type for creating 1:10,000-scale maps, which would require a set or group of tasks (or steps) needed to accomplish the creation of that

1:10,000-scale map (e.g., define AOI, create map [i.e., title, surround elements, and other graphics]), verify/QC map against map specification, and print map. Steps are further discussed in more detail in the following section. This type of job would also require that a standard group of layers be displayed. These layers would contain features that are graphically meaningful at a scale of 1:10,000 (e.g., buildings, residential streets).

We may also need to define a job type for creating 1:250,000-scale maps. The steps needed to perform this type of job may be the same as for creating 1:10,000-scale maps, but the layers (and symbology of those layers), of course, would differ from the larger-scale product. Displaying county boundaries, major highways, and cities would probably make more sense at that scale than buildings and residential roads.

Furthermore, we may want to define some job types for updating features in the layers we are using to create the map products. We could specify a job type to update street and parcel boundary data and another job type to update major hydrographic layers. These editing job types would have a different set of steps than the mapping jobs above, but they would all probably have steps in common with each other. Editing jobs would typically have different layers from mapping jobs and other editing jobs. The symbology of those layers would differ as well, since high-quality cartographic rendering is not necessary for performing editing tasks.

Now the requirement may be to create 1:10,000-scale maps for the entire state of Texas. Typically, a manager would create several discrete jobs, each with the same job type (create a 1:10,000-scale map). Each job type covers a different portion of Texas, and each would be assigned to a different user.

Job types are a primary mechanism for streamlining the workflow of jobs. The steps defined for a job serve as an automatic overview of the tasks ahead as well as a checklist to ensure no step is missed. Also, the layers required to do the work, as well as the symbology of those layers, have been predefined. When the user enters ArcMap to create a map, for example, the layers are loaded automatically, saving the user valuable time.

**Steps** As described in the section above, each job type has an ordered set of steps associated with it. These steps can be as detailed or as general as necessary.

A set of steps for an editing or data maintenance job might be

- Create an AOI.
- Create a version of the database.
- Edit data.
- QC data.
- Post data to master database.

A set of steps for a mapping job might be

- Create an AOI.
- Create a map.
- QC map against product specification.
- Print map to hard copy.

Different job types can also have the same steps associated with them (e.g., varying scale mapping job types would likely have the same set of required steps).

Steps can be configured to control the order in which they are executed with respect to each other and restrict users from performing steps out of order or reversing a step. Steps are stored in the JTX administration database in a specified order. Each step maintains an attribute that lets the user or client application know whether it can be undone or not. For example, a manager may need to refine the area of interest boundaries for a job. That step would be defined to allow reexecution. However, the "Create a version" step should only be executed once for a job. This step would be configured to prohibit users from undoing that step. Similarly, certain steps can be set up to be checked off automatically after a single execution (e.g., "Create a version" step); however, other steps may require several days and several sessions to complete such as an "Edit data" or "Create a map" step. Those steps would be configured to require the user to mark the step as complete.

An executable can be associated with each step in that job type. Examples of executables are ArcMap, ArcCatalog™, Microsoft Excel, Crystal Reports, or any custom executable. As a corollary to that, an argument string can be configured for passing information to the executable. For example, the user may want a step to launch a custom executable that takes certain arguments. He/She can specify the arguments for that executable when the steps are being defined in the configuration phase of the implementation.

**Layers** Any layer in the master database can be registered with JTX. This effectively assigns a unique identifier to the layer in the JTX administration database. Detailed layer properties (e.g., symbology, definition queries, label settings) can be configured in advance for all JTX registered layers and stored in the JTX administration tables. Those properties become the default settings for any job that uses that layer. This saves users valuable time setting up their ArcMap table of contents and layer properties each time a new job is created. As mentioned above, layers are associated to a job by its job type. Each job type has a set of default layers assigned to it.

Furthermore, layers that are related topologically or thematically (e.g., transportation data) or are simply part of some "catch-all" category to be dealt with collectively (e.g., all background or reference data) can be assigned to a layer group. When layers belonging to a group are displayed in ArcMap, all layers in the group will be represented by one entry in the table of contents. When the group is reordered or its visibility is toggled, all layers belonging to the group are affected accordingly.

**Version** When a new job is created that requires editing the database, one of the steps for an editing job type should be to create a new version of the database. The JTX framework

provides a step executable for creating job versions. That version exists for the lifetime of that job, and all edits for that job will occur against that version and not against the master database. Version names are based on the Job ID to easily associate versions in the database with their corresponding JTX jobs. Having a unique version for each job allows multiple users to concurrently edit the same layers. When they have completed the job, their edits can be posted back to the parent version. Each JTX job type specifies the parent version to be used in creating versions for jobs of that type. The parent version can be any existing version in the database. In most cases the parent version will be the DEFAULT version of the database.

### Area of Interest

The AOI is the geographic extent of a job. This helps to confine the unit of work to a geographic area. It also helps managers and users track when edits are in progress to avoid conflicts and prioritize and coordinate activity between many users and user groups. The AOIs for all jobs are stored in a special feature class in the database that is maintained by JTX.

The Job Tracking for ArcGIS framework provides a standard tool for specifying the AOI for a job. Therefore, any step defined for a given job type that involves creating the area of interest will use this tool as its executable. The area of interest can be derived in one of the following three ways:

- Selecting a feature from a predefined product grid. The layer containing the product grid can be configured to be the predefined layer for the area of interest. This indicates to the AOI tool that the user will be selecting one or more features (grids) from that layer, whose extents will comprise the area of interest for the job.
- Selecting the extents of another source (e.g., georeferenced imagery, shapefiles).
- Specifying a user-defined extent.

When a new AOI is being defined, the AOIs of any existing jobs are also displayed. This allows the user to identify which areas have already been assigned and helps to avoid defining extents that overlap existing jobs. Job Tracking for ArcGIS can also be configured to load certain other layers into the AOI tool by default, serving as background or reference information for delineating the job.

When a job step is executed that necessitates the use of ArcMap, the job's AOI is detected by the JTX ArcMap extension and used to zoom the current map display to the extents of that AOI.

### Map Document

The Job Tracking for ArcGIS framework provides a step executable that launches ArcMap and automatically creates an MXD for the current job, if one does not already exist. Once created, that MXD is maintained by that job for the lifetime of the job.

The MXD keeps track of the map and data-specific characteristics of that job: which layers were loaded for that job (as determined by the job type), which version of the database those layers are pointing to, and the extent (or AOI) the user is concentrating on

as well as any additional ArcMap session information (e.g., visible toolbars, other extensions loaded).

The MXD is stored in the database to provide greater portability of work. This allows the Job Tracking for ArcGIS user to perform tasks on any machine on the network. When the user begins or resumes work on a job, for a step that requires the use of ArcMap, its MXD is retrieved from the database and stored locally for that session. When ArcMap is launched, the MXD is automatically loaded into that ArcMap session. Any changes to the MXD are saved and posted back to the database. The name of the MXD is based on the Job ID so users can easily associate the MXD with its corresponding job.

## Framework

### Geodatabase Implementation

Job Tracking for ArcGIS is implemented on top of ArcSDE® geodatabase technology. There are certain key assumptions that Job Tracking for ArcGIS makes that are essential to its successful implementation for any project or organization.

- Data used in each job resides in a geodatabase and, if it is to be edited, must be registered as versioned. Many features of Job Tracking for ArcGIS hinge on the notion of data being versioned for edit job types. Such features include history tracking, feature-level metadata maintenance, posting edits to the master database, and loading and displaying editable layers in preparation for editing in ArcMap.
- Each geodatabase feature class that is registered as a JTX layer should contain a field for storing a GFID. This number should be unique across all features in the database, not just within a particular layer. JTX does *not* maintain this GFID but relies heavily on it for recording feature-level transactions and metadata. How that field is maintained is entirely up to the JTX customer.

Job Tracking for ArcGIS is a highly configurable and database-driven application. The JTX administration database tables are the foundation of any JTX implementation. A set of JTX administration tables exists for each geodatabase in the implementation. These tables store all configurable information such as job types, job type to JTX layer mapping, JTX layer symbology, job steps, JTX users, user groups, and privileges. They also store the specification and properties of each job that is created as well as the area of interest extent and MXD of the job. Feature-level history (transaction) and feature metadata information are logged there as well.

### Integrated With ArcGIS

Job Tracking for ArcGIS requires significant integration with ArcGIS applications such as ArcMap and ArcCatalog. JTX provides a registered extension to ArcMap as well as ArcMap custom tools and a toolbar to host them.

- **JTX ArcMap Extension**—This extension provides integration points to be able to perform many of the ArcMap software-specific JTX operations. The Job Tracking for ArcGIS extension, once made aware of the active job, knows how to automatically load data from the correct (job) version, render the job layers on the ArcMap canvas, zoom to the area of interest, and expose the JTX toolbar. The Job

Tracking Extension for GIS is also the mechanism for specifying and storing feature-level metadata.

- **JTX Toolbar**—This toolbar hosts the custom ArcMap commands detailed below.
- **Add Data Command**— This command allows you to add data to your ArcMap Table of Contents from the version of the current job.
- **ArcMap Reconcile Command**— This is the default Reconcile button that comes out of the box with ArcMap. This button is exposed from the JTX toolbar for convenience because it is used in conjunction with the JTX Post command (versions must be reconciled prior to being posted).
- **Custom Post Command**— Users can post their edits at any time throughout the life span of a job using the custom JTX Post command. The user experience is the same as if he/she had used the out-of-the-box ArcMap Post command but, in actuality, before any data is integrated with the parent version, Job Tracking for ArcGIS performs difference detection on the job version to identify all differences between it and the version it was derived from (in most cases, the default version). Any additions, modifications, or deletions are summarized in an XML block and written to the database as a transaction.

In addition, as the custom post operation iterates through the changes of each feature, it will retrieve any feature-level metadata that has been created and stored for that feature (in the MXD of the job). That metadata information is then inserted into the appropriate location in the transaction XML before it is stored in the JTX Transaction table.

- **Transaction Inspector Tool**— The JTX Transaction Inspector is a tool that allows the user to interactively identify a feature on the map and view the lineage of changes for that feature. When a feature is selected on the map, the tool interrogates the JTX Transaction table and retrieves all the transactions having the same GFID as the feature identified. It utilizes a tree-view structure to allow the user to step back through each of the past feature states. If a geometry change has occurred, users will be able to see the past geometries overlaid onto their ArcMap session for visual inspection.
- **Job Transaction Loader Command**— This command prompts users to enter a Job ID and will add a graphical layer to their map showing all the features added, modified, or deleted in the specified job. This can be useful to get an overall assessment of the quantity of edits made in a job, the type of edits made (e.g., adds, modifies, deletes), and in which area those edits were concentrated for more targeted quality assurance (QA)/quality control efforts.

## JTX Application

Job Tracking for ArcGIS provides a default, stand-alone application for managing jobs. From this interface jobs can be created, assigned, queried, and inventoried and worked. All job properties can be managed through the application from which all steps are executed, related applications are launched, and so forth. The features of this application

are illustrated in more detail in the section Working With JTX (below). It is important to note that little application logic resides in this default JTX application. It was developed on top of the JTX API (described in more detail below).

### JTX Application Programming Interface

The JTX API is the development platform for Job Tracking for ArcGIS. The JTX business objects expose the full range of functionality necessary for creating, managing, and working with jobs. The JTX Objects were built using Microsoft's Component Object Model (COM) technology and extends ESRI's ArcObjects™ framework. ArcObjects is used to perform all database access functions for platform independence and legacy compliance with database access protocols. It is through ArcObjects that you can create and maintain the MXDs for your job, create and manage job versions, and provide custom functionality in ArcMap (via the Job Tracking for ArcGIS extension and toolbar). This API allows developers to customize the default JTX application or write an entirely new application to manage jobs.

### JTX Configuration Manager

Because it is so configurable and database driven, Job Tracking for ArcGIS also provides a user-friendly stand-alone application for managing the JTX administration database. The JTX administration database tables are the foundation of any JTX implementation. These tables host all the information about your JTX implementation: users, privileges, steps, job types, layer definitions, and so forth. A full suite of tools is provided for creating, updating, deleting, and cross-referencing information for these tables, effectively customizing JTX to meet the needs of your project.

The Configuration Manager also provides a mechanism for transporting your configuration setup from one JTX server to another, if your project calls for multiple databases with essentially the same JTX configuration values. Any JTX administration database can be exported to a JTX Configuration Exchange XML file. This file can then be imported into any other JTX administration database.

Jobs and their elements can also be managed through this application. The Configuration Manager provides tools for reassigning jobs to a different user or user group (regardless of to whom it is currently assigned) as well as deleting jobs and their associated information. You can even selectively delete certain job elements such as the geodatabase version associated with the job or the job's MXD, while preserving the job itself. It is important to mention that the Configuration Manager is intended for use by a manager or database administrator and typically will not be accessible to other JTX users.

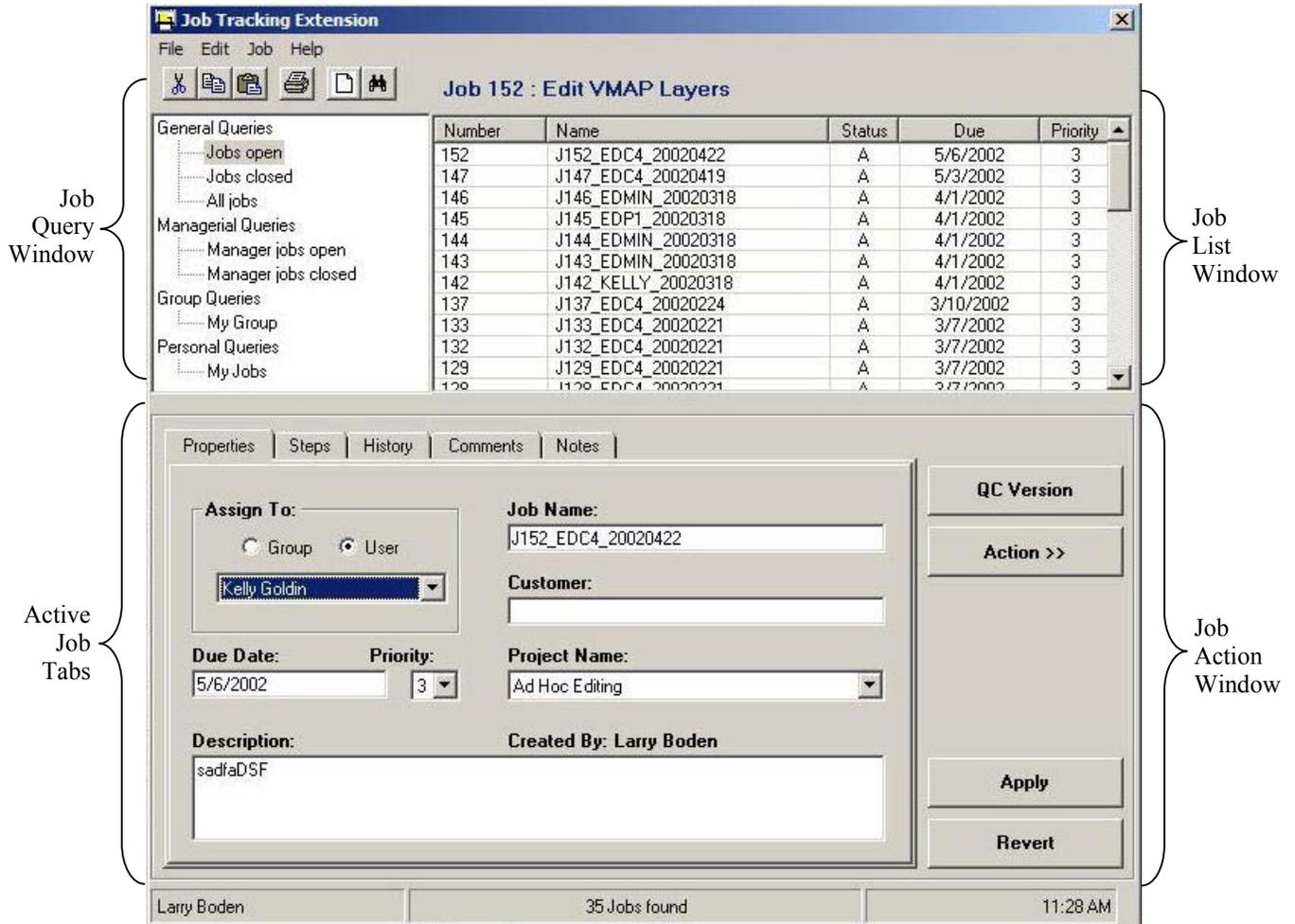
The application was developed on top of the JTX API so all interaction with the JTX tables occurs through the JTX business objects.

### Working With JTX

This section will take you through an editing job from start to finish. It will demonstrate the concepts above and familiarize you with the user experience of JTX. It assumes that JTX has already been configured to support the job type specified, the layers for that job type, the users, and any other setup and administrative tasks.

J-9005

**Overview of the JTX Application** The JTX application provides an interface where jobs can be created, assigned, queried, and inventoried.



## Sample Workflow for an Editing Job

### *Create a Job*

To create a job, click the Create Job tool or select Create Job from the Jobs menu. Select the appropriate Job Type from the list, and click Create Job. The job is created and appears in the job list window.



### *Assign the Job*

To assign a job, select either the Group or User radio button. Then select the user group or user name from the list, and click Apply. Job steps will only be enabled for the user assigned to the job. Jobs assigned to a user group can be found by selecting My Group from the job query window. Jobs assigned to a user can be found by selecting My Jobs from the job query window.



### *Set Job Properties*

Some properties, such as Job ID and Job Name, are automatically set when a job is created. It is up to the user to set the other properties (e.g., Due Date, Priority). Click Apply to commit the changes.

J-9005

**Documenting the Job**

Job History is an important part of job tracking and the one form of job documentation that takes place automatically. As you interact with the interface by creating the job, assigning the job, and stepping through the tasks, you will see that each action is logged by the system and displayed under the History tab.

Time	User	Description
02/06/2002 3:14:00 PM	Kelly Goldin	Created Job kellym
02/06/2002 3:14:36 PM	Kelly Goldin	Assigned Job to Kelly Goldin by
02/06/2002 3:16:30 PM	Kelly Goldin	Assigned Job to Kelly Goldin by
02/06/2002 3:16:52 PM	Kelly Goldin	Started executing step AOI
02/06/2002 3:22:07 PM	Kelly Goldin	Started executing step Create V
02/06/2002 3:22:29 PM	Kelly Goldin	Started executing step Edit Vers

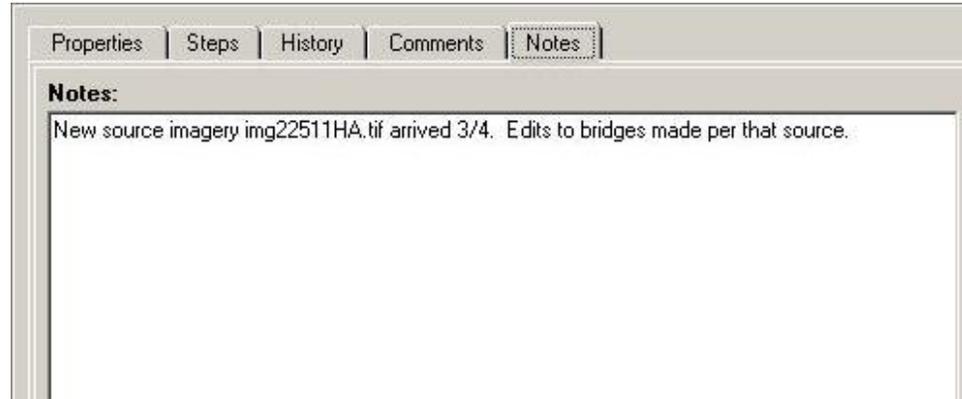
The Comments tab displays job comments chronologically and provides tools for adding new comments.

**New Comment:**

[Text Area]

Time	User	Description
02/20/2002 4:11:50 PM	Kelly Goldin	Make sure to double check all NAM values for nulls
02/20/2002 4:12:22 PM	Kelly Goldin	There are no buildings in this area, verify with ATPD

The Notes tab shows the job's notes log.

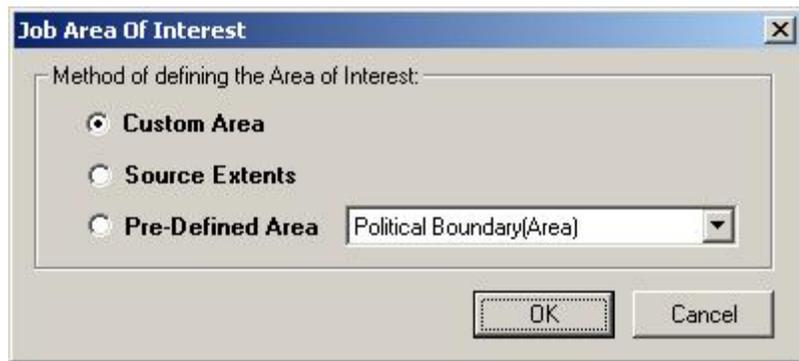


***Execute the Steps of the Job***

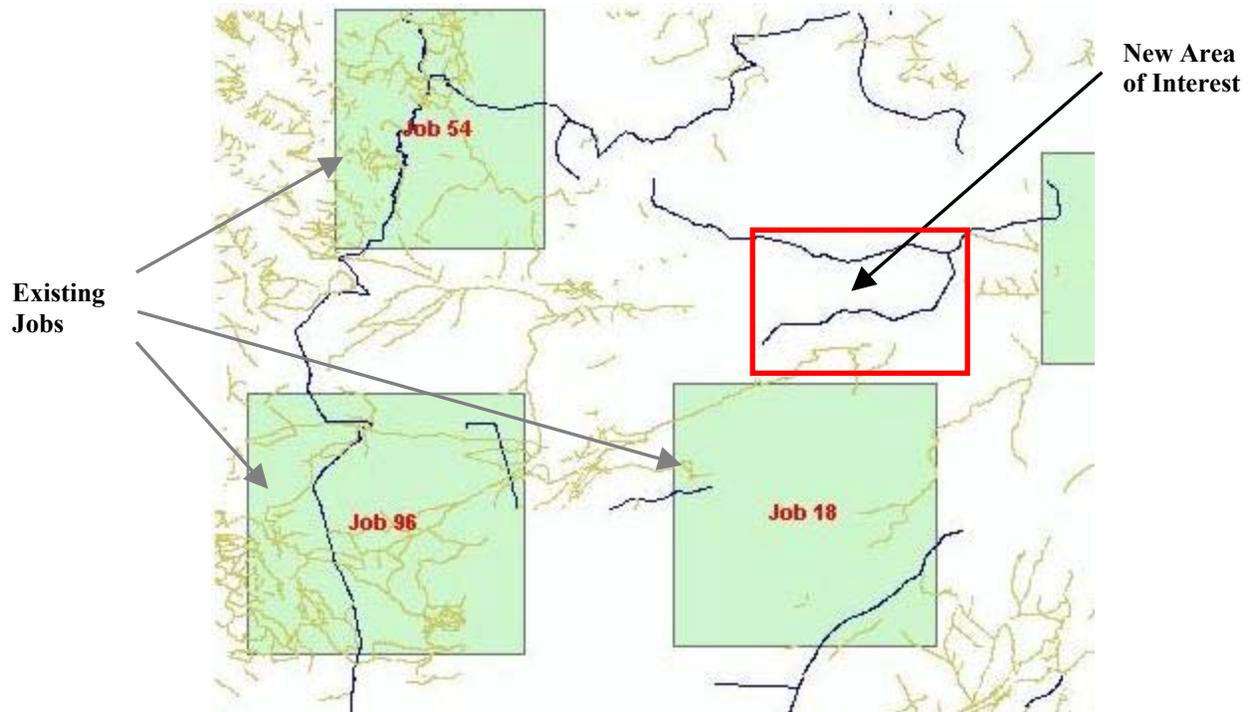
Once the job has been created and assigned, the user to whom the job has been assigned can begin working through the steps of the job. The Steps tab presents the steps configured for the particular job type. The button at the top of the job action window always displays the name of the next step to be executed. This allows the user to execute the step without having to navigate to the Steps tab.

***Step 1: Define Area of Interest***

For this job, the first step is to define an AOI. Most job types will list this step first because the AOI extent is a valuable tool for job tracking. Executing this step will launch the AOI tool, which presents three different methods of defining an AOI.



By choosing Custom Area, you are presented with a dialog displaying the AOI layer (which shows the AOIs for any existing jobs) and any reference layers that have been specifically configured for display in the AOI tool.



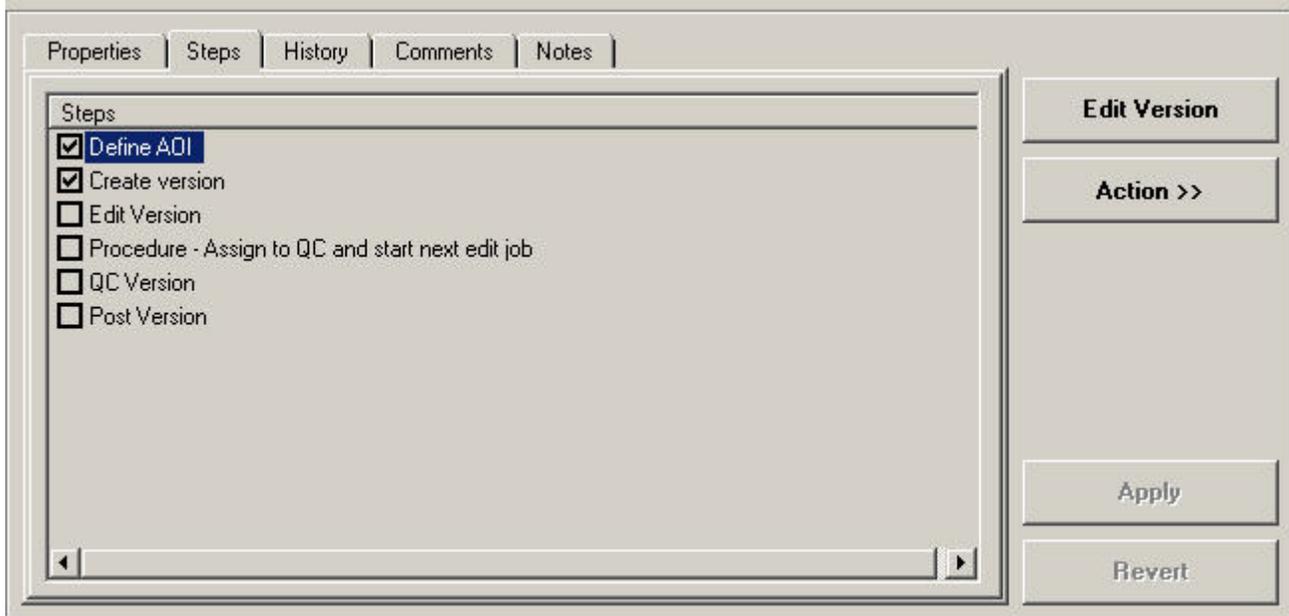
You can click the Add Data button on the toolbar to add other layers for reference. Pan-Zoom tools help navigate to the correct area. Using the draw tool, you can delineate the area of interest by drawing a rectangle on the map for the AOI extent. If you know the bounding coordinates of the AOI, you can manually key in the extents with the tool provided.

Once this step is complete, JTX automatically checks the step off the list and identifies the next step to be executed.

*Step 2: Create Version*

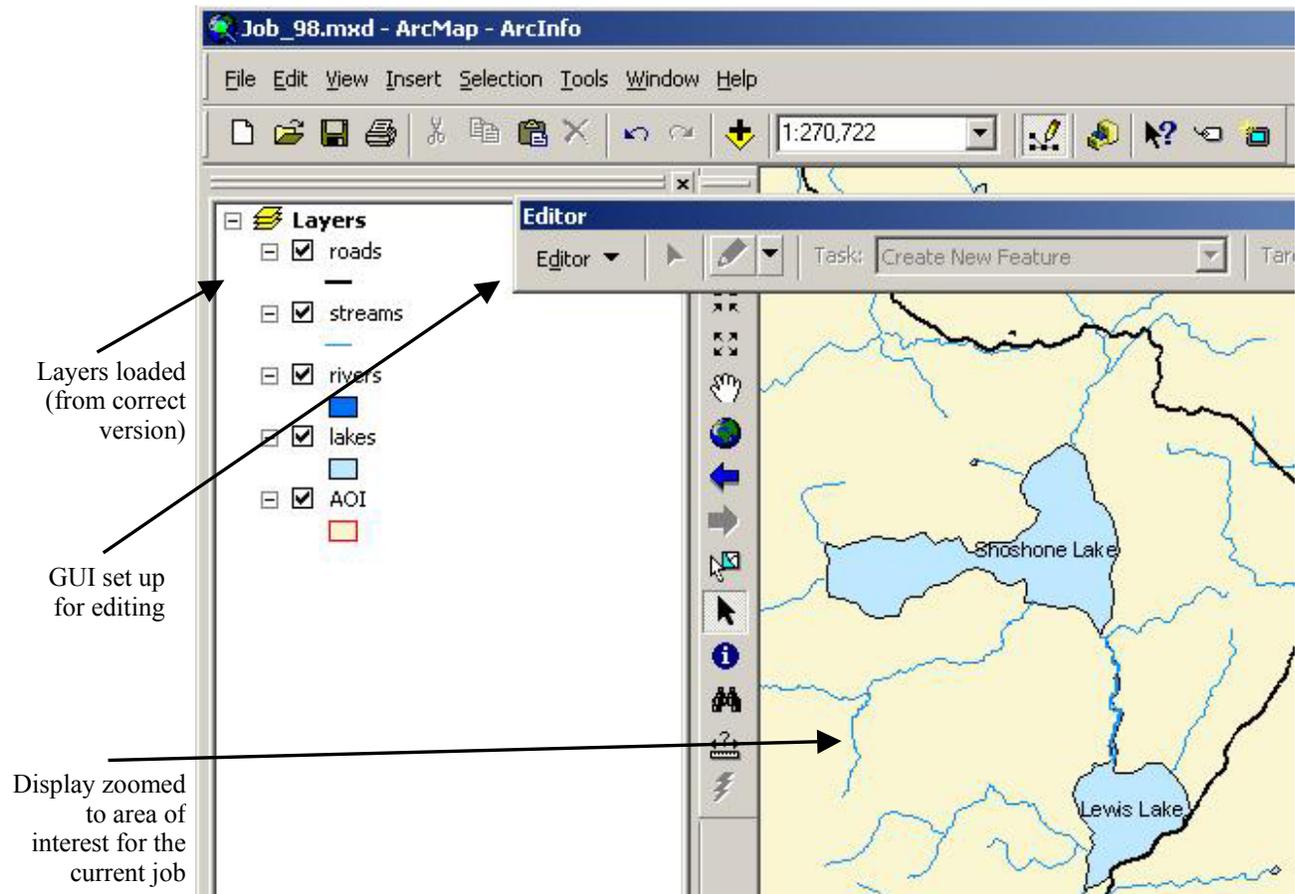
This step requires no user interaction except to execute the step. This calls an executable that will create a new version of the database for this job, which is named after the job for easy reference while working with that version.

By now the first two steps should be checked off the list.



J-9005

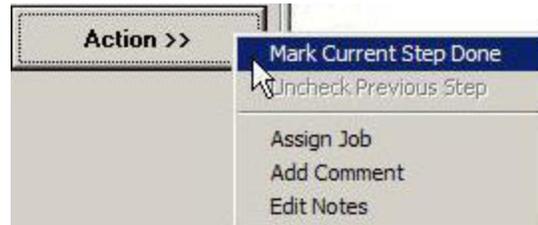
*Step 3: Edit Version* When this step is executed, the ArcMap application starts up. Layers are loaded using the job version as the source and the default symbology that has been configured for each layer. The map is zoomed to the AOI of the job. JTX also makes sure the Editor toolbar is visible (which was configured for this editing job type).



Within ArcMap you can use any ArcMap or custom tools provided to edit the layers of your job. This includes adding, modifying, reshaping, and attributing features within the layers.

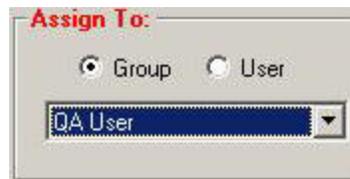
When you exit ArcMap, the JTX application appears again. At this time, you will notice that the Edit Version step has not been checked off. This is because this step has been configured to require manual checkoff. At this point you can shut down your computer for the day, come back tomorrow, launch JTX, execute this step again, and continue editing where you left off.

In this case you are finished editing and will use the Action drop-down menu to mark the step as done.



#### *Step 4: Assign to QC*

This step was specifically configured to ensure that editors assign their job to the QC user group or to a particular QC analyst for review of their data. Navigate to the Properties tab, change the currently assigned user or group, then click Action to mark the step as done.



#### *Step 5: QC Version*

To execute this step, you must be logged in as the QC staff member to whom the job is assigned. Like the Edit Version step, when this step is executed, the ArcMap application starts up, layers are loaded, and the map is zoomed out to the AOI for the job. This time, however, a custom QC toolbar appears (rather than the Editor toolbar), which sets up the interface for performing a full review of the edits performed.

If there are any edits, then the job would need to be assigned back to the editor. The editor would then uncheck both the QC Version step and the Edit Version step (using the Action drop-down menu) and reexecute the Edit Version step to perform corrections to the data. When all edits are complete, the job will need to be routed to the QC user again. This cycle continues until no errors are found in the data.

#### *Step 6: Post the Job*

When the edit/QC cycle is complete, the data is ready to be integrated with the master database. When this step is executed, ArcMap is launched and set up the same way as in the Edit Version and QC Version steps. Start an Edit session and use the reconcile button on the JTX toolbar to reconcile the data in this version with the parent version. During the reconciliation process, if conflicts between the parent version and the version being posted for that job exist, they are identified and displayed. You must then select between the edited version, the default version, and the preedit version for conflicts.

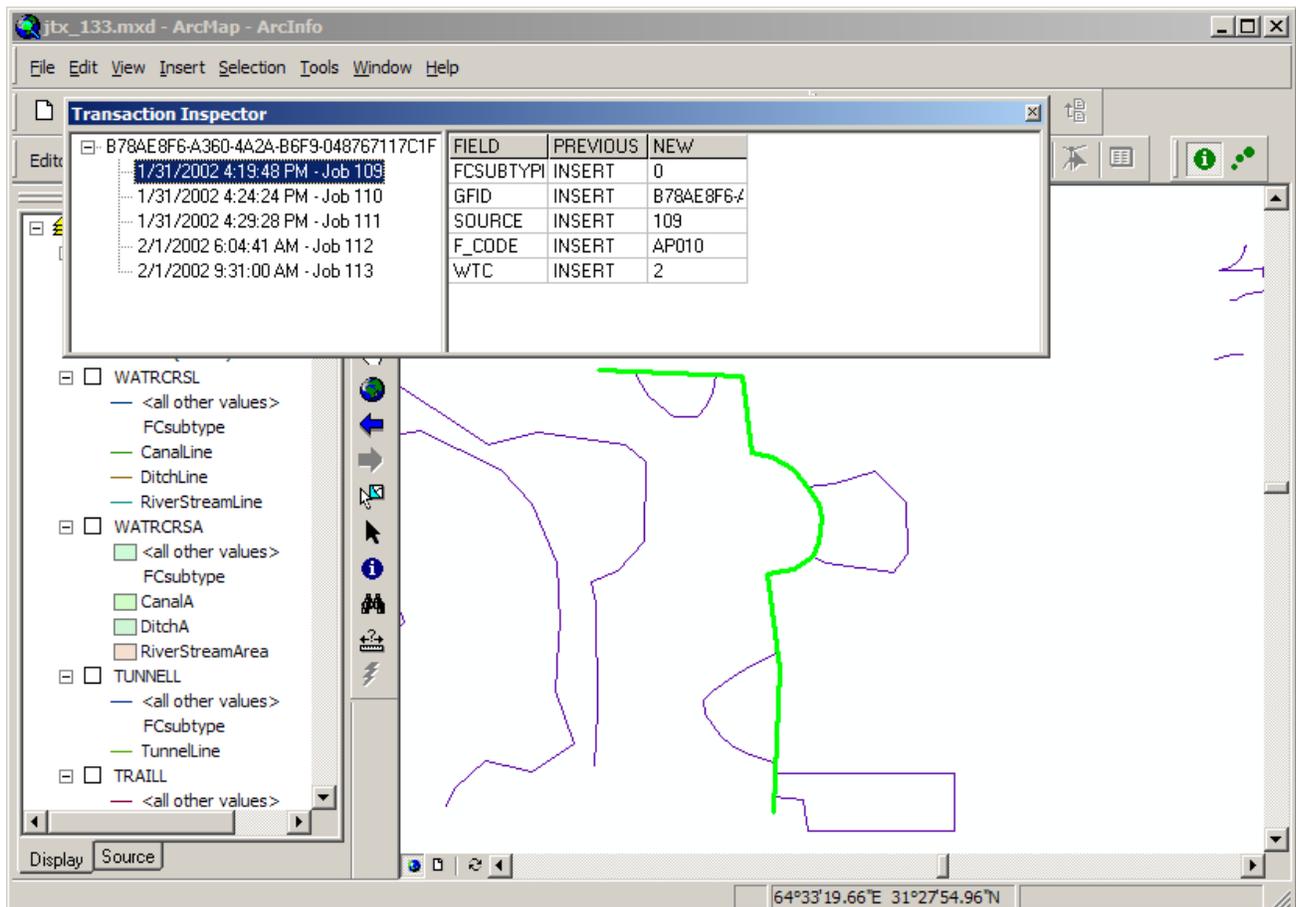
Once conflicts are reconciled, the data can be posted to the database using the custom Post button on the JTX toolbar. Remember that before data is physically integrated with the parent version, comprehensive change detection is performed on all data in the job version, and any changes found are identified as transactions and written to the JTX Transaction table in the database. When posting is complete, exit ArcMap and mark the step as done in the JTX application.

When the last step of a job is marked as complete, select the Close Job Option from the Action menu. The job will now appear in the job list of Jobs closed and will no longer appear in the list of Jobs open.

### *Inspecting Feature History*

#### *Using the Transaction Inspector*

The first step is to start ArcMap. ArcMap does not have to be launched from the JTX application but can be launched from the Windows start menu or from any shortcuts available. Once in ArcMap, make sure the JTX toolbar is visible, select the Transaction Inspector tool, and click the feature whose history you wish to display. This brings up a dialog similar to the ArcMap identify window.



On the left you will see a tree view listing the GFID of the feature and a series of transactions nested below it, each with a time stamp sorted chronologically and the ID of the job in which the transaction was made. On the right, a list of fields and values is displayed.

For INSERT transactions, all fields in the database are shown. The PREVIOUS values are all set to INSERT since no values existed for the feature before it was created. The values assigned to the new feature when it was created are listed in the NEW column. The map display shows, in green, the geometric representation of the feature when it was created.

For MODIFY transactions, as shown in the figure below, only the fields that were changed are listed in the right-hand pane of the Transaction Inspector dialog. The geometric representation of the feature is shown in blue. If a geometry change was made as part of that transaction, the previous geometry is displayed in a lighter blue.

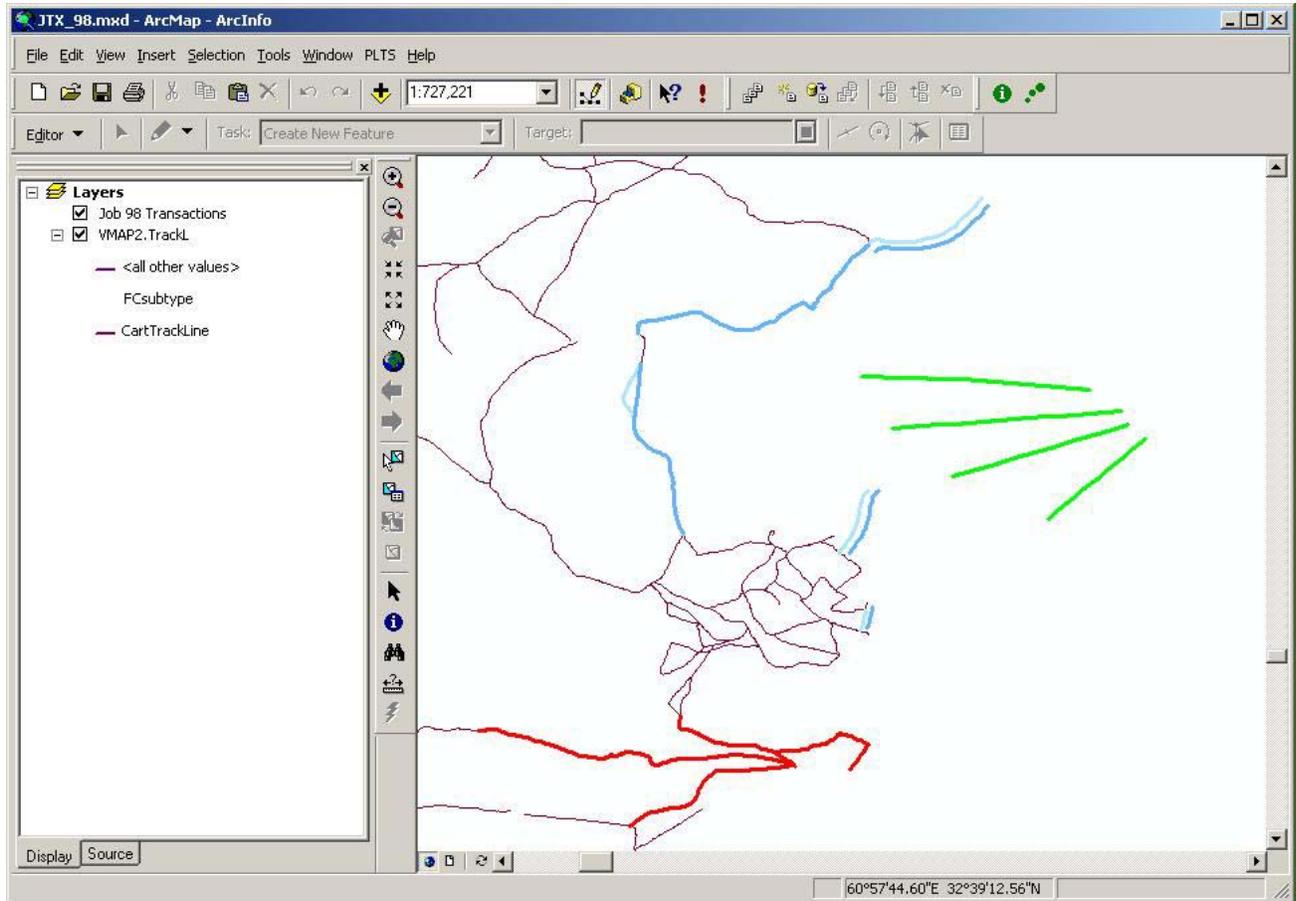
For DELETE transactions, all fields in the database are shown. Contrary to how the INSERT values are shown, the NEW values are all set to DELETE, and the last values the feature contained before it was deleted are listed in the PREVIOUS column. The geometry is displayed in red.



### *Loading All Transactions for a Job*

In some cases, we may want to see an overall visual representation of all the edits made in a particular job. To accomplish this, use the Load Job Transactions command on the JTX toolbar. This launches a dialog that asks for a Job ID. Once specified, a new layer is added to your map, showing each ADD in green, MODIFY in blue, and DELETE in red (see the following figure).

J-9005



## Security

Within the ArcGIS security model, specific user logins to the database are given permission to perform select, insert, update, or delete operations against certain feature classes. In this manner different users can have different security policies based on their unique database login. The cornerstone to this security model is the assignment of unique logins for each user in the database. Within the specific database implementation it is also possible to assign users to database ROLES allowing for easier management of user privileges.

The assignment of security through the database model allows for the security to be ubiquitous regardless of the application used to access the database. A user accessing via ArcMap, ArcCatalog, or any other database accessible application would have the same security model imposed, making for a more secure environment.

Core functionality in Job Tracking for ArcGIS assists in the use of the ArcGIS security model by automatically using the user's Windows login for database access, avoiding the requirement for users to repeatedly log onto the system. In this manner the user opens a job and, rather than using a generic application login, the user is diverted through his/her unique User-ID.

**Summary** The Job Tracking for ArcGIS extension is a comprehensive solution for job tracking and workflow management. Job Tracking for ArcGIS provides a framework and tools that integrate with core ArcGIS software, applications, and database technology to streamline job workflow, manage long transactions (geodatabase versions), and track history both of job transactions and feature edits. The use of XML provides cross-platform flexibility and portability of historical feature information. Job Tracking for ArcGIS is highly configurable and database driven. It provides a full suite of administration tools for customizing JTX to meet the specific needs of any project. Job types, steps, layers, toolbars, and other elements are completely configurable. Job Tracking for ArcGIS also comes with a rich API providing the flexibility to develop a custom user interface for performing all job tracking tasks or leverage key pieces of Job Tracking for ArcGIS functionality in other applications.



For more than 30 years ESRI has been helping people manage and analyze geographic information. ESRI offers a framework for implementing GIS technology in any organization with a seamless link from personal GIS on the desktop to enterprisewide GIS client/server and data management systems. ESRI GIS solutions are flexible and can be customized to meet the needs of our users. ESRI is a full-service GIS company, ready to help you begin, grow, and build success with GIS.

## Corporate

ESRI  
380 New York Street  
Redlands, California  
92373-8100, USA  
Telephone: 909-793-2853  
Fax: 909-793-5953

For more information  
on ESRI, call

**1-800-447-9778**

(1-800-GIS-XPRT)

or contact an ESRI reseller near you.

Send e-mail inquiries to

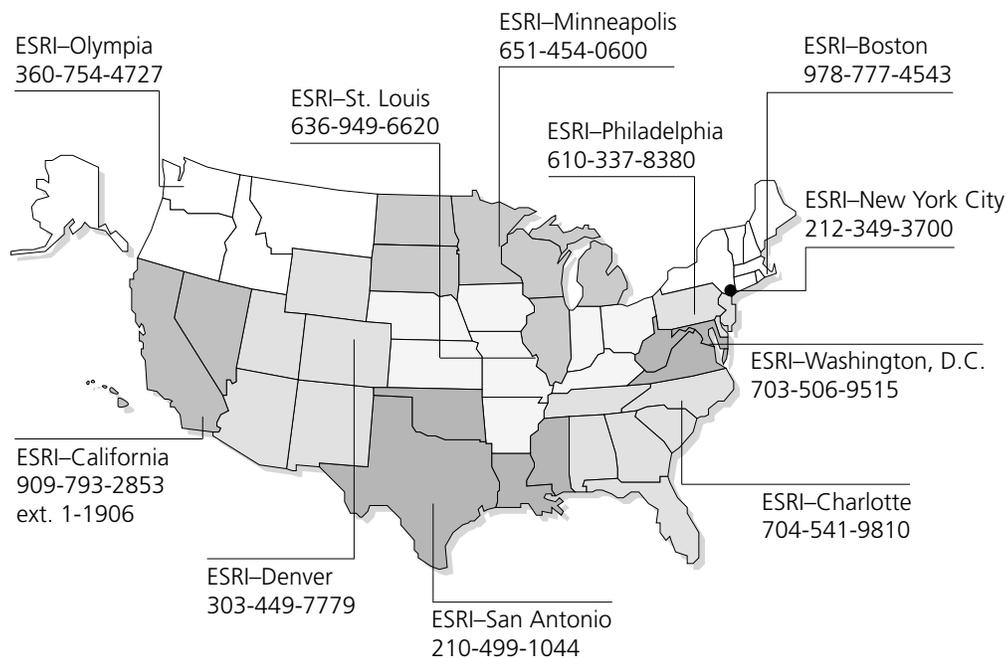
**info@esri.com**

Visit ESRI's Web page at

**www.esri.com**

Outside the United States,  
contact your local ESRI distributor.  
For the number of your distributor,  
call ESRI at 909-793-2853,  
ext. 1235,  
or visit our Web site at  
**www.esri.com/international**

## Regional Offices



## International Offices

Australia 613-9867-0447	France 33-1-46-23-6060	Japan 81-3-3794-6681	Singapore 65-742-8622
Belgium/Luxembourg 32-2-460-7480	Germany/Switzerland 49-8166-677-0 41-1-360-2460	Korea 82-2-571-3161	Spain 34-91-559-4375
Bulgaria 359-2-964-0850	Hungary 361-428-8040	Netherlands 31-10-217-0700	Sweden 46-23-755-400
Canada 416-441-6035	India 91-11-2620-3800	Poland 48-22-326-7300	Thailand 66-2-678-0707
China 852-2730-6883	Indonesia/Malaysia 62-21-570-7685 603-7874-9930	Portugal 351-2-1-781-6640	United Kingdom 44-1296-745-500
	Italy 39-06-406-96-1	Romania 40-21-231-13-81	Venezuela 58-212-285-9394



Place ESRI business partner or distributor address here.



No. GS-35F-5086H