Apollo Project
Version 1.0

User Guide

Nanometrics
SEISMOLOGICAL INSTRUMENTS
Kanata, Ontario • Canada

Apollo Project Version 1.0 User Guide

The information in this document has been carefully reviewed and is believed to be reliable for Version 1.0.*x*. Nanometrics Inc. reserves the right to make changes at any time without notice to improve the reliability and function of the product.

Nanometrics Inc.
250 Herzberg Road
Kanata, Ontario, Canada K2K 2A1
Tel +1 613-592-6776
Fax +1 613-592-5929
Email info@nanometrics.ca
www.nanometrics.ca

Part number 16106R1
Release date 2008-02-01

Preface

Document Conventions

Element	Font or symbol	Description and examples
Blank pages	header and footer text is BLANK PAGE	A page that is intentionally left blank
Cautions	 **Caution**	Essential information explaining (1) a risk of damage to equipment, data, or software where the recovery is likely to be troublesome; and (2) preventive action
Cross references	text is dark blue	Links to referenced information within the document
External links	text is blue	... at http://www.nanometrics.ca ... ... Get `jakarta-tomcat-5.0.28.tar.gz` and save it to ...
File names	`courier`	... Copy the file `ApolloProject.war` from the installation CD ...
GUI buttons	text is **bold**	Function buttons on graphical user interface (GUI) pages ... Click **Run Now** to run the project immediately ...
Links in the table of contents and in figure/table lists	green arial	... Chapter 1 Creating Projects
Notes	 **Note**	An explanation or comment that is related to the main text but is not essential information
Path names	`courier`	... and save it to the directory `/usr/local` ...
Token names	`courier`	Elements for building expressions; they include the % character ... output to directories `\%project\%year\%month\%day` for ...
User input to command windows	**`courier bold`**	Commands that are shown exactly as they must be entered ... and then enter **`mkdir $APOLLO_LOCATION/config`** ...
Variables and New terms	text is *italics*	Variables such as parameter names and value placeholders ... Run `ApolloProjectInstall-`*`versionNumber`*`.exe` ...
Warnings	 **Warning**	Essential information explaining (1) a risk of injury, or a risk of irreversible damage to data, an operating system, or equipment; and (2) preventive action

Document Scope

This User Guide provides Apollo Project Version 1.0 installation information, example projects, and procedures for using each of the features.

- Chapter 1 “Creating Projects” – an overview of how to create projects, and some examples
- Chapter 2 “Installing and Starting Apollo Project” – a brief introduction to operating environment requirements, installation instructions, and application run instructions
- Chapter 3 “Working with Apollo Project” – feature descriptions and procedures

Technical Support

If you need technical support please submit your request via the Nanometrics customer support site, or by email or fax. Include a full explanation of the problem and related information such as log files.

Support site: http://support.nanometrics.ca
Email: techsupport@nanometrics.ca
Fax: To: Support, +1 613-592-5929

Contents

Figures

Tables

Chapter 1
Creating Projects

This chapter provides the conceptual information required to create a project. It is recommended that this chapter be read in its entirety before the remainder of this book.

1.1 Application configuration

There are a few configurable items that can be set up prior to creating your first project. Setting up these configurable items before creating any projects can save repetitive project data entry and manipulation each time a project is created. Section 3.2 describes these items in detail.

The following items should be configured before creating any projects:

- Station Table Filename (Section 1.2.3)
- Base Directory (Section 1.2.6)
- Network Name, Location, and Channel Names (Section 1.2.3)

1.2 General steps for creating a project

Creating a project involves the following steps:

- Choosing the type of project
- Specifying the store(s) from which data will be retrieved
- Choosing the channels
- Specifying the time range(s)
- Choosing the output format
- Choosing the output location

1.2.1 Choosing the type of project

Apollo Project includes four project types that serve as templates for individual projects:

- Archive
- One Time
- Controlled Source
- Trigger

These project types differ in the way that the requested time range(s) to retrieve are specified.

Archive projects specify an open-ended time range to retrieve from the specified store(s). Archive projects attempt to retrieve all requested data from the specified start time until the current time. Successive runs of archive projects normally yield additional retrieved data if additional data have

been added to the specified store(s) in the meantime. Archive projects can be scheduled to be run automatically at a pre-determined interval or on particular days of week at a specific time. Archive projects are therefore particularly suited to situations where data must be accumulated over time.

One Time projects specify a single, specific time range for which to retrieve data.

Controlled Source projects specify one or more manually-entered shot times whose corresponding time ranges are bounded by pre and post times. For example, if the project pre and post times were specified as 30 and 90 seconds respectively, adding shot times x and y would result in the following time ranges being retrieved from the specified store(s): [x-30 seconds, x + 90 seconds), [y-30 seconds, y + 90 seconds). In addition, controlled source projects only allow output as SegY combined files.

Trigger projects are very similar to Controlled Source projects with the difference that a trigger file is provided from which the time ranges to retrieve are calculated. The trigger file is a CSV file that can be downloaded from a Taurus.

1.2.2 Specifying the stores from which data will be retrieved

At least one store specification must be specified in a project. A store specification is used to identify one or more store instances from which data should be retrieved. There are four different kinds of store specifications:

- File
- Directory
- IP Address
- Discovery

A file store specification is used to specify a particular store that resides on a filesystem accessible on the server running Apollo Project. Note that the filesystem can be a local or a networked filesystem (for example, accessible via SMB or NFS).

A directory store specification is used to specify the set of stores that reside in a particular directory hierarchy on a filesystem accessible on the server running Apollo Project. A recursive search from the specified directory is used to identify all of the stores to include. Note that the filesystem can be a local or a networked filesystem (for example, accessible via SMB or NFS).

An IP Address store specification is used to specify the store of a particular instrument. The specified IP address must be an IP address bound to the Ethernet interface of a Taurus. The Taurus must be running in communications mode and be reachable from the server running Apollo Project.

A discovery store specification is used to specify the stores of multiple instruments. A class D (multicast) IP address and port is specified. The store of any Taurus in communications mode that is reachable from the server's LAN running Apollo Project and has discovery enabled for the specified IP address and port will be included in the project when it runs. By default, the Taurus has 224.199.71.138:6776 set as its discovery address.

Note The actual store(s) from which data are retrieved may vary on successive project runs. For example, a discovery store specification may result in a different number of instruments being included depending on the current reachability of the instruments.

This behaviour can be leveraged by using directory or discovery store specifications as a convenient way to add additional stores to an existing project without having to modify the project.

1.2.3 Choosing the channels

When a set of stores is specified for a project, Apollo Project retrieves the time series channels present in each store identified by the set of store specifications associated with the project and allows the user to specify which ones are to be retrieved. They can be filtered based on being channel 1, channel 2, or channel 3. In addition, you can filter on the Network, Station, Location or Channel names; as well as the instrument name. All filters use * as wild cards, (for 1 or more characters) and are comma separated.

Apollo Project allows for a mapping to occur between a given instrument/channel (i.e. taurus_1234/digitizer/timeSeries1) and a corresponding network, station, location, and channel name. This mapping is file-based. This is referred to as the StationTable file. A project can have project-specific Station Table file or multiple projects can share a Station Table file if they refer to the same mapping file. A Station Table file can also be set as the default Station Table.

The Station Table file tells Apollo Project what station naming information to put in the resulting output files. Apollo Project does not use the playback information configured on a Taurus. Without listing a Station Table file for a project, the information on output will be the network, location, channel information set in the configuration page with the station name being A (if the data is from a Taurus) or R (if the data is from a Trident305) followed by the serial number of the unit.

For example, using the default configuration, if the Network = NE, Location = [blank] and Channel1 = BHZ then taurus_1234/digitizer/timeSeries1 would map to NE.A1234..BHZ

For example, using the default configuration, if the Network = SA, Location = AA and Channel1 = BHZ then trident305 _5678/digitizer/timeSeries1 would map to SA.R5678.AA.BHZ.

To have different values for the names a Station Table file should be created. It is highly recommended that you create a Station Table file and that it be set as the default Station Table file to use for all projects. The default Station Table file can be set in the application configuration.

The following is the recommended way to create a Station Table file.

1. Set the network, location and channel names in the configuration.
2. Create a new Project (by default the Station Table file is generated with the name of the project as part of the file name, so it is suggested you name this project Station Table).
3. On the Store Selection tab select as many stores as possible.
4. On the Channel Selection tab you should see a list of all of the selected channels with the Network.Station.Location.Channel column being blank.
5. Click the Create Default Station Table button. This will create a default Station Table with default entries listed above. You should never have to use this button again. Note the Station Table file Name. This will be used later.
6. Edit the station table entries in the table. All of the names can be edited to appear in the output files as desired. If the Location is empty, it is interpreted as the SEED default of two spaces.
7. Continue to define the project as normal and then click OK. (If you are just creating a project to create the Station Table file you can click OK now).
8. In the configuration enter the Station Table file Name that was generated above. This file and all of these station mappings will be used by default for future projects.

Channels that do not have a mapping will appear as blanks in the channel table on the Channel Selection tab. Mapping information for these channels can be entered and saved in the Station Table

file for future projects. At any time the entries in the station mapping can be changed and such changes will be reflected in future runs of any project using that Station Table file.

Note Only entries that are applicable for the given channels selected appear in the selected channels table.

Also, for any project you can create a new Station Table file. This file can be specific to a project or it can be shared among many projects. There are two recommended ways to do this:

- Copy the existing Station Table file to a new file name before creating the project, enter that File Name on the Channel Selection tab and edit the entries as required.
- On the Channel Selection tab, click the Create Default Station Table button and edit the entries in the selected channels table.

1.2.4 Specifying the time range(s)

As discussed in section 2.1.1, there are a variety of ways to specify time ranges for which data are to be retrieved. The choice of project type governs this.

One must also specify the “file interval” to use when creating output file(s) for the retrieved data. The file interval specifies what duration of time each generated file should represent. For example, if a One Time project is used and an hour of data is specified to be retrieved from a single channel for a single instrument whose data are present in the specified store(s), then a file interval of 10 minutes will result in 6 files being generated.

Caution Care should be taken when specifying the file interval to use. For example, retrieving many hours or days of data but specifying a file interval of seconds or minutes granularity can result in potentially many thousands of files being generated. This is not only very time-consuming and resource-intensive but represents a data management problem as well.

1.2.5 Choosing the output format

All project types with the exception of Controlled Source allow the retrieved data to be made available in the following formats:

- MiniSEED
- SEGY
- Seisan

The Controlled Source project type requires that the retrieved data be made available in SEGY format only.

The MiniSeed choice allows one to further specify whether the data from multiple instruments be combined in the target output file(s) or not. The record size can also be specified (the default record size is 512 bytes). If sorted is selected, the data is guaranteed to be sorted in chronological order for each individual channel.

The SEGY choice allows one to further specify whether the data from multiple instruments be combined in the target output file(s) or not and whether or not they should be sorted based StationName.NetworkName.

1.2.6 Choosing the output location

A very flexible mechanism is used to define where to store retrieved and processed data. There are three components used to specify the output location for a project:

- Base Directory
- Subdirectories
- Filename

The Base Directory is the root of all data retrieved for the project and can be anywhere in the file-system of the server running Apollo Project for which write access is possible. A lot of data can potentially accumulate here so sufficient care must be taken to ensure that adequate disk space is available. The Base Directory can be configured to be the same by default for all projects. For deployments of Apollo Project where the clients are not running on the server machine, it is recommended that the filesystem containing the base directory on the server machine be shared or exported as a networked filesystem if possible to facilitate easy and convenient access to the output data.

The Subdirectories component specifies a template for the directory hierarchy used to organize the retrieved data. The following variables can be used in the template. The variables are substituted at run-time according to the data being retrieved and output by Apollo Project:

- %year: maps to the 4-digit year of the beginning of the time range represented by the file being created
- %month: maps to the 2-digit month of the beginning of the time range represented by the file being created
- %day: maps to the 2-digit day of the beginning of the time range represented by the file being created
- %time: maps to yyyyMMdd_HHmmss (i.e. a complete timestamp of the beginning of the time range represented by the file being created)
- %project: maps to the project name
- %format: maps to the requested output format (e.g. MiniSeed)
- %network: maps to the network name
- %station: maps to the station
- %channel: maps to the channel
- %instrument: maps to the instrument ID (e.g. taurus_1234)

The Filename component specifies the name of each output file that is created. The same variable substitutions available for the Subdirectories component are available to the Filename component.

Note that some of the variables can be ambiguous depending on the data being retrieved for the project (e.g. %channel is ambiguous if multiple channels are requested for the same instrument).

Apollo Project creates output files corresponding to each instrument ID found in the specified store(s) unless the option to combine instrument data is chosen. The combination of the set of retrieval time ranges for the project, the chosen file interval, and the number of instruments whose data are retrieved drives the number of output files created.

Note Apollo Project stores intermediate files in a “temp” subdirectory of the Base Directory specified for a project’s output location. These intermediate files correspond to the actual channel data retrieved from the store(s) referenced by the project. If a project is deleted from Apollo Project, these temporary files will be automatically removed. Alternatively, disk space can be reclaimed if necessary by removing these temp files manually (meaning, without deleting the project). However, if the project is run again, the output files produced may be incomplete if previously retrieved data from these temporary files is required.

Example:

Project Name: Project1

Project Type: One Time

Channels selected for retrieval: CA.OTT.STN1.BHZ (on taurus_1234)

Requested Time Range: January 30, 2008 14:00 – 15:00 UTC

File Interval: 20 minutes

Output Location Base Directory: /disk2/ApolloProjectOutputData

Output Location Subdirectories: /%project/%year/%month/%day

Output Location Filename: %instrument_%time.%format

When the project is run successfully, the following output files will be created on the server running Apollo Project:

- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234_20080130_140000.MiniSeed
- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234_20080130_142000.MiniSeed
- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234_20080130_144000.MiniSeed

Example:

Project Name: Project2
Project Type: Controlled Source
Channels selected for retrieval: CA.OTT.STN1.BHZ (on taurus_1234), CA.OTT.STN1.BHY (on taurus_1234)
Requested Shot Times: January 30, 2008 14:00 UTC, January 30, 2008 14:30 UTC
Pre Time: 30 seconds
Post Time: 30 seconds
File Interval: 30 seconds
Output Location Base Directory: /disk2/ApolloProjectOutputData
Output Location Subdirectories: /%project/%year/%month/%day
Output Location Filename: %instrument/%channel_%time.%format

When the project is run successfully, the following output files will be created on the server running Apollo Project:

- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234/BHZ_20080130_135930.MiniSeed
- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234/BHY_20080130_135930.MiniSeed
- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234/BHZ_20080130_142930.MiniSeed

- /disk2/ApolloProjectOutput-Data/Project1/2008/01/30/taurus_1234/BHY_20080130_142930.MiniSeed

1.3 Running Projects

When a project is run, Apollo Project performs the following steps:

- The requested time range(s) are identified
- The channels for which data should be retrieved are identified
- The actual stores specified by the project's store specifications are identified

For each store identified, the following steps are performed:

- A data availability analysis for the outstanding time range(s) is performed (i.e. those ranges that have not already been retrieved). The availability of the data is recorded. It is possible that a given store was not present at the time the project was last run. If this is the case then an attempt will be made to retrieve all of the project's required time range(s) from the store.
- If any data exists in the store for any of the requested time range(s) then a retrieval of this data is performed and the retrieved data are stored to a local cache.
- Otherwise, if the number of retrieval attempts for a given time range exceeds a threshold (configurable via the application configuration facility described in Section 3.2), then the range is marked to not be attempted again.

If any data was retrieved during the run of the project then corresponding output file(s) of the desired output format are generated according to the project's output location specification.

It is possible that the requested data for a given channel exist across multiple stores. If all of these stores are included by the project's store specifications then Apollo Project is able to assemble the requested data, regardless of the fact that the data are distributed over more than one store.

When a project runs, detailed information regarding the results of the retrieval and processing are available. For each requested channel, one is able to see the amount of data retrieved, how much was unavailable, and how much was not retrievable (e.g. due to a store access problem). For unavailable ranges, one is also able to see how many retrieval attempts remain before Apollo Project declares the range irretrievable. This capability is described in more detail in Section 3.9.

1.4 Example Projects

The following sections provide examples of possible deployment scenarios.

1.4.1 One Time

It may be required to download three days of data from multiple Tauruses on the network, saving the data in MiniSeed format, each file having one hour in duration. It is also required that the data for each day be saved to a different directory.

In this situation, a One Time project would be most suitable. Since the three specific Tauruses are each in communications mode, each Taurus' store can be made available to Apollo Project by creating a corresponding IP Address store specification with the IP address of the Taurus. The requested time range can be specified along with a file interval of one hour. The project should have at least the "%day" variable present in the Subdirectories component of the Output Location specification to ensure that a each day's data are saved to a different directory.

1.4.2 Scheduled Archive

It may be required to automatically accumulate seismic data from a set of Tauruses on the network every six hours. Since it assumed that the Tauruses are always reachable and it is required that data be accumulated, a scheduled archive project would be most suitable. The “Regular Interval” scheduling option can be used and the interval would be set to six hours.

Note that at any time a request to run a scheduled project can be made. For example, if the scheduled retrieval interval was six hours and mid-way between intervals one wanted to retrieve the data early, the project can be run and approximately three additional hours of data would be accumulated. Approximately three hours later the project would run automatically and accumulate the remaining three hours of data.

1.4.3 Manual Archive

It may be required to accumulate seismic data from a set of Tauruses but their respective stores are only available for retrieval on occasion (i.e. no regular, dependable schedule for retrieval from these stores is possible). In this situation, a manual archive project can be created and the project run whenever any of the Tauruses are available for retrieval.

1.4.4 Media Harvesting

It is often necessary to collect the media from a set of instruments in the field at some regular interval (e.g. once per week). Furthermore, it may be required that all of the seismic data on these media be accumulated for subsequent analysis.

In this situation, an archive project would be most suitable. A directory on the filesystem of the server running Apollo Project could be created initially. A directory store specification would be created with this directory as the base directory from which to search for stores. The content of each media instance in the week’s collection could be copied via a media reader to a sub-directory of the base directory. When the project is run, all of the copied stores would be automatically available to the project and data for each instrument would be accumulated.

Note that due to the nature of the media harvesting and replacement procedure in use for this example, there will always be data missing for the last time range output file(s) in the current week. This time will correspond to the time that the media was taken from the instrument and replaced with a new one. However, in the following week when the media are harvested again from the instruments in the field, the project will fill the missing time in these files with the new store data, regardless of the media instance that the data exist on.

1.4.5 Trigger Download Example

Problem: You have a network of 5 Tauruses around a Dam all of which are running in communication mode and are accessible over the Ethernet. You come in Monday morning and you want to see the activity from all of the five Tauruses over the weekend. You just want to see for 30 seconds before and 5 minutes after any activity but you don’t want to look at all of the data for the weekend.

Solution:

1. The Taurus that sits at the site that has the least environmental noise, or what you consider your best site should be configured to generate triggers (see the Taurus manual for details on how to do this). This needs to be done in advance. Lets call this Taurus A.

2. On Monday morning download the trigger file from Taurus A (See the Taurus manual for details on how to do this). Look at the file, you may want to edit this file if there are too many triggers. If there are 10 triggers in the file we should expected 10 files outputted.
3. Create a new Project to download the information as follows:
   a) On the Projects page select new project.
   b) Enter a project name (ie, Triggers Jan 4-7) and select a Triggers file.
   c) On the Store Selection tab enter the IP address of all five of the Tauruses (Tip: if you have entered the IP addresses previously they might be available in the Recent Specs). After doing this you should have 5 selected stores in the lower box.
   d) On the Channel Selection tab, make sure that the channels have an appropriate mapping. See Section 3.5.2 for more information.
   e) On the Time Selection tab, click on the browse button and select the trigger file you downloaded in the previous step and submit. This should populate the triggers list with all of the triggers. Enter a pre time of 30 seconds and a post time of 5 minutes.
   f) On the Output Format tab, select MiniSEED and Combined.
   g) On the Output Location tab, select the appropriate output base directory (say c:\output and subdirectory, for example /%project%/. A suggested filename would be %time.
   h) Click OK.
4. On the Project Summary page, check that the project looks okay
5. Click the Run Now button.
6. In a few minutes there should be ten files in c:\output\Triggers Jan4-7 directory each containing 5 min 30 secs of data from 5 Tauruses.

Chapter 2
Installing and Starting Apollo Project

2.1 About Apollo Project

Apollo Project Version 1.0 is a web server product for the convenient retrieval of time-series data from Nanometrics Version 2.*x* Stores. The web browser client provides options for you to define complex, reusable data-retrieval tasks; these tasks are called “projects”.

You can define projects to retrieve data from multiple data sources, such as from Tauruses discovered on a network and stores that have been copied to PCs on your LAN. The data can be downloaded to MiniSEED, Seisan, or SEGY format.

Apollo Project includes four project types that serve as templates for individual projects:

- **Archive** – Archive projects are for retrieving data from sources that are continually updated, such as a directory on a PC that has new data saved to it periodically, or a multicast group of Tauruses which has different instruments included at different times.
- **One Time** – One Time projects are for retrieving a single time segment.
- **Controlled Source** – Controlled Source projects are for retrieving data specified as a set of individually entered shots.
- **Trigger** – Trigger projects are for retrieving data specified by a CSV-formatted trigger list.

All projects can be run manually, and any manually-run project can be run repeatedly. Archive projects can also be defined to run automatically on a schedule.

Apollo Project requires a valid software licence to allow access to all options (Section 2.2.2.1 on page 12).

2.2 Environment requirements

Apollo Project Version 1.0 has operating environment requirements for clients and for the server as listed below.

2.2.1 Client

A client on any platform is a web browser with Javascript enabled and IP access to the server. Apollo Project Version 1.0 has been tested with Internet Explorer 7 and Firefox.

2.2.2 Server

Environment requirements are listed for running Apollo Project Version 1.0 on Linux (Table 2-1) and Windows (Table 2-2).

Table 2-1 Requirements for a server running Linux

System component	Requirement
Operating system	Any distribution running 2.4 or later kernel ◆ Tested on RedHat 9/kernel 2.4.20-8
Memory	◆ 512 MB RAM minimum ◆ 2 GB RAM recommended
Disk space	Free disk space for each 3 channels of data retrieved is approximately 2 GB per month. This is roughly twice as much space as the retrieved data occupies in a store.
Java	Sun Java Development Kit (JDK) 1.6.0 or later
Tomcat	Version 5.0.28 only http://archive.apache.org/dist/jakarta/tomcat-5/v5.0.28/bin/jakarta-tomcat-5.0.28.tar.gz
Software licence	The licence file is installed on the Apollo Project server PC; see also Section 2.2.2.1.

Table 2-2 Requirements for a server running Windows

System component	Requirement
Operating system	Windows XP Sevice Pack 2
Memory	◆ 1 GB RAM minimum ◆ 2 GB RAM recommended
Disk space	Free disk space for each 3 channels of data retrieved is approximately 2 GB per month. This is roughly twice as much space as the retrieved data occupies in a store.
Java	Sun Java Runtime Environment (JRE) 1.6.0 or later
Tomcat	Version 5.0.28 only ◆ This is included in the Apollo Project installation package for Windows XP
Software licence	The licence file is installed on the Apollo Project server PC; see also Section 2.2.2.1.

2.2.2.1 Software licence

Apollo Project requires a valid software licence in order to allow access to all features. It determines the maximum number of saved projects, maximum number of channels, and the maximum number of concurrent project runs. In the absence of a valid licence Apollo Project will default to the unlicenced configuration. That is, it will allow you to save one project definition at a time for a store of up to three channels.

- Click the About icon on the home page to view licence information (Figure 2-1).

Figure 2-1 The About page and licence information

2.3 Installing Apollo Project

The server can be on one computer. Clients can run locally on the server or on additional remote computers. For information about planning the installation of the Apollo Project software, see Chapter 1, “Creating Projects” . All stores must be accessible from the server hosting the Apollo Project software. All output files are created on the server and its accessible file systems, as described in Section 1.2.6.

2.3.1 Install on Linux

1. Install Tomcat version 5.0.28 (as root). The recommended installation directory *`tomcatLocation`* is `/opt`, if Tomcat is not already present.
   a) Get Tomcat and save it to *`tomcatLocation`*.
   (http://archive.apache.org/dist/tomcat/tomcat-5/v5.0.28/bin/jakarta-tomcat-5.0.28.tar.gz).
   b) Enter these commands:
   `cd` *`tomcatLocation`*
   `gunzip jakarta-tomcat-5.0.28.tar.gz`
   `tar xf jakarta.tar`
2. Install a supported version of Java Development Kit: JDK 1.6.0 or later.
3. Set up the Tomcat environment.
   - If Tomcat is used exclusively for Apollo Project:
     a) Copy the `startApolloProject.sh` and `shutdownApolloProject` scripts from the Apollo Project installation medium to the desired location (for example, `/usr/local/bin`).
     b) Edit the `startApolloProject.sh` script and specify values for the following three variables:
     `TOMCAT_LOCATION=`*`absolute path to Tomcat location`*
     `APOLLO_LOCATION=`*`absolute path to appropriate filesystem location`*

```
JAVA_HOME=absolute path to JDK installation
```

(`APOLLO_LOCATION` is the place in the filesystem where all Apollo Project data files, except output files, are stored.)

- If Tomcat is not used exclusively for Apollo Project and a mechanism already exists for starting Tomcat, then please ensure that all of the variables for which values are set in the `startApolloProject.sh` script are in the environment Tomcat when it is run. You can do this either by modifying the environment of the user who runs Tomcat or by making the corresponding changes to the script used to start Tomcat.

4. Copy `ApolloProject.war` from the installation medium into the `webapps` subdirectory of the Tomcat installation directory (that is, `$TOMCAT_LOCATION`).
5. Enter the command **`mkdir $APOLLO_LOCATION/config`**.
6. Copy `apollo.lic` (the licence file provided by Nanometrics) from the installation medium to `$APOLLO_LOCATION/config`.
7. At this point the `startApolloProject.sh` script can be used to start Apollo Project, or you can invoke Tomcat via an existing mechanism if applicable.

2.3.2 Install on Windows

Note APOLLO_LOCATION is an environment variable. If ApolloProject is uninstalled and then re-installed with a different *APOLLO_LOCATION* without rebooting Windows in between, the old *APOLLO_LOCATION* is still in effect. This location is in effect because the parent process from which the environment is inherited still has the old value, even though the Tomcat service was restarted. The parent process only reads the system environment at boot time.

If you change *APOLLO_LOCATION* after ApolloProject has already been installed, then a reboot will be required to put the change in effect. The reboot is required regardless of whether such a procedure we define involves simply changing the value of APOLLO_LOCATION via the control panel or uninstalling/re-installing via the installer.

To install on Windows:

1. Run the Apollo Project installer `ApolloProjectInstall-`*`versionNumber`*`.exe`.
2. Set the APOLLO_LOCATION environment variable to the value you entered

   (`APOLLO_LOCATION` is the place in the filesystem where all Apollo Project data files, except output files, are stored.)
3. Install the file `apollo.lic`: Run the licence file installer `ApolloProjectLicenseInstall-`*`versionNumber`*`.exe`.

2.4 Running Apollo Project

2.4.1 Running a client

To run an Apollo Project client:

- Open a browser and go to http://*serverAddress*:*port*/ApolloProject. The application name ApolloProject in the URL is case-sensitive on all operating systems.

  For example, with the application server and the client browser running on the same computer, go to http://localhost:8080/ApolloProject.

Note Some Firefox browser plug-ins may interfere with Apollo Project client start-up on Windows XP.

2.4.2 Stopping and starting the server

Note During installation the server is set to run automatically. For a typical Apollo Project setup this default would be left unchanged.

On Linux:

- To shut down the server:
  ```
  cd /opt/jakarta-tomcat-5.0.28/bin
  export JAVA_HOME=absolute path to JDK installation
  ./shutdown.sh
  ```
- To start the server, run the script `startApolloProject.sh` (or you can start Tomcat via an existing mechanism if applicable).

On Windows you can use either Windows Services or the Monitor Tomcat application to stop and start the server:

- Go to Start > Control Panel > Administrative Services > Services, and then select Apache Tomcat in the list of services to access its options.
- Go to Start > All Programs > Apache Tomcat 5.0 > Monitor Tomcat to start Monitor Tomcat, and then right-click its icon  in the system tray to access the options.

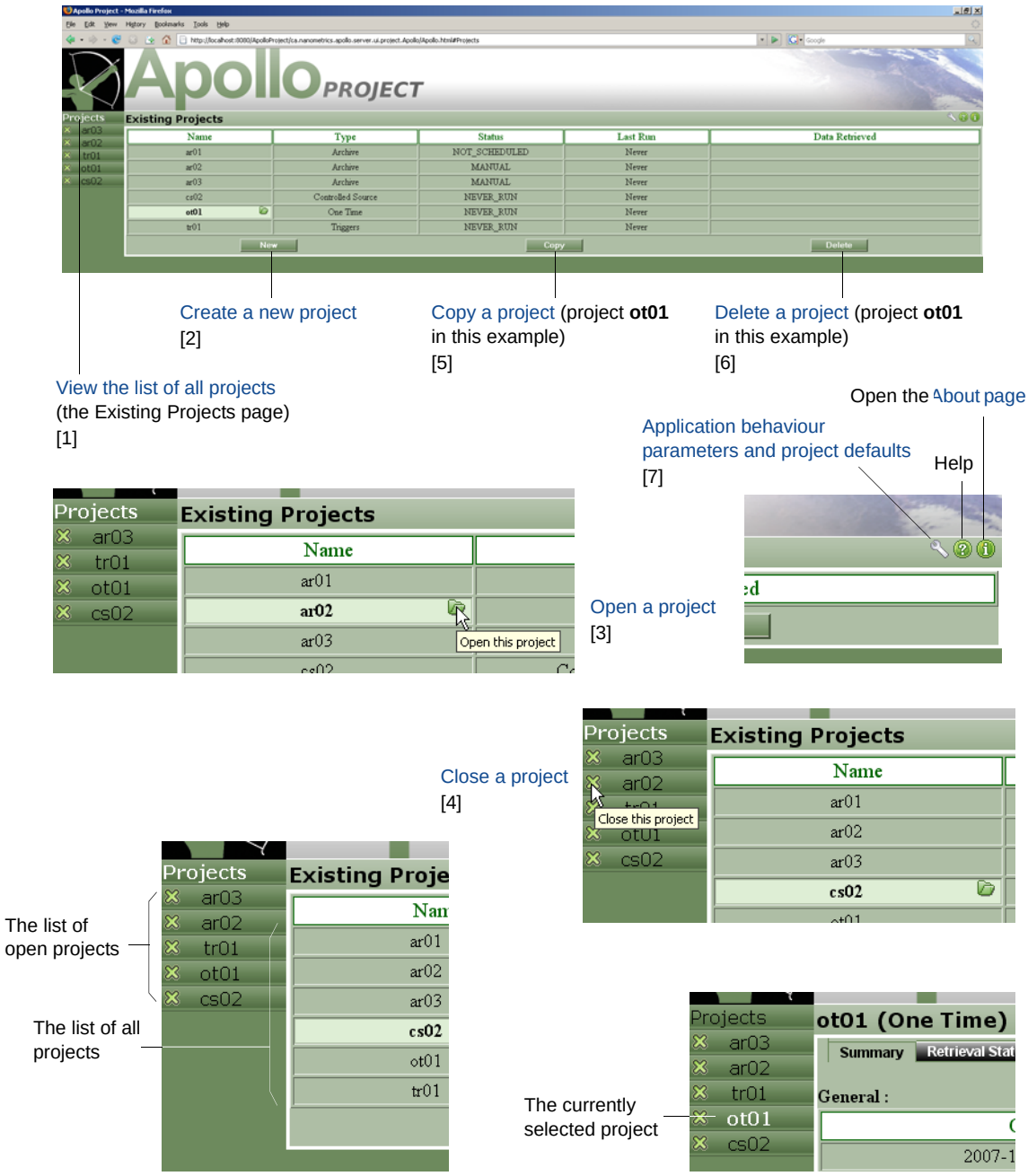
Chapter 3
Working with Apollo Project

This chapter includes feature descriptions of Apollo Project basic functions for working with projects, and procedures for defining projects. It is recommended that you read Chapter 1, “Creating Projects” in its entirety before reading this chapter.

3.1 Apollo Project basic functions

The Apollo Project home page and Existing Projects page provide these functions for working with projects (see also Figure 3-1).

Figure 3-1 Basic functions on the home page*



* The numbers [1] ... [7] are references used in procedures below.

3.1.1 View the list of all projects

- Click **Projects** at the left side of the main page ([1] on Figure 3-1).

This will open the Existing Projects page. The Existing Projects page shows the list of all projects and a status overview of each project (Section 3.9 on page 36). It is the default page on opening an Apollo Project session.

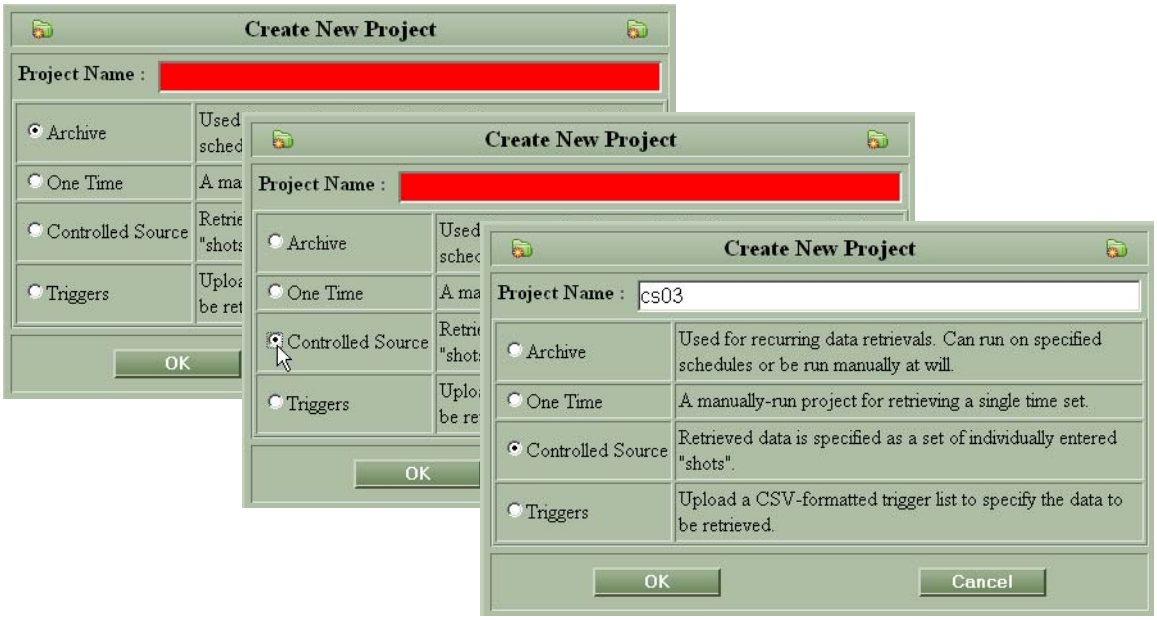
3.1.2 Create a new project

You can create multiple projects, up to the number supported by the software licence (see also Section 2.2.2.1 “Software licence” on page 12).

To create a new project:

1. Click **New** in the Existing Projects page ([2] on Figure 3-1). This will open the Create New Project page.
2. Choose a project type and then enter a unique Project Name (Figure 3-2).
3. Click **OK**. This will open the new project’s definition pages. (Section 3.5)
4. Edit settings as required on the definition pages:
   - Store Selection (Section 3.5.1 on page 25)
   - Channel Selection (Section 3.5.2 on page 27)
   - Time Selection (Section 3.5.3 on page 29)
   - Output Format (Section 3.5.4 on page 32)
   - Output Location (Section 3.5.5 on page 33)
5. Click **OK** and then wait a few moments for the new project to open.

Figure 3-2 Create a new project



Note You can edit settings for the new project only before the project has run for the first time or before it has been scheduled to run. However, if you cancel the schedule, you can then edit the project settings. If the project has been run or scheduled to run, you can copy the project (Section 3.1.5) and then edit the settings in the copy.

3.1.3 Open a project

- Click on the project name and then click the Open Project icon ([3] on Figure 3-1).

3.1.4 Close a project

- Click the Close Project icon on the project tab ([4] on Figure 3-1).

3.1.5 Copy a project

1. If it is a scheduled Archive project, cancel the schedule.
2. Click on the project name and then click **Copy** ([5] on Figure 3-1).
3. Enter a unique name.
4. Edit settings if required (Section 3.5).
5. Click **OK** to save the new project

3.1.6 Delete a project

- Click on the project name and then click **Delete** ([6] on Figure 3-1).

None of the processed data files are deleted for the project. All files in the temp directory are deleted for that project.

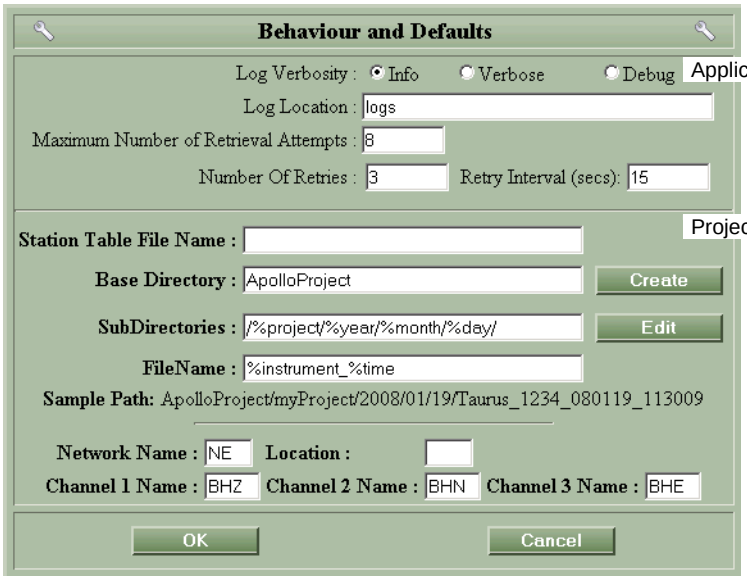
3.2 Application behaviour parameters and project defaults

You can edit basic settings for Apollo Project behaviour and the default values that will be used for new projects. Project default settings include, for example, the Base Directory path, and network and channel names. Application behaviour settings include, for example, the maximum number of data retrieval attempts for missing segments.

Changes to Project Defaults parameters take effect only for new projects. Application behaviour parameters take effect for all projects, starting from the next time any project is run. (Figure 3-3).

1. Click the configuration icon on the main page ([7] on Figure 3-1). This will open the Behaviour and Defaults page (Figure 3-3).
2. Edit the settings (Table 3-1).
3. Click **OK** to save the new settings.

Figure 3-3 Behaviour and Defaults page



Application Behaviour parameters

Project Defaults parameters

- Create a new Base Directory
- Use the Subdirectory and File Configuration page to define data output locations

For information about data output locations, see Section 1.2.6.

Table 3-1 Behaviour and Defaults parameters

Parameter		Description
Log Verbosity		The verbosity of the Apollo Project log. ◆ Options: Info, Verbose, Debug. Default is Info.
Log Location		The pathname of the Apollo Project log file as defined during installation. ◆ Valid values: a valid pathname. Default is *APOLLO_LOCATION*\logs.
Maximum Number of Retrieval Attempts		Every time a project is run, there is 1 retrieval attempt for all missing time segments on all channels. This parameter sets the maximum number of project runs to attempt retrieval of missing segments for a previously attempted time range for a channel. (The case where the Taurus is busy does not decrement this value; for example, if the Taurus is processing an earlier download request.) You can view the channel details page to see if there are any remaining retrieval attempts for any channels (Section 3.10). ◆ Valid values: an integer in the range 0 to MAX INT. Default is 8.
Number of Retries		The number of times for Apollo Project to try to connect to a data source if there is an IP network connectivity problem. This is specific to the project run; every project run starts with the maximum number of retries. ◆ Valid values: an integer in the range 0 to MAX INT. Default is 3.
Retry Interval (sec.)		The interval in seconds between IP connection retries. ◆ Valid values: an integer in the range 0 to MAX INT. Default is 15.
Output location	*Base Directory*	The pathname of the base directory for the processed data subdirectories and files. This value is held in the APOLLO_LOCATION environment variable. ◆ Valid values: any valid absolute pathname on the server. It can also include any additional characters that are valid for paths. Default is ApolloProject.
	SubDirectories	The expression used to define pathnames for subdirectories under *Base Directory*. ◆ Valid values: any of the tokens in any order; it can also include any additional characters that are valid for paths. Default is \%project\%year\%month\%day
	FileNames	The expression used to define filenames for processed data files within *Base Directory**SubDirectories*. ◆ Valid values: any of the tokens in any order; it can also include any additional characters that are valid for filenames. Default is %instrument_%time
	Sample Path	A dynamic example of the resulting path and filename as you enter *Base Directory*, *SubDirectories*, and *Filenames* values.
Station Table File Name		The path and filename of a station table file. A station table is used to assign station names to all of the selected stores. ◆ Valid values: a valid path and filename. No default value.
Default Network Name		A standard network code. ◆ Valid values: any 2 characters. Default is NE.
Default Channel1 Name		A channel code. ◆ Valid values: any 3 characters, each of which must be either an upper case letter or a digit. Default is BHZ.
Default Channel2 Name		A channel code. ◆ Valid values: any 3 characters, each of which must be either an upper case letter or a digit. Default is BHN.
Default Channel3 Name		A channel code. ◆ Valid values: any 3 characters, each of which must be either an upper case letter or a digit. Default is BHE
Location		A location within a station. ◆ Options: any 2 ASCII characters. Default is 2 spaces for “none”. (Remember to delete the 2 default spaces if you enter a new location value.)

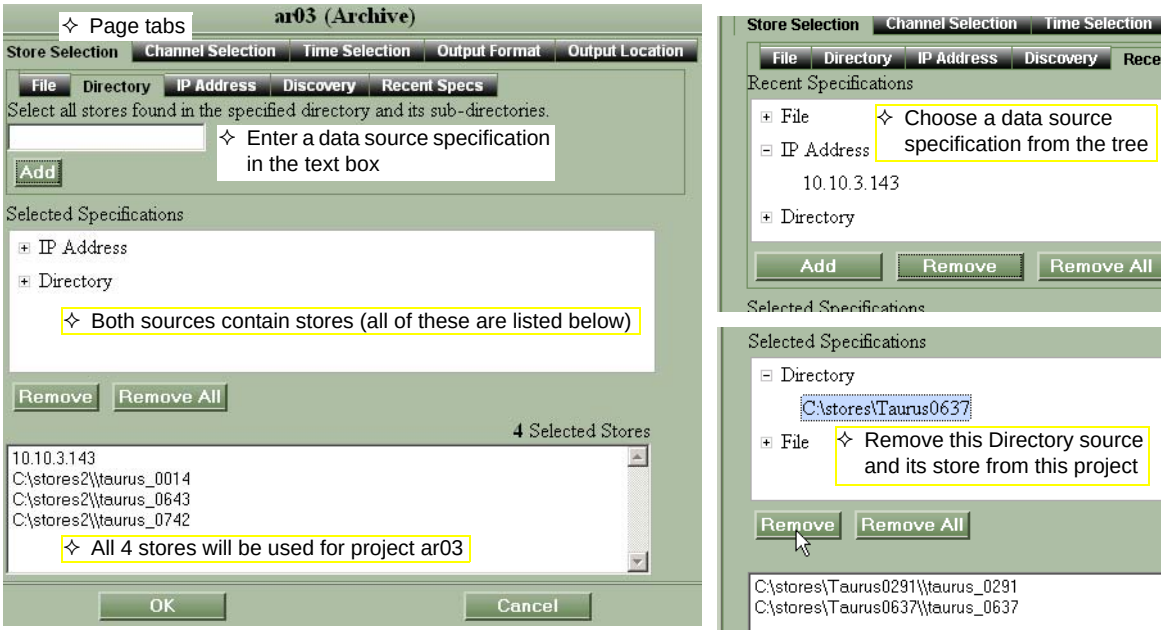
3.3 Project definition basic functions

Basic project definition functions and page features are listed in Table 3-2 and shown in Figure 3-4. See also Section 3.4 “About data sources and selected stores”.

Table 3-2 Project definition page basic functions

Function or feature	Usage
Selected Specifications	◆ View the list of data sources that are specified for the project ◆ Select a data source to be removed from the project
Selected Stores	View the list of stores to be used in the project
Add	Specify a data source to include in the project
Remove	Remove a store from the project’s currently specified set
Remove All	Remove all data sources from the project
OK	Save the project definition and exit to an Apollo Project main page
Cancel	Discard the new project definition or the edits to an existing project, and exit to an Apollo Project main page.

Figure 3-4 Project definition pages



3.4 About data sources and selected stores

To add or remove a store, you add or remove its store specification.

Data is retrieved from stores. You specify which stores to download from by entering store specifications. A store specification can result in zero or more stores being selected.

The Store Specification page is divided into three sections. You enter the store specification in the top portion of the page. The specification can be entered in one of five ways, as described in Section 3.5.1. The middle section shows the store specifications that you have entered. The bottom section

shows the current stores selected from those specifications. Each time the project is run, it generates the stores selected from the store specifications.

For example, assume you use a discovery specification and have two stores on the network using that discovery address. The bottom part of the page will show two stores selected. If you plug in another Taurus (with that discovery address) and then run it, it will show three stores selected.

3.4.1 Adding stores to a project

- Choose one of the 5 Store Selection methods to add a store to a project.

  The store is included in the project as an item under its Selected Specification. All stores from all selected specifications are listed in the Selected Stores panel, and all of these stores will be used by the project. Any combination of store selection methods can be used in any one project.

3.4.2 Removing stores from a project

To remove a store you must remove its data source specification from the project definition.

Note You can't remove individual stores from a data source specification, but you can add a different specification to select a different group of stores.

To remove a single store:

1. Expand the category node in a Selected Specifications panel.
2. Click on the data source specification to select it, and then click Remove.

To remove the group of stores within a singe data source:

1. Expand the category node in a Selected Specifications panel.
2. Click on the data source specification to select it, and then click Remove.

- Expand the category node and choose its data source specification, and then click Remove.

To remove all stores:

- Expand the data source specification and select its store, and then click Remove.

3.5 Defining projects

You customize projects as needed to retrieve the desired data by setting parameters in a project's definition pages. This may be to create a new project (Section 3.1.2 on page 19) or to edit an existing project (Section 3.7 on page 34). Example projects are provided in Section 1.4.

The project definition parameters are grouped on five tabbed pages:

- Store Selection (Section 3.5.1 on page 25)
- Channel Selection (Section 3.5.2 on page 27)
- Time Selection (Section 3.5.3 on page 29)
- Output Format (Section 3.5.4 on page 32)
- Output Location (Section 3.5.5 on page 33)

Time Selection methods are specific to the project type; the remaining definition parameters are generic. See also Section 3.2 “Application behaviour parameters and project defaults” on page 20.

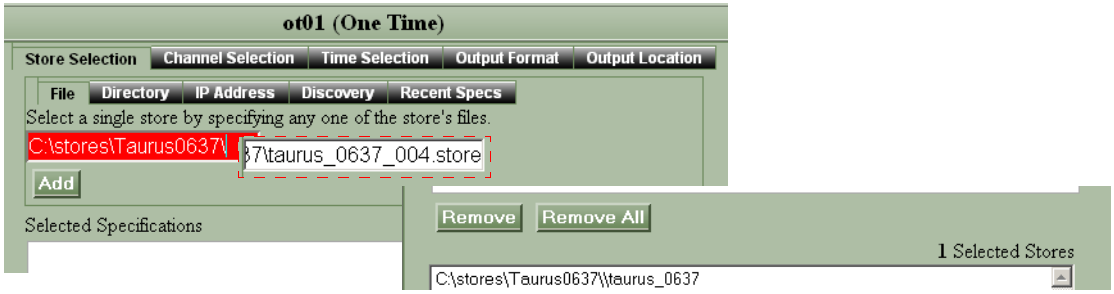
3.5.1 Store Selection

Store Selection is specifying which stores are to be used in your project. You can add stores to a project, and remove stores.

- There are 5 store selection methods:
  - File – Enter the path and filename of any of the store’s files to select the store
  - Directory – Enter a directory path; all stores in that directory and its subdirectories are selected
  - IP Address – Enter the IP address of an instrument on your network to select its store
  - Discovery – Enter a multicast address; all instruments’ stores in that group are selected
  - Recent Specs – Choose an existing, recent store specification from a list

3.5.1.1 File

- Enter the path and filename of any one of the store’s files, and then click **Add**.

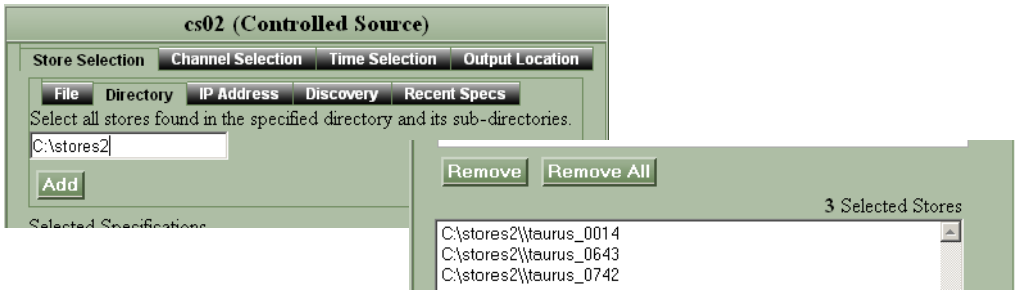


3.5.1.2 Directory

Apollo Project will find and select all stores in the specified directory and its subdirectories.

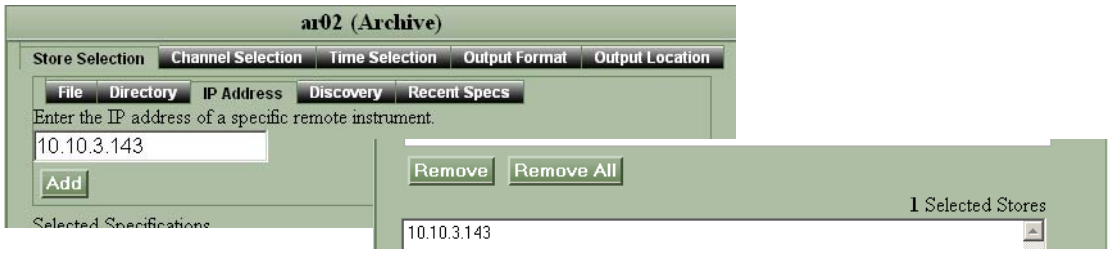
Note Using a high level directory, such as the root directory of a file system, will result in a very long search.

- Enter a directory pathname, and then click **Add**.



3.5.1.3 IP Address

- Enter the IP address of an instrument, and then click **Add**.



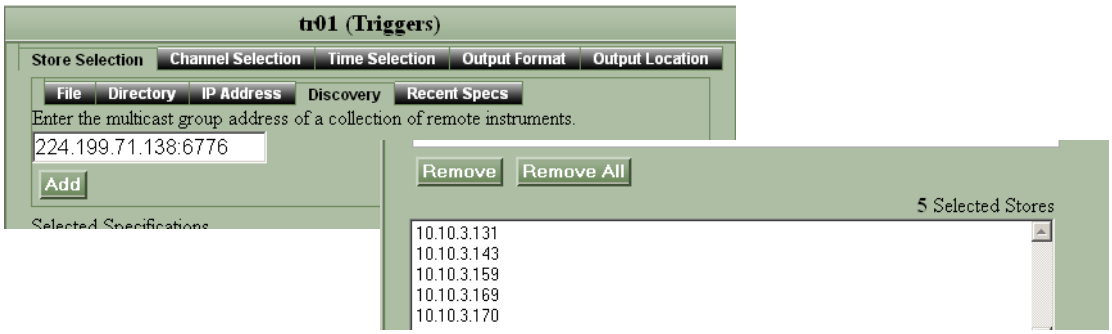
3.5.1.4 Discovery

Apollo Project will find and select all instruments' stores in the specified group.

- When multiple multicast groups are used for discovery, the port number must differ from existing discovery port numbers by 2. That is, for each discovery port number in use *x*, there must be no other discovery port configured for use with a value of *x* or *x*+1.

  For example, the default discovery multicast group used by Taurus is 224.199.71.138:6776. To add another multicast group you must also choose a different port number, which cannot be 6776 or 6777.

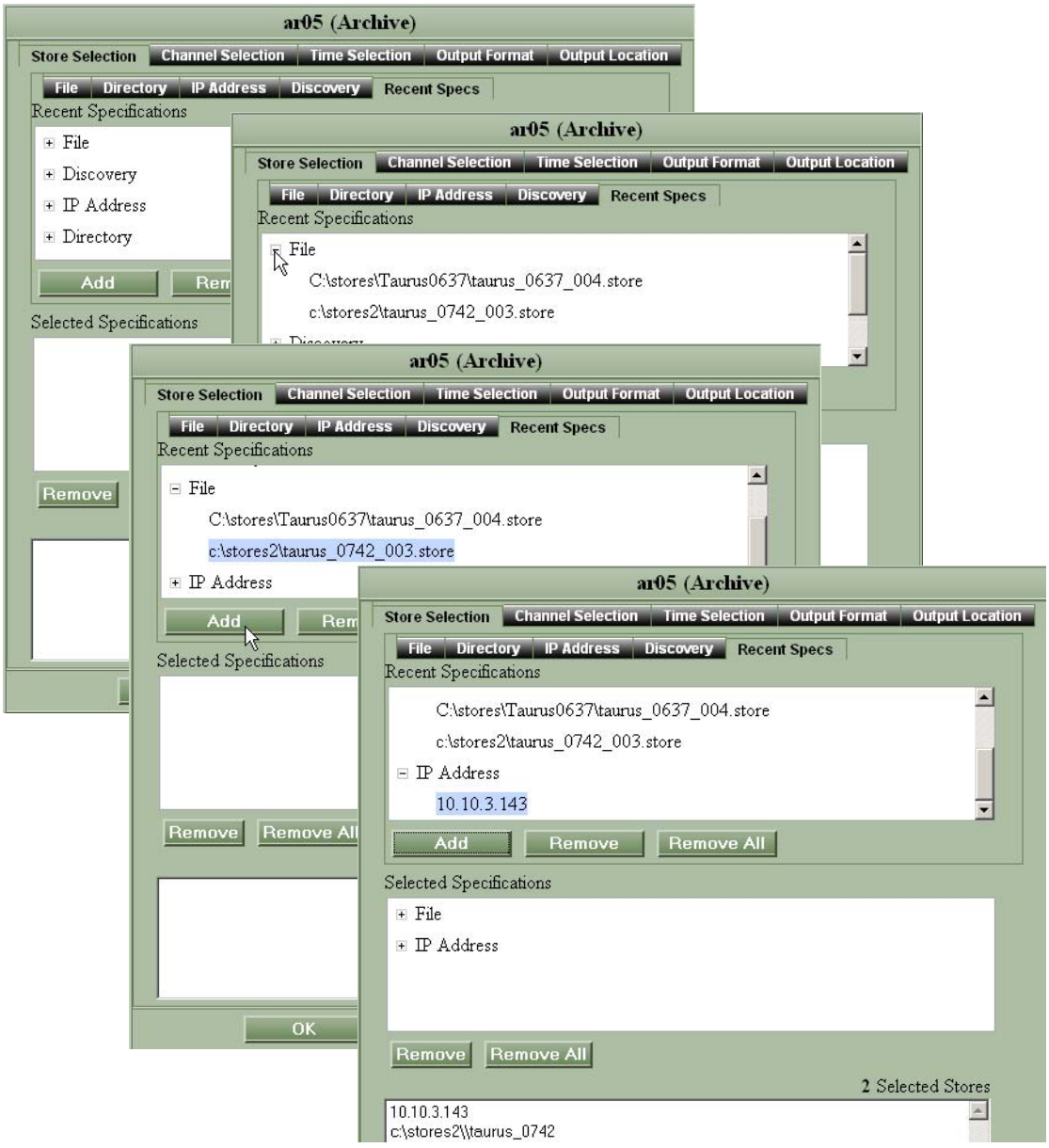
- Enter a multicast group address, and then click **Add**.



3.5.1.5 Recent Specs

Apollo Project keeps the 16 most recently defined store specifications; these specifications are listed in a tree.

- Expand a tree node and choose a store specification, and then click **Add**.



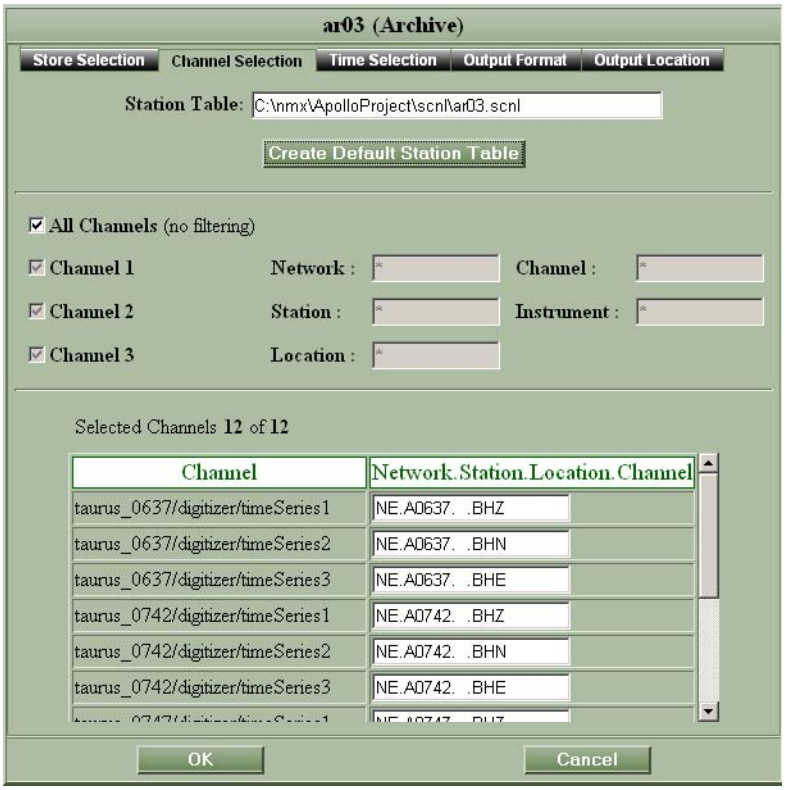
3.5.2 Channel Selection

The Channel Selection page provides options for you to define the channels for which the project will extract data (Figure 3-5); the options are listed in Table 3-3. A Station Table is used to assign channel names to all of the selected channels. The default station table is the one defined by Station Table File Name in the Behaviour and Defaults page.

Table 3-3 Channel Selection parameters

Parameter	Options
Station Table File Name	◆ Use the default value as defined on the Behaviour and Defaults page (Station Table File Name, Table 3-1). ◆ Enter the path and filename of the station table to be used for this project. ◆ Click **Create Default Station Table** to create a default station table for all of the currently selected stores and discovered instruments. The file is saved to *APOLLO_LOCATION*\scnl*project_name.scnl* For the default station name, Apollo Project uses Taurus A*serialNumber* and Trident305 R*serialNumber*. These are put into the MiniSEED files.
Filter parameters	◆ All Channels (no filtering) ◆ Deselect All Channels to use the channel filter parameters. • Select any one or more of the channels listed. • Enter any additional filter information (Network, Channel, Station, Instrument, Location) as comma-delimited text; an asterisk * will represent 1 or more characters.
Selected channels	◆ Displays the list of channels based on the filter selections and shows the selected channels and their associated mappings. You can also edit the mappings.

Figure 3-5 Channel Selection page



3.5.3 Time Selection

Time Selection is the definition of time segments for which to extract the data. There are 4 methods, with each method being specific to a project type:

- Archive
- One Time
- Controlled Source
- Trigger

3.5.3.1 Archive

Archive projects attempt to retrieve data up to the current time. They can be defined to run manually, or automatically on a schedule (Figure 3-6). Archive projects that are defined as scheduled projects can also be run manually at any time.

Note The time ranges for which output files will be produced are driven from the file interval and align to the project start time. It is therefore unlikely that the "most recent" file interval will align with the current time. Therefore the most recent range for which data retrieval is attempted will most likely have only a portion of its data retrieved and the remainder (which extends into the future) will be declared unavailable.

However, when the project is run later (after the end of the particular time range) then the unavailable data will normally be retrieved and the situation will likely repeat itself with the "new" last range to retrieve.

3.5.3.1.1 Manual run

To define a manually-run project:

1. Enter the project start time.
2. Choose the Manual run option.
3. Set the File Interval (the duration of each output data file).

3.5.3.1.2 Scheduled run

There are two methods for defining a scheduled project:

- Weekly Schedule, with one run per day at Project Start Time on any of one or more days per week
- Regular Interval, such as for multiple downloads per day every 7 days

  For a Regular Interval schedule, runs do not start at the Project Start Time. The next run will start (*at the time you click **Start Schedule** + the Interval duration*). If the server has been rebooted, the next run will start (*at the time the server has booted + the Interval duration*).

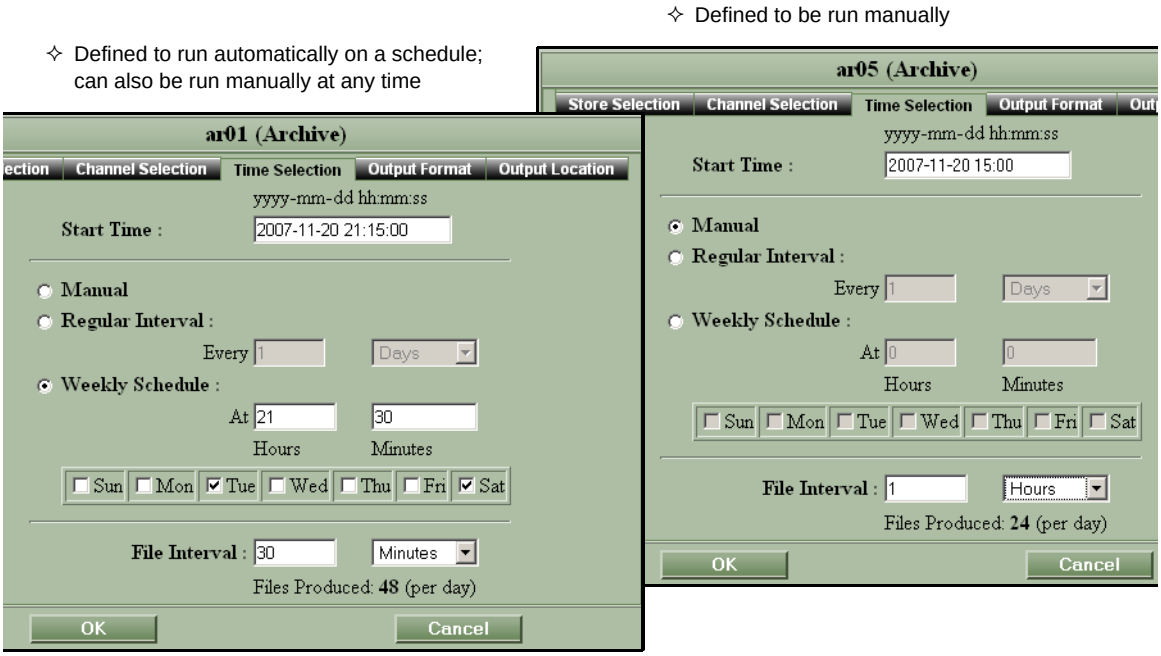
  - If you want the project to run at the specified Project Start Time, use Weekly Schedule.

To define a scheduled project:

1. Enter the project start time.

2. Set the schedule:
   - For a Regular Interval schedule, enter how often the project will run (for example, every 2 days).
   - For a Weekly Schedule
     i. Enter the time of day to start the project run.
     ii. Select on which one or more days of the week it is to run.
3. Set the File Interval (the duration of each output data file).

Figure 3-6 Archive project Time Selection page

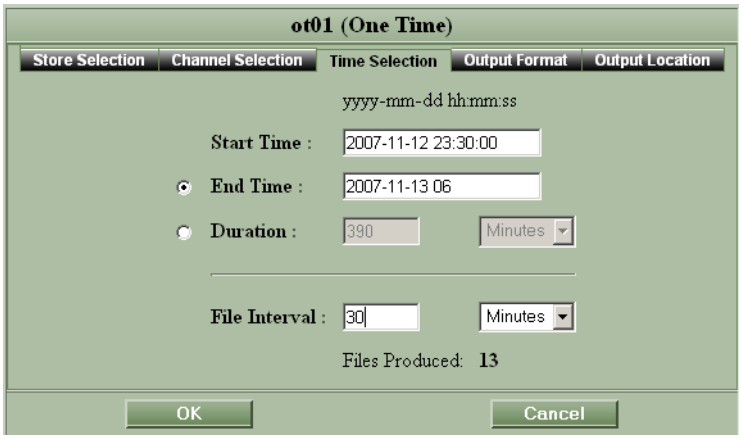


3.5.3.2 One Time

A One Time project will retrieve a single time segment.

- Specify the Start Time, End Time or Duration, and File Interval (Figure 3-7).
  - Duration is the length of the requested time segment.
  - File Interval is the duration of each output data file.

Figure 3-7 One Time project Time Selection page

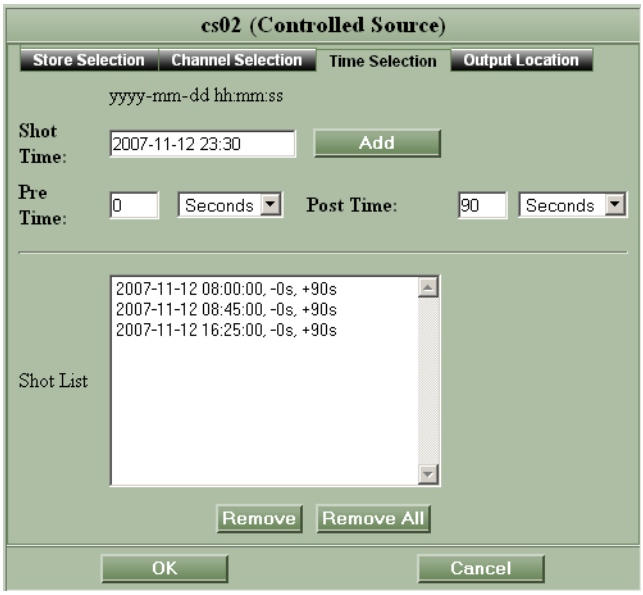


3.5.3.3 Controlled Source

A Controlled Source project will retrieve data specified as a set of individually entered shots.

1. Enter a shot time manually (Figure 3-8).
2. Set the post-shot time period, and optionally the pre-shot time period.
3. Click **Add**.

Figure 3-8 Controlled Source project Time Selection page



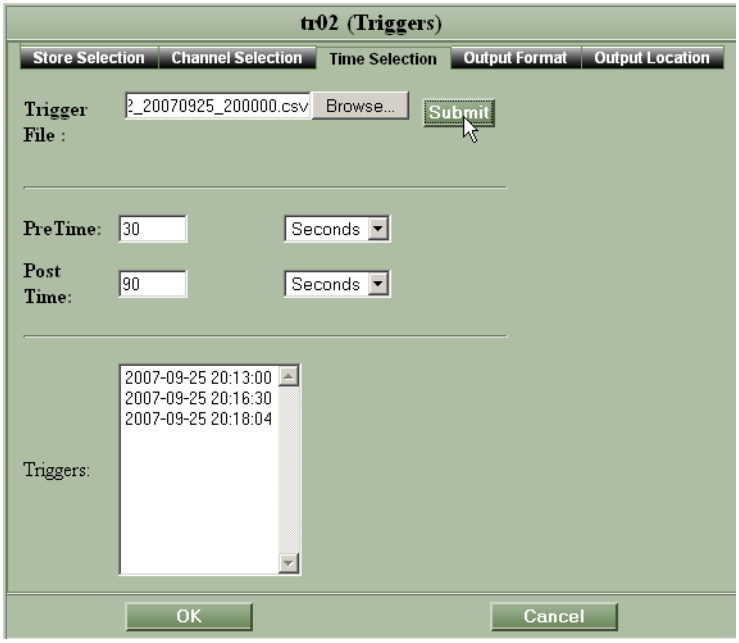
3.5.3.4 Trigger

A Trigger project uses a CSV-formatted trigger list to specify the data to be retrieved, for example a trigger file that has been downloaded from a Taurus to the filesystem.

1. Browse for or enter the trigger file pathname and filename.
2. Click **Submit** to upload the triggers (Figure 3-9). They will be added to the Triggers list automatically.

3. Set pre- and post-trigger time periods. These will be used for all of the triggers.

Figure 3-9 Trigger project Time Selection page



3.5.4 Output Format

Note This page is not available for Controlled Source projects as the output is always combined SEGY format.

You can output the data to any of MiniSEED, Seisan, or SEGY format.

- MiniSEED files are created as individual files for each instrument. The number of files is determined by the file interval (that is, the duration of each output data file). For example, if the time duration requested is 10 hours and the file interval is 1 hour then there will be 10 files created per instrument. The created files will contain all of the selected channels for that instrument.

  MiniSEED is available with common options for data record length; the default is 512 bytes.

  There are options to Combine and to Sort output. You can select either or both of these options.

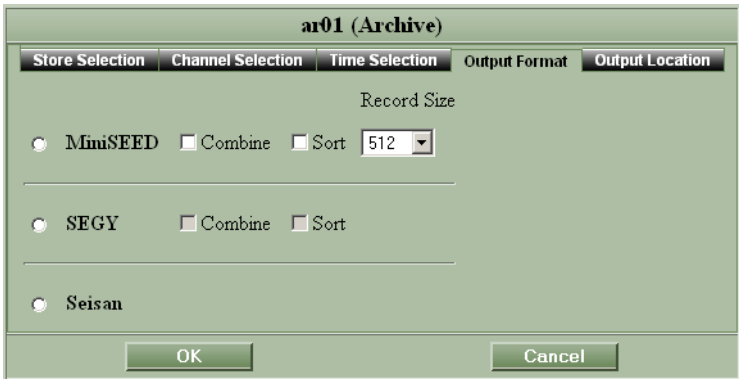
  - Select Combine to combine all stations/instruments into one MiniSEED file.
  - Select Sort to ensure the output data is sorted by time.
- Seisan files are downloaded to the `SEISAN_TOP` directory. (Ensure that you have Seisan installed on your PC, with a `WAVEFORM_BASE` and `CONT_BASE` line included for each station in the `DAT/SEISAN.DEF` file.) The number of output files is determined by the file interval, as described above for MiniSEED.
- ***Output to SEGY format has these 2 requirements:***
  - All stations have the same sample rate
  - The maximum number of samples per trace is 32767

  SEGY trace headers from discovered Tauruses will not have station location information.

There are options to Combine and to Sort output. You can use either or both of these options.

- Select Combine to have all stations combined into one SEGY file.
- Select Sort to have stations sorted on *StationName.NetworkName*.

Figure 3-10 Output Format page



3.5.5 Output Location

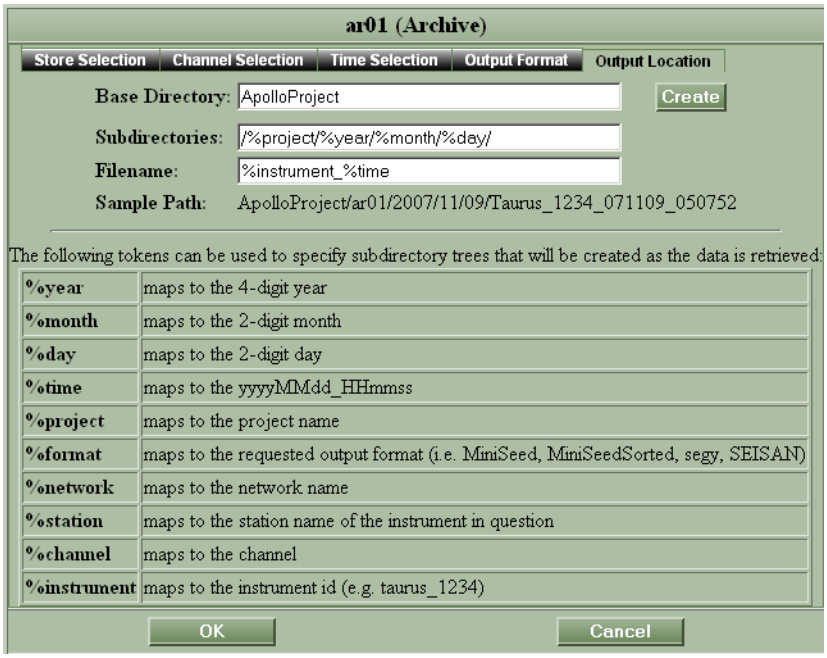
The Output Location page provides options to define the path and filenames for the project output files (Figure 3-11). This page also includes a description of the tokens available to define the output location. You can use any of the tokens in any order, in any of the fields except Base Directory. Optionally, you can include additional characters in the definition. The parameters are listed in Table 3-4.

Note It is not recommended to use the %channel token if the project includes more than one channel per instrument for data retrieval.

Table 3-4 Output Location parameters

Parameter	Description
Base Directory	The pathname of the base directory for the processed data subdirectories and files of this project. ◆ Valid values: any valid absolute pathname on the server. It can also include any additional characters that are valid for paths. Default is the value as defined in the Behaviour and Defaults page (Table 3-1 on page 22).
SubDirectories	The expression used to define pathnames for subdirectories within *Base Directory*. The subdirectoriesare created as needed when the project is run. ◆ Valid values: any of the tokens in any order; it can also include any additional characters that are valid for paths. Default is the value as defined in the Behaviour and Defaults page.
FileNames	The expression used to define filenames for processed data files within *Base Directory**SubDirectories*. ◆ Valid values: any of the tokens in any order. It can also include any additional characters that are valid for filenames except for 2 or more adjacent dots (. .). Default is the value as defined in the Behaviour and Defaults page.
Sample Path	A dynamic example of the resulting path and filename as you enter *Base Directory*, *SubDirectories*, and *Filenames* values.

Figure 3-11 Output Location page



Note In the Subdirectories or Filename fields, avoid using %channel if more than one channel per instrument is selected for data retrieval.

3.6 Viewing project definitions

For a project that has been run or has been scheduled, on the project Summary page, the **Edit** button is replaced with the **View** button.

- Click **View** to open the project definition pages. Click **Cancel** to exit (the **OK** button is inactive in view mode).

3.7 Editing projects

To edit a project definition:

1. Click **Edit** in the project Summary page.

Note The **Edit** function is available only if the project has never been run or, for an Archive project, has never been run or scheduled.

- If the project has been run or scheduled, you can copy it (Section 3.1.5 “Copy a project” on page 20) and then save the copy as a new, editable project.

2. Edit the settings as required in the project definition pages (Store Selection, Channel Selection, Time Selection, Output Format, and Output Location).
3. Click **OK** to save the changes.

3.8 Running projects

For all project types, you can run a project manually at any time. Archive projects can also be defined to run automatically on a schedule.

3.8.1 Run a project manually

- Click **Run Now** to run the project immediately.

3.8.2 Cancel a project run that is currently in progress

- Click **Cancel Run** to stop the current run of the project. You can run the project again with **Run Now**.

3.8.3 Merge data files

At any time for any non-Archive project, you can run merge to combine all the downloaded data for the stations and instruments into one file per interval.

- To merge data into one file, open the project Summary page and click **Merge**. A single file will be produced for each time period. The original data files remain and are not altered.

3.8.4 Running scheduled Archive projects

3.8.4.1 Run a scheduled project automatically

- Click **Start Schedule** to activate the project schedule.
  - For a Weekly schedule, the next run will start at the specified Project Run Time on the selected days.
  - For a Regular Interval schedule, the next run will start at the time you click **Start Schedule** + the Interval duration.

  If you want the project always to start at the specified Project Start Time, use the Weekly option.

3.8.4.2 Run a scheduled project manually

- Click **Run Now**.

3.8.4.3 Cancel and restart a schedule

If you cancel a scheduled project before it runs for the first time, you can edit the project settings. However, if the project is currently running, the current project runs to completion and subsequent project runs will not occur.

- Click **Cancel Schedule** to switch a scheduled Archive project to run manually only. If you want to switch the project back to using scheduled runs, click **Start Schedule**.

3.9 Monitoring project status

Status information is provided as summaries on the Projects, Summary, and the Retrieval Status pages.

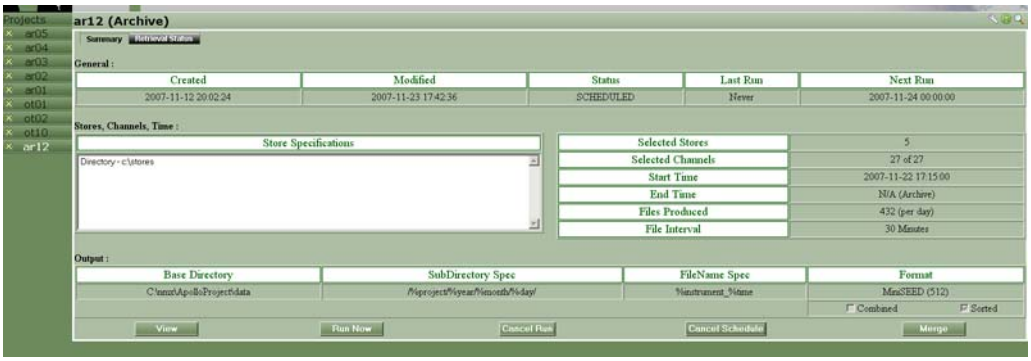
3.9.1 View the basic status of all projects

The Existing Projects page shows a summary of the status of all projects, such as current run status and overall data retrieval success (Figure 3-13 on page 37).

3.9.2 View the status of an open project

The Project Summary page shows status information for the project. This includes its current run status, the last time it ran, and when it will run next if it is a scheduled archive project (Figure 3-12).

Figure 3-12 Project status on the Summary page



3.10 Monitoring data retrieval status

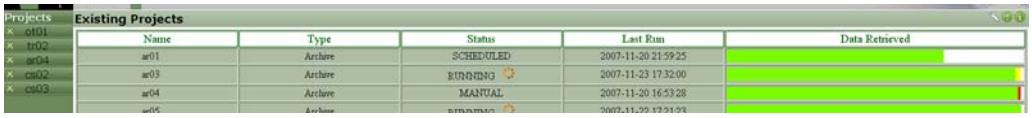
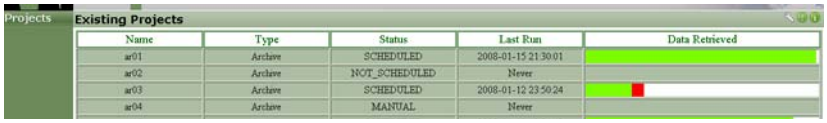
The project Retrieval Status page shows the channel data availability and retrieval status. The information displayed updates continuously while a project is running.

The status for each channel is expressed as a percentage retrieved of the time available and as a percentage available of the total time requested. Bar graphs use the colour codes described in Table 3-5. In addition, retrieval details are available for each channel listed on the project Retrieval Status page (Figure 3-15).

3.10.1 Bar chart status colours

The data availability and retrieval status bar graphs use the colour codes listed in Table 3-5.

Table 3-5 Bar chart colour codes

Colour	Status
Red	Data are available in the store but Apollo Project has failed to retrieve it on this attempt. Apollo Project will make additional attempts to retrieve the data, up to the configured Maximum Number of Retrieval Attempts (Table 3-1). 
Yellow	Apollo Project is attempting to retrieve available data from the store. 
Green	Apollo Project has retrieved the data. 
White	Data for some portion of the requested time segment are not available in the store. For example, there are gaps in the data, or the retrieval request is for a future time segment. 

3.10.2 View the data retrieval status of all projects

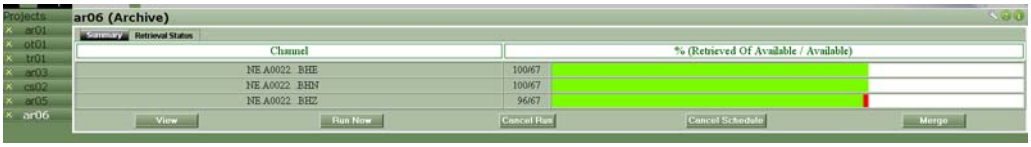
The Data Retrieved bars on the Projects page show the overall status. The status for all channels is expressed as a percentage retrieved of the time available and as a percentage available of the total time requested. (Figure 3-13).

Figure 3-13 Retrieval status of all projects

3.10.3 View the data retrieval status of an open project

The project Retrieval Status page shows the overall status for each channel, numerically and as a bar graph (Figure 3-14).

Figure 3-14 Retrieval Status page for all channels in a project



3.10.4 View the data retrieval details for each channel of a project

The channel details page shows the retrieval status of each time segment for the channel (Figure 3-15).

- To open the channel details page, click on the channel name and then click the Channel Details icon  .

Figure 3-15 Retrieval details for a channel