GRIN-Global Administrator Guide

Revision Date
October 31, 2016

This guide provides an overview to the GRIN-Global Admin Tool and provides other information useful to GG administrators. Change notes pertaining to this document are summarized in the appendix. Refer to the online GG Bug Report for information about known bugs and their "workarounds."

The Table of Contents contains links to the document's sections.

Comments/Suggestions:
Please contact feedback@ars-grin.gov with any suggestions or questions related to this document.

Who Should Use This Guide?

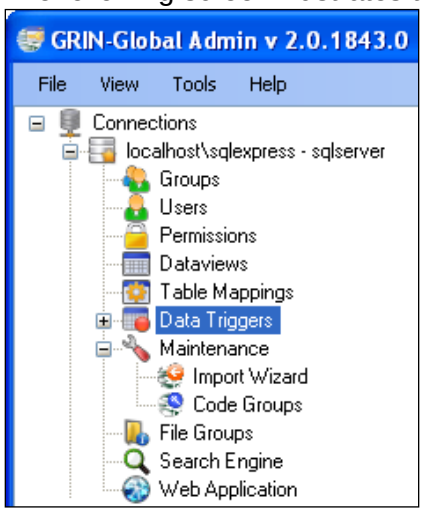
Personnel responsible for administering GRIN-Global should become familiar with the contents of this guide. The guide provides detailed instructions for using the GRIN-Global Updater as well as the GRIN-Global Admin Tool.

GRIN-Global consists of a collection of programs. The Updater is the program used to install or upgrade the other GRIN-Global programs. (Beginning with version 1.9.x, the Curator Tool is installed via a self-extracting exe file.)

The Admin Tool handles many diverse functions. Using the GRIN-Global Admin Tool, an administrator can:

- add user accounts and edit GRIN-Global users' settings such as passwords, permissions, etc.
- review, edit, and create dataviews which are used in the Curator Tool, the AT's Import Wizard, and the GRIN-Global Public Website
- use Table Mappings to associate a dataview field with a table field
- import .dll files (see the *Data Triggers* section)
- use the Import Wizard to import data into GRIN-Global as well as review existing data
- add / edit / delete GRIN-Global Code Groups
- configure the GRIN-Global Public Website settings

The following screen illustrates the main Admin Tool components:



Administrator Guide –Table of Contents

Who Should Use This Guide? 2
Introduction to the GRIN-Global Suite of Applications 8
Main GRIN-Global Software Components 8
GRIN-Global Documents and Resources 9
Data Dictionary 9
GRIN-Global Schema Overview 9
GRIN-Global Database 9
Schema Goals 9
Discoverability 9
Auditability 10
Multilingual 10
Multiple Database Engine Support 10
Consistency 11
System-Related Tables 12
Friendly Names 12
GRIN-Global Schema Tables & Fields 12
Schema Changes 12
Installing GRIN-Global (Updater) 13
Updater Program 13
Updater Options 14
Tools Menu 15
Check for New Updater 15
Delete Cached Files 15
Database Engine Information… 15
Tools | Options 16
General 16
Download Cache 16
System Info 17
Installer CD 18
Mirror Server 18
GG User Files 19
Overview 19
CT User Files 20

AppSettings.txt 20
Customized Treeview Name 21
Treeview Virtual Node Definitions 21
Application Setting Definitions 22
ReportsMapping.txt 22
Reports Displayed in the Order Wizard Dropdown (Example) 23
Creating Reports Using Crystal Reports 23
WebServiceURL.txt 24
Admin Tool Overview 26
Opening Screen 26
Database Connection 27
Server 27
Windows Authentication 27
Test Connection 27
GRIN-Global Login 28
Admin Tool Menu 28
File | New Connection 28
View | Status Bar 29
View | Refresh 29
Tools | Clear Admin Cache 29
Tools | Options 30
Auto Synchronize By Default Checkbox 30
Show SQL for All Database Engines Checkbox 30
Adding Users 31
Default Users 31
Adding a New User to GRIN-Global 31
General and Contact Information 32
Creating a User Name and Password Recommendations 32
Enabled 33
Active Users 33
Web Login Tab 33
Assigning a Web Login for Internal (Genebank) Users 34
To associate a Web Login with a CT User account 36
To Reset a Public Website User Password 36
Contact Info Tab 37
Geographic Tab 37

Site Code 38
Notes....39
Adding Users to Groups 39
Groups....40
Creating a New Group 42
Security....45
Administrator Security Settings 45
Security Overview....46
Ownership....46
Permissions 46
Permission Defined 47
Disabling Permissions 47
Setting Up a New Permission....48
"Applies to Dataview" vs. "Applies to Table" 49
Setting Restrictions at the Row Level (Add …New Restriction) 50
Globally Enabling and Disabling Security 51
Same Computer Requirement for Certain Administrative Rights....51
Dataviews 52
Dataviews: Overview....52
Dataview Fields 53
Transform? 56
User Interface 57
Drop Down Source & Lookup Picker Source 57
Parameters 60
Data Types....61
Properties 61
Titles and Descriptions....62
Preview....63
Importing and Exporting Dataviews 63
Exporting Dataviews....64
Importing Dataviews....65
Importing a Dataview File Containing Multiple Dataviews....65
Language Checkboxes....66
Creating New Dataviews 67
Determining Where Lookup Tables are Used 68
Dataview – Show Dependencies 68

Table Mappings 69
Table Mappings Overview 69
The Concepts of “Mapped” and “Unmapped” 69
Field Context Menu Options (under Table Mapping) 71
Table Mappings: Field Detail Window.....72
Table Mappings | Relationships.....75
Data Triggers 76
Import Option 77
Maintenance 79
Import Wizard 79
Code Groups 80
Background Information 80
Code Groups Editor.....80
Extracting All Codes for a Specific Group Name 83
Extracting All of the Code Values.....84
File Groups 86
Overview 86
Web Application Settings 86
Public Website.....86
Parameters and Properties.....87
GoogleMapsAPIKey 92
Connection Strings.....92
Other Website Configuration Settings.....93
Tools Menu Option on the Public Website.....93
Graphical Appearance 94
Images.....95
Titles, States, and Country Settings 96
Accession Details Page 97
Database Naming Conventions.....99
Foreign Keys 100
Source Code.....101
GRIN-Global Design Overview.....102
Tiers 103
Client Tier 103
Middle Tier 103
Data Tier 103

Reports 103
Reports in the Curator Tool: General Steps in Producing Reports 103
Location of the Crystal Report (.rpt) Files 104
Public Website Reports 105
GRIN-Global Technical Overview.....105
Design of the GRIN-Global System.....106
Miscellaneous Facts.....107
Text Box Search Fields 107
Text Box Searches 108
Curator Tool: 108
Public Website:.....109
Appendix: SQL for Displaying Dataview Field Names 110
SQL Query for Displaying Tables and Table Fields 110
SQL Query for Displaying Dataviews and Dataview Fields 111
SQL Query for Downloading the Schema 111
Appendix: Changes in this Document 113

Introduction to the GRIN-Global Suite of Applications

GRIN-Global has been developed jointly by the **USDA Agricultural Research Service**, **Bioversity International**, and the **Global Crop Diversity Trust**.

GRIN-Global is a new version of the Germplasm Resource Information Network (GRIN), providing the world's crop genebanks with a powerful, but easy-to-use plant genetic resource (PGR) information management system.

GRIN-Global can be implemented different ways, ranging from a simple genebank inventory application on a single PC to a widely distributed system supporting online user searching and germplasm ordering. GRIN-Global:

- Operates over a network or on "stand-alone" personal computers
- Supports Microsoft SQL Server, PostgreSQL, Oracle, and MySQL databases
- Enables maximum flexibility in delegating user rights

Main GRIN-Global Software Components

GRIN-Global consists of a suite of applications. Refer to the online GG Installation Guide for complete installation directions. Also included in the document are links to the most recent source for the GG Updater file and other software.

Note: Versions of the Curator Tool before version 1.9.x were installed via the GG Updater; beginning with 1.9.x, the CT is installed via its self-extracting installation file.

The following list summarizes the main components:

- **GRIN-Global Web Application (also known as the "middle tier")**
  Web site and web services used for interacting with all GRIN-Global data. Makes calls into the database and search engine as requested.

- **GRIN-Global Admin Tool**
  Windows application for administering GRIN-Global. Requires a direct database connection.

- **GRIN-Global Curator Tool**
  Windows application, designed primarily for curators and genebank personnel, for viewing and manipulating GRIN-Global data. Uses GG web service.

- **GRIN-Global Search Tool**
  Windows application for performing searches of GRIN-Global data. Uses GG web service (which in turn calls search engine).

- **GRIN-Global Updater**
  Windows application for updating all the GG server applications. Will also update itself as needed.

GRIN-Global Documents and Resources

Besides this Administrator's Guide, there are other resources documenting the GRIN-Global system. Please refer to the GRIN-Global wiki pages, especially the Training page. where there are many links to documents and resources including additional training exercises and videos. The primary document that Curator Tool users should read is the Curator Tool User Guide, also available o the training page.

Data Dictionary

The GRIN-Global Data Dictionary is an online google doc.

Note

The dictionary details information on the dataviews (and their tables) of primary interest to genebank personnel who input and view the GRIN-Global germplasm curatorial data. This dictionary does not include information about system-type tables.

GRIN-Global Schema Overview

GRIN-Global Database

GRIN-Global requires underlying database management software. Four open-source database software engines are currently supported by GRIN-Global. GG data can be stored in any of the following: Microsfot's SQL Server, MySQL, PostgreSQL, and Oracle.

Schema Goals

GRIN-Global borrows a considerable amount of its conceptual schema from the GRIN system. However, several changes have been made to accommodate goals of the GRIN-Global schema. These goals include:

- Discoverability – intuitive, non-abbreviated, descriptive names for tables and columns
- Auditability – ability to audit data in all tables
- Multilingual – support for multiple languages simultaneously
- Multiple Database Engine Support – conform to database engine specific limitations for all four database engines
- Consistency – overall naming and data typing consistency

Discoverability

The following conventions were adhered to as closely as possible when naming tables and fields (in English):

- Table and field names use full words instead of abbreviations whenever possible
- Tables that are related logically usually begin with the same word. e.g.: accession, accession_name, accession_action, etc.

Prefix	Tables
accession_	Accession
inventory_	Inventory
order_request_	Order
taxonomy_	Taxonomy
crop_	Crop
web_	Public Website
sys_ and app_	GRIN-Global System

- Prefixes and/or suffixes are applied to field names to logically categorize fields with similar functionality: **is_ / _id / _code / _date** etc.

Auditability

All auditable tables in GRIN-Global contain fields explicitly for auditing purposes:

- Who created the record (created_by) and when (created_date)
- Who last modified the record (modified_by) and when (modified_date)
- Who currently owns the record (owned_by) and when they received ownership (owned_date)

These fields are auto-populated by the Middle Tier when data is being saved. The user cannot override the values in these fields. Auditing applies to the cooperator record of the user who manipulated the data, that is, the **created_by**, **modified_by**, and **owned_by** fields contain cooperator_id values.

Generally, the owner of a record is the same UserID who created the record. However, this is not always true. {More information on ownership can be found in the Curator Tool Users Guide under "Ownership.") Some exceptions are:

- Ownership has been transferred since the record was initially created. This is possible only through the owner explicitly transferring ownership.

- A record exists in sys_table_relationship that defines an "ownership" relationship with a parent table. The "owner" of the newly created record will be assigned the same value as the **owned_by** field value in the parent table record. For example, an accession_action record will be marked as being owned by the owner of the accession, not the creator of the accession_action record itself.

Multilingual

GRIN-Global supports displaying data in multiple languages for system-level data. That is, if the system requires text to be displayed that is not actual GRIN-Global data, that text should be in the appropriate language for the current user. This is accomplished by using a table ending with "_lang" as a child table. Some of the language tables included in GRIN-Global are:

- code_value_lang
- crop_trait_code_lang
- sys_lang
- sys_dataview_lang
- sys_table_field_lang

Multiple Database Engine Support

GRIN-Global currently works with four different database engines. (This is true for version 1.0.7. Interim releases have been requested from the GG development team in the US since 1.0.7 was released. Those interim versions had the supporting dataviews only for Microsoft SQL Server.) Since a genebank may already be invested in a particular database engine (with staff to support it, existing servers, etc.), this flexibility allows GRIN-Global to run in almost any IT environment. The following database engines are supported:

Database Engine (version)*	Related Website
SQL Server (2008)	http://www.microsoft.com/sqlserver/en/us/default.aspx
Oracle XE (10g)	http://www.oracle.com/technetwork/developer-tools/sql-developer/overview/index.html
MySQL (5.1)	http://dev.mysql.com/downloads/workbench/5.2.html
PostgreSQL (8.3)	http://www.pgadmin.org/

* version displayed here or later versions

Note GRIN-Global may or may not work on different *versions* from those listed above.

Consistency

GRIN-Global strives to be consistent in the naming, semantics, and patterns used in the schema. The schema uses the following conventions:

Convention	Description
All tables have a single primary key field.	The key's type is integer; keys are auto-generated.
_id	Primary key field names always start with the table name and end with "_id." Example: **accession.accession_id**
_id	Foreign key field names always contain the foreign table name and end with "_id." Example: **accession.taxonomy_species_id** (taxonomy_species is the foreign table name)
Names are in lowercase	containing only a-z and underscores (_)
_map	Tables which represent a many-to-many relationship usually end with "_map"
_lang	Tables which represent language-specific data always end with "_lang"
sys_ / app_	System-specific tables always start with sys_ or app_
web_	Web-specific tables always start with "web_"
Table names – limited to 27 characters; Field names – limited to 30 characters	Conforms to the Oracle limitation of 30 characters for object names. (Since a table name + "_id" is always used for the primary key field, the max length for a table name is 30 – 3 (length of "_id") = 27
"_date	Fields that contain date and time based data end with "_date" and are stored as an actual DateTime data type
_code	Fields that contain coded values end with "_code" and are always stored as character data with a max length of 20.
is_	Fields that represent Boolean data begin with "is_" Boolean data is always stored as "Y" for yes and "N" for no; always stored as a non-nullable character field of length 1.
	Most primary tables define a sensible unique compound key comprised of the actual data fields

System-Related Tables

System tables are tables GRIN-Global needs to function properly, but do not have anything to do with the actual genebank data. They are broken down into two major sections:

- sys_* tables represent tables required by all applications to function properly
- app_* tables represent tables required by certain applications to function properly

The flexibility of GRIN-Global is due to the data stored in the system tables. As a whole, the system tables represent the following:

- User Interface customization
- Data presentation and manipulation extensibility
- Data workflow customization
- Language localization
- Security – including users, permissions, and groups
- Downloadable content
- Cached information about the schema itself

Friendly Names

The "friendly" names are the headings used in one's own language as compared to table field names (the actual field names as defined in the physical database). A SQL query that lists the "friendly" names used for the table field names is included in the Appendix. See SQL Code on page **Error! Bookmark not defined.**.

Note

GG's default language is English. However, when installed, the GG database includes four other languages: Spanish, French, Russian, and Arabic. Switching the interface to display in one of these embedded languages is quite simple. Furthermore, languages can be added to the database. For complete instructions for adding another language to GG or for editing column headings, refer to the separate guide: http://www.ars-grin.gov/npgs/gringlobal/docs/gg_language_notes.pdf

GRIN-Global Schema Tables & Fields

The GRIN-Global data dictionary is available online at GG-dictionary. The dictionary contains table and field names as well as descriptions of every field in the GRIN-Global schema.

There is some sample SQL in this guide's appendix, for downloading the GRIN-Global dataview / field names.

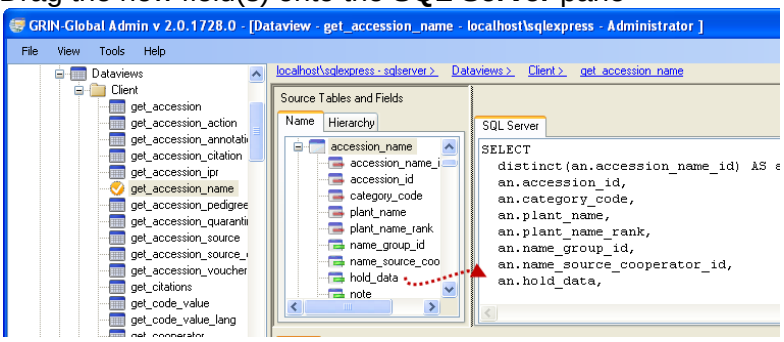
Schema Changes

GRIN-Global is designed to be very flexible and meet the needs of many genebanks. Much of the schema was designed form the U.S. GRIN schema, with modifications and suggestions contributed by many international users. If there are fields that your organization will be not using, it is recommended that you leave them in the schema. You can modify dataviews in the GRIN-Global Admin Tool so that the fields are never displayed.

If you need additional fields, you will need to use your database management tool such as SQL Server Management Studio, Oracle, etc. to add the new field to a table. You then will use the Admin Tool to incorporate that field. The general steps to follow are:

1. Make schema change in SSMS (or whatever database tool you are using)

2. In the Admin Tool, run **Table Mappings | Inspect Schema for Unmapped Tables…**
3. Load the dataview in the Admin Tool Dataview Editor
4. Drag the new field(s) onto the **SQL Server** pane



5. In the Dataview Editor, edit the field friendly name(s).

Note

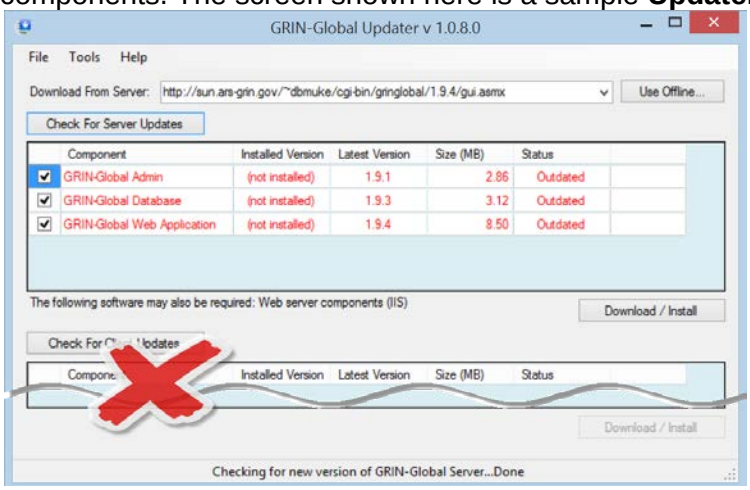
When any change is made to GRIN-Global, the date-time stamp of the transaction is not based on the PC's clock but on Greenwich Meant Time (GMT).

Installing GRIN-Global (Updater)

Updater Program

The first step in installing the GRIN-Global program suite is to install the **Updater** program. When you are ready to install the Updater program, refer to the GRIN-Global Installation Guide for detailed directions. However, before proceeding with the installation, we recommend that you read the remaining part of this overview to get familiar with the general features of the Updater.

After the **Updater** program is installed, you will use it to install GRIN-Global's server components. The screen shown here is a sample **Updater** window:



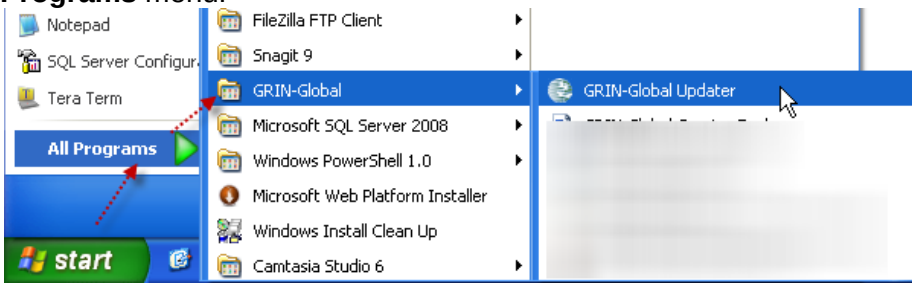
Note

Beginning with version 1.9.x, the Curator Tool is installed via its self-extracting .exe file; the GG Updater program is *not* used to install the CT.

Installing GRIN-Global (Updater)

For users who will only be using the GG Public Website, no installation is necessary. The user's browser needs to be pointed to the URL of the organization's GG website.

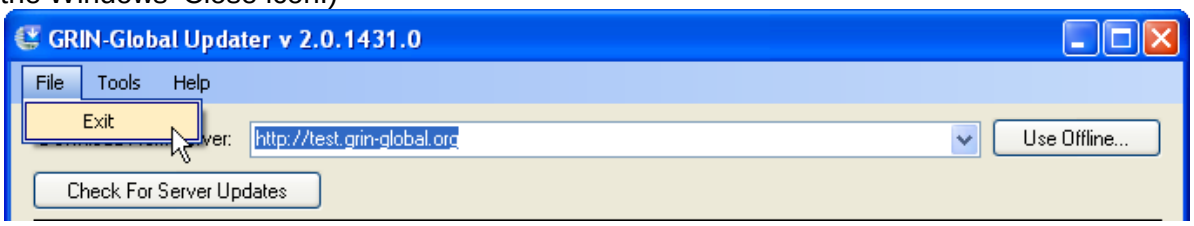
When the Updater is installed properly on your PC, its menu option will display on the **Start Programs** menu:



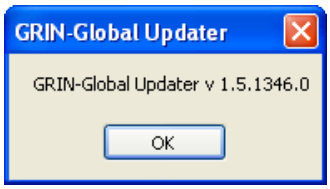
Updater Options

There are three items on the Updater menu – **File**, **Tools,** and **Help**.

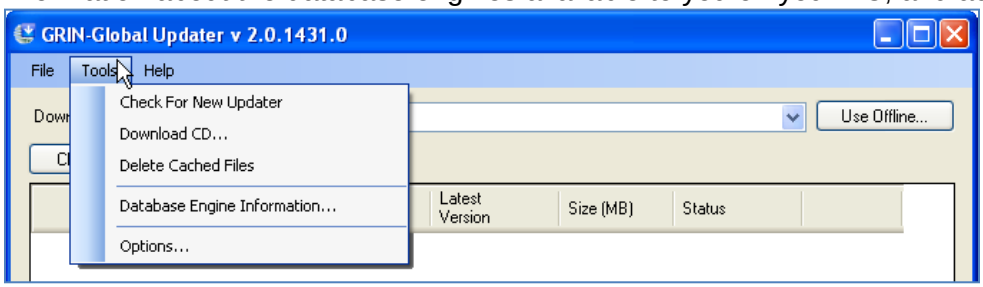
File has just one option: **Exit**. Use this option to stop running the **Updater.** (You can also use the Windows' Close icon.)



(Since Help is so simple, we'll explain it before discussing Tools.) Use **Help** to determine the version.



Under **Tools**, you can check for the latest version of the Updater, delete any cached files, obtain information about the database engines available to you on your PC, and access other options.



Tools Menu

Check for New Updater

Select **Tools | Check for New Updater** to download the latest version of the Updater. However, this option should not be necessary since the Updater automatically checks for a new version when it is invoked.

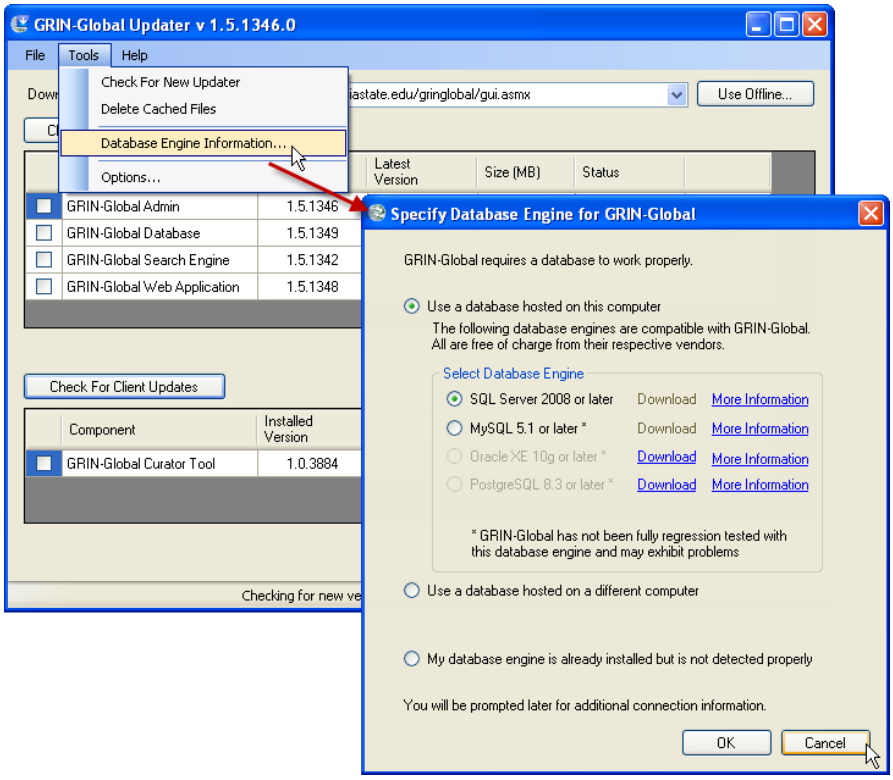
Delete Cached Files

When you click the **Download/Install** button on the Updater window, you start the downloading of files into a "download cache." The downloaded files vary, depending on which component(s) you select to download. Eventually, when the files have been successfully installed, they are moved from the download cache to an "installer cache." This "**Delete Cached Files**" option frees up the hard drive space used by these cache files.

(The "Delete Cached Files" option is equivalent to the **Tools | Options | Download Cache | Delete Selected** command.)

Database Engine Information…

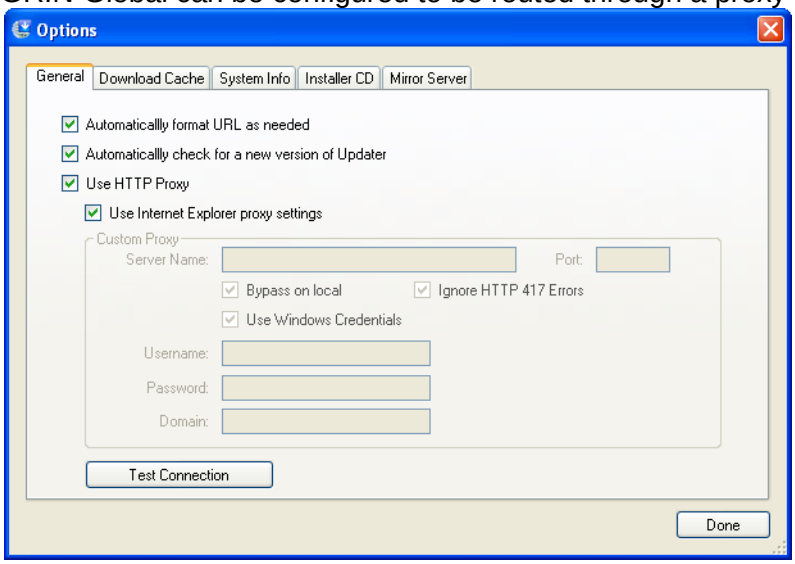
Use this option to either switch to a different database engine or to repair the connection to the current database engine when the engine is not functioning properly.



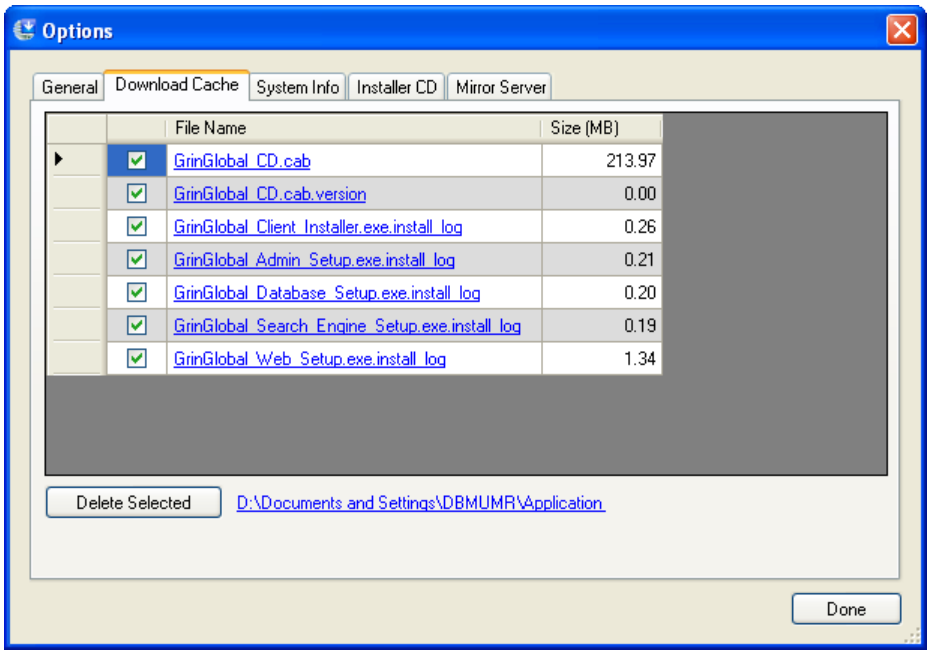
Tools | Options

General

GRIN-Global can be configured to be routed through a proxy server.



Download Cache



When you click the **Download/Install** button on the Updater window, files begin downloading into a “download cache.” The files downloaded vary, depending on which component(s) you select to download. Eventually, when the files have been successfully installed, they are moved from the download cache to an “installer cache.”

The download cache makes it possible to resume from the point where the last download stopped if the download doesn't fully complete.

In this screen, you can delete any files that have downloaded, but which are not yet installed.

When the download has completed, the Updater runs the files from this download cache folder (i.e., ".exe" files are executed and the ".msi" files are installed.) After the files have completely installed, they are moved from the download cache to an "installer cache."

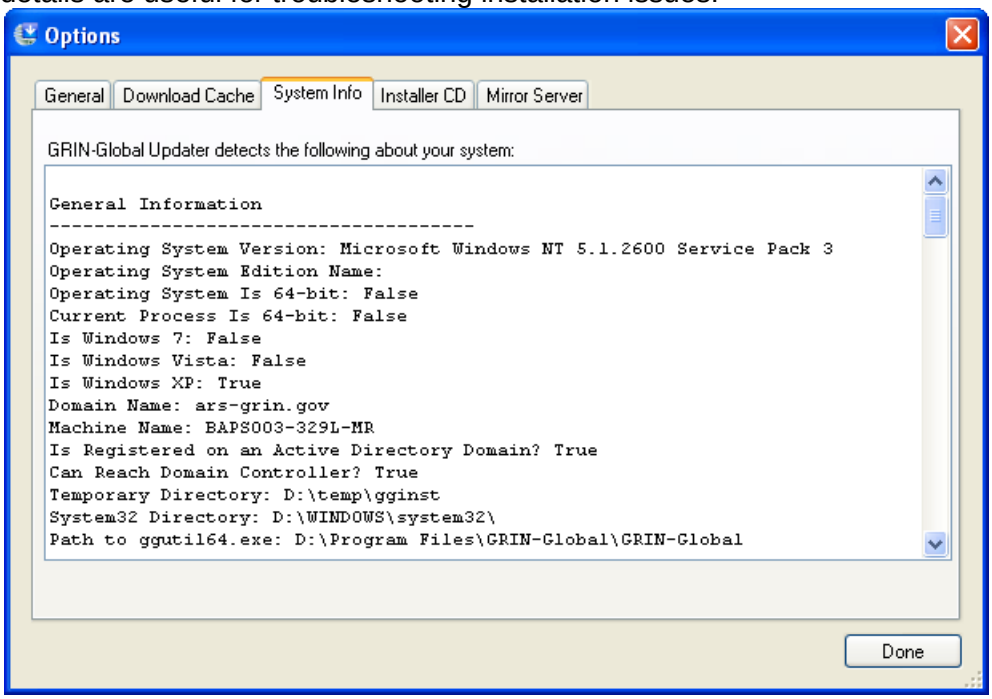
Note

The installer cache is not shown anywhere in the Updater interface. It is in a backup area. If problems do arise, the files in this cache could be manually reinstalled to load the last successfully updated version.

To show the download cache in Windows Explorer, click the link next to "Delete Selected." From there, "move up" one folder (to the parent folder) to display an "installed" folder. The installer cache resides here.

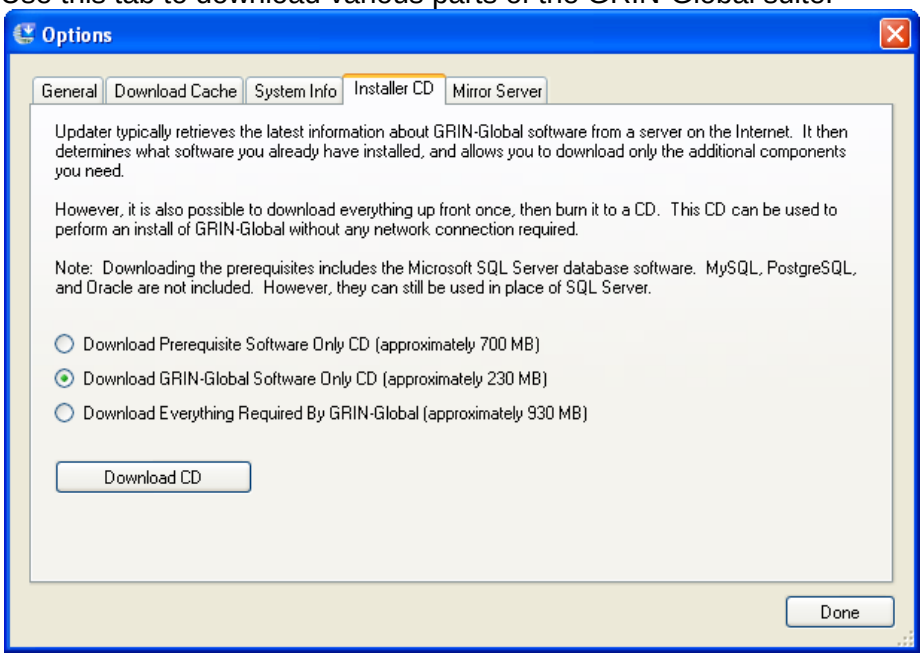
System Info

Details are displayed about the user's local PC as the Updater program sees them. These details are useful for troubleshooting installation issues.



Installer CD

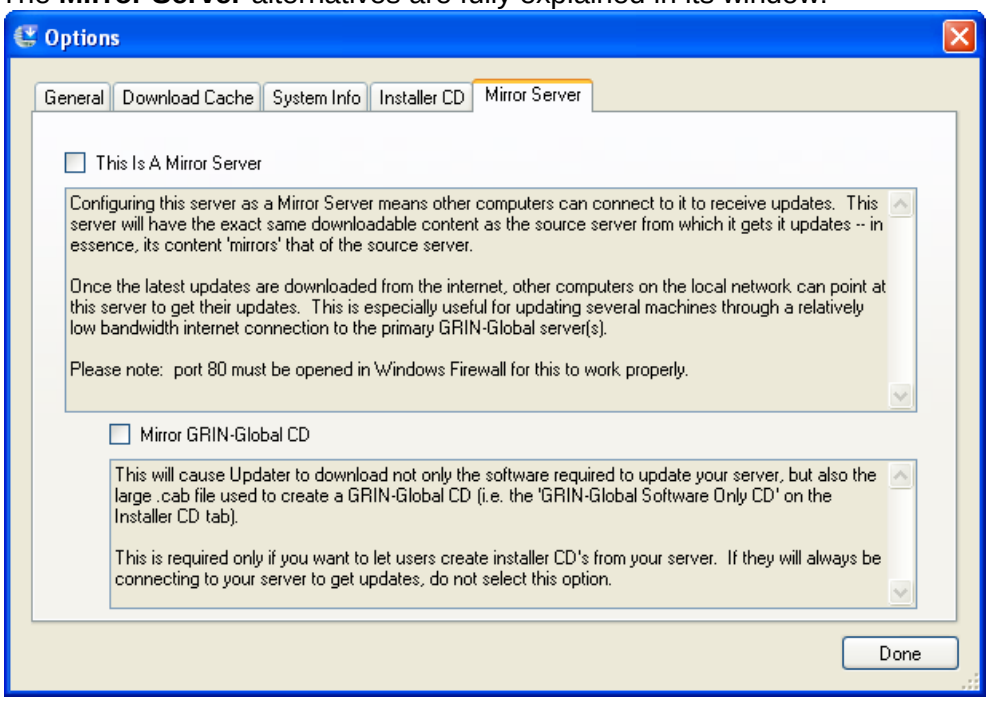
Use this tab to download various parts of the GRIN-Global suite.



Refer to the Installation Guide for aditional details.

Mirror Server

The **Mirror Server** alternatives are fully explained in its window:



The mirroring doesn't add any overhead if the **Mirror GRIN-Global CD** checkbox isn't checked. It mirrors things after it installs them, so the files have already been downloaded. The exception to this is when the **Mirror GRIN-Global CD** checkbox *is* checked. Then a very large (120 MB) file is downloaded.

GG User Files

Overview

When the GRIN-Global Curator Tool is installed on a User's PC, several files are installed on the PC's hard drive which affect user settings. The advantage of these files is that they can be configured by a GG administrator (or a user).

Several folders are installed during the CT installation:

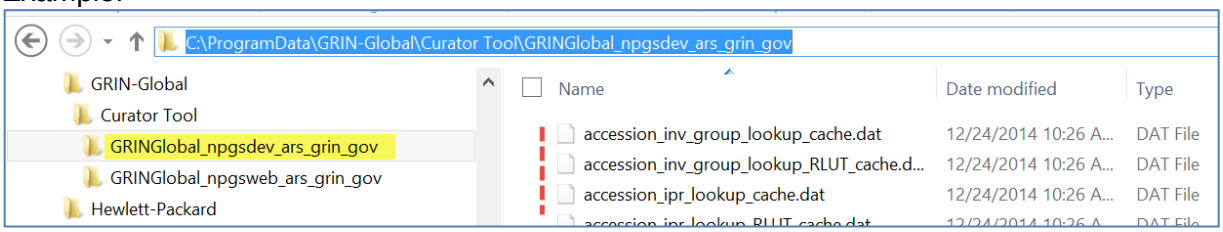
- C:\ProgramData\GRIN-Global\Curator Tool
  (3 .txt files are stored here – see the note below)

Note

Three files are placed into the **C:\ProgramData\GRIN-Global\Curator Tool** folder when the CT is installed and are considered are considered "master copies." The three files: **AppSettings.txt**, **ReportsMapping.txt**, and **WebServiceURL.txt**. are placed there during CT installation. When the user starts the Curator Tool *the first time*, the master copies of those three files are then copied from the **ProgramData…** directory to the **username…** directory (such as **C:\Users\MartyR\AppData\Roaming\GRIN-Global\Curator Tool**)

- C:\ProgramData\GRIN-Global\Curator Tool\GRINGlobal_*servername*
  (current _cache.dat files are stored here)

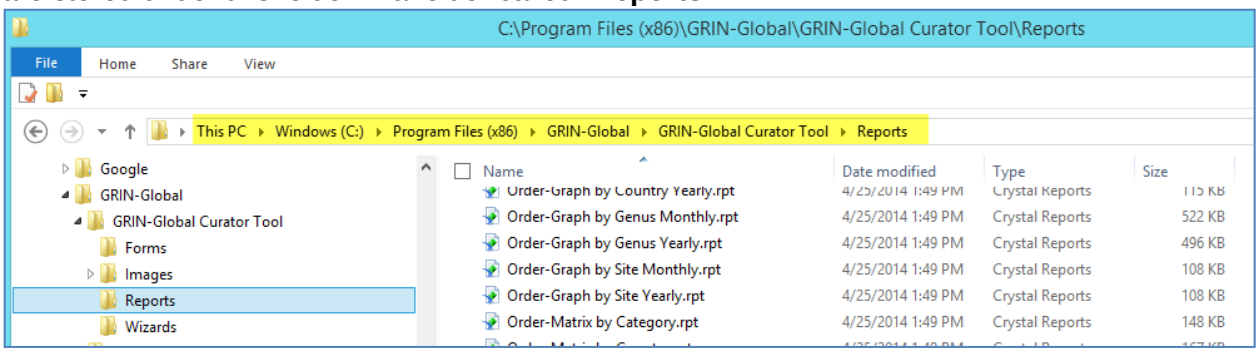
  Example:



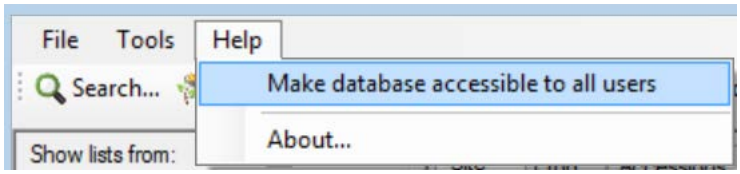
- Related: [A folder not created during the CT installation (but created and populated the *first* time the user starts the Curator Tool)
  C:\Users*username*\AppData\Roaming\GRIN-Global\Curator Tool

  (The 3 working .txt files are stored here)

- **C:\Program Files (x86)\GRIN-Global\GRIN-Global Curator Tool**
  Among other folders and files, the .rpt files for reports to be used by the Curator Tool are stored under this folder in a folder called "**Reports**"



The user must have read/write privileges for these folders. A GG administrator who installs the CT on the user's PC can use the CT's Help command to "**Make database accessible to all users**"

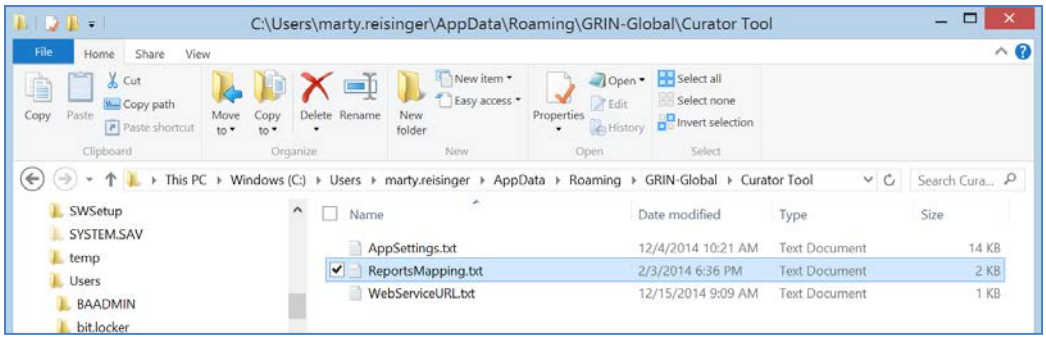


Note

This Help command does not (yet) adjust the user read/write settings for the **C:\ProgramData\GRIN-Global\Curator Tool\GRINGlobal_*servername*** folder. To manually adjust the **Program Data…** folder's settings, log on as the PC's *Windows* Administrator and adjust the **Properties | Security** settings so that the user can Read, Write, Modify.

CT User Files

File	Purpose
AppSettings.txt	Controls certain CT user settings such as application-level settings for the GUI controls
ReportsMapping.txt	Associates a .report file (".rpt") with a dataview file (".dataview")
WebServiceURL.txt	This file indicates which servers are available to the CT



AppSettings.txt

The AppSettings.txt file controls certain CT user settings such as:

(1) customized treeview names
(2) virtual node definitions
(3) a few miscellaneous application setting definitions.
(4) which reports display in the CT Order Wizard **Printing** dropdown

When the CT is installed on a PC, this .txt file is copied into the following folder: **C:\ProgramData\GRIN-Global\Curator Tool**.

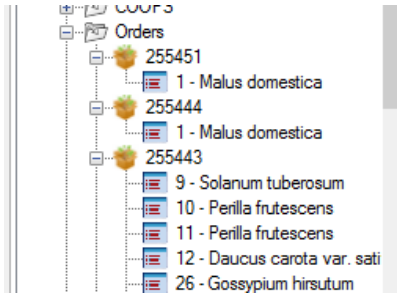
On the server there is also an app_setting table which establishes "default settings" for each user. As the CT loads itself into memory, it will read the app_setting table first and then the

AppSettings.txt file. The AppSettings.txt file entries will override the table entries. The local AppSettings.txt file “trumps” the settings in the app_setting table.

Customized Treeview Name

The list items can be assigned custom names that may include the contents from a field as well as text. In the following example, all order request items in a list will display their sequence number and the taxonomy. The following statement will alter the name of the Order_Request_Item_IDs:

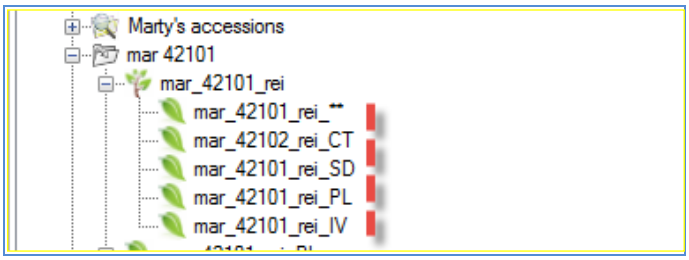
> # ORDER_REQUEST_ITEM_ID_NAME_FORMULA = {get_order_request_item.sequence_number} + " - " + {get_order_request_item.external_taxonomy}



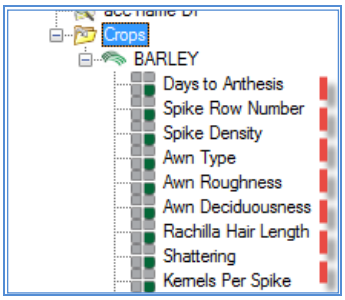
The syntax of the statement is *item*_ID_NAME_FORMULA = {field1} + “text” +…

Treeview Virtual Node Definitions

A “virtual node” can be automatically generated for a list item. For example, when accession items are in a list, their respective inventory items are displayed when the “+” symbol is clicked. In this example, the accession **mar 42101 rei** has five inventory items listed:



CROP_ID_VIRTUAL_NODE_DATAVIEW = get_crop_trait



Application Setting Definitions

A few settings can be set. [tbd]

ux_numericupdownMaxRowsReturned = 10000

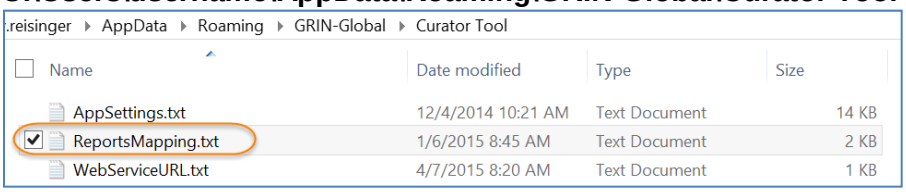
ux_numericupdownQueryPageSize = 1000

ux_checkboxOptimizeLUTForSpeed = true

ReportsMapping.txt

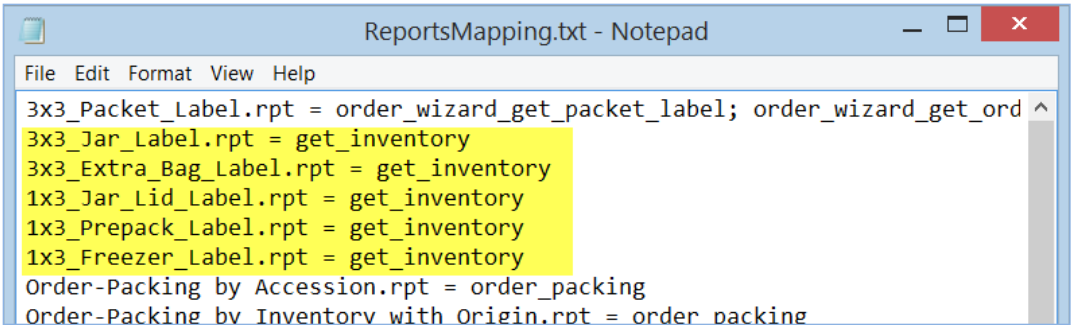
The **ReportsMapping.txt** file must be stored on the user's PC in the folder:

C:\Users*username*\AppData\Roaming\GRIN-Global\Curator Tool

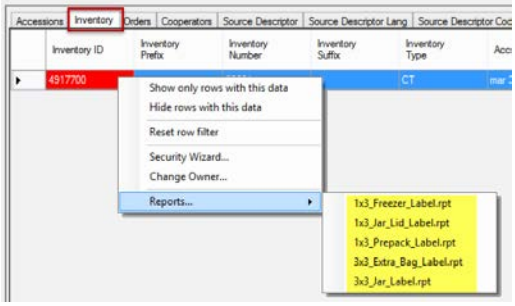


The ReportsMapping.txt file is used to associate a .report file with an ".rpt" with dataview file(s) (.dataview). A report may be designed to work with more than one dataview. By setting up this mapping file correctly, the user will not be inundated with many report files when invoking reports – instead, only the appropriate reports will be displayed in the menu or dropdown.

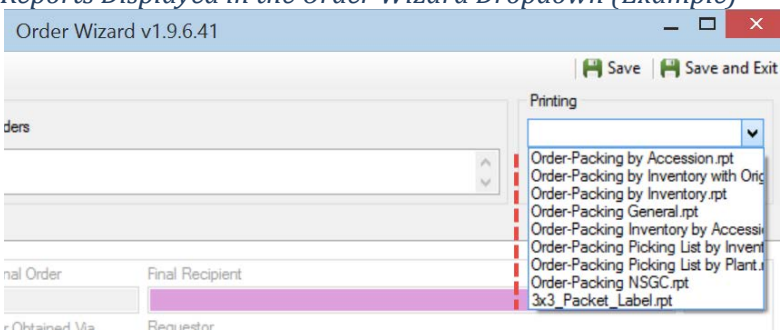
For example, in the ReportsMapping.txt file shown below, there are five reports associated with the get_inventory dataview:



When working in the Inventory dataview, the user will see the five reports listed (in alphabetical order) when he invokes the **Reports…** menu option:



Reports Displayed in the Order Wizard Dropdown (Example)



The statement below, when included in the ReportsMapping.txt file, will display reports in the Order Wizard dropdown:

> OrderWizardCrystalReports = Order-Packing by Accession.rpt; Order-Packing by Inventory with Origin.rpt; Order-Packing by Inventory.rpt; Order-Packing General.rpt; Order-Packing Inventory by Accession.rpt; Order-Packing Picking List by Inventory.rpt; Order-Packing Picking List by Plant.rpt; Order-Packing NSGC.rpt; 3x3_Packet_Label.rpt

Note

If the statement has a **#** preceding it, the # made it a comment – delete the #.

Creating Reports Using Crystal Reports

To create a new Crystal Reports "CR report," you start by saving an Excel file created by exporting some sample data from a CT dataview. This indicates to Crystal Reports what data fields with which it has to work. Later, when using the report in the CT, the CT sends data selected by the user to the CR report in the form it is expecting.

The following are the basic steps needed to create a new report:

- Choose a dataview on which to base the report; drag some sample data from the CT to Excel
- Save that data to a file as type: Excel 97-2003 Workbook (*.xls)
- Open SAP Crystal Reports
- Select Report Wizard
- Create New Connection
- Access/Excel (DAO)
- Database Type: Excel 8.0
- Database name: … browse to the saved .xls file
- Finish

From there it is a matter of selecting fields to use in the report from Sheet1$. Look for tutorials on how to do format your data in Crystal Reports.

After creating the CR report, the saved report file (extension ".rpt") must be stored in the CT report directory on each PC where it is to be used. That directory is:

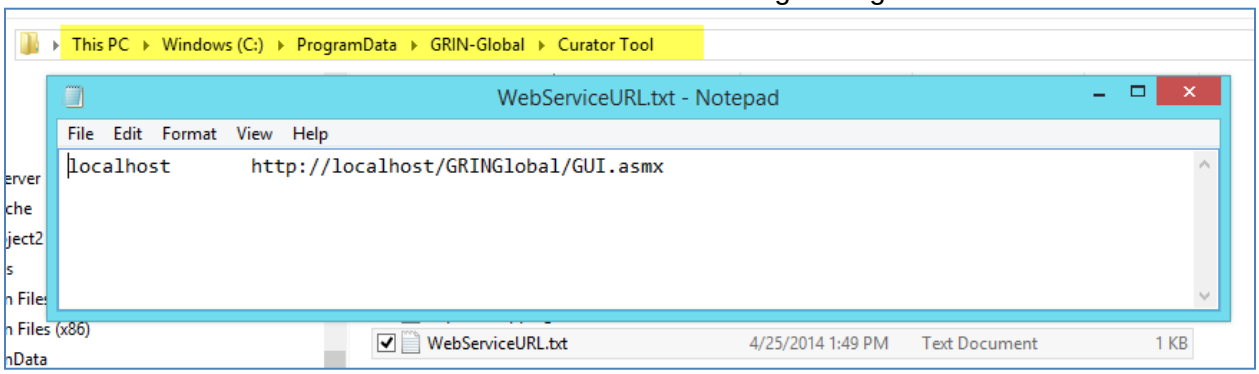
C:\Program Files (x86)\GRIN-Global\GRIN-Global Curator Tool\Reports

The last step is edit a file on the PC which indicates to the CT which report goes with which dataview. This is the **ReportsMapping.txt** file in the folder

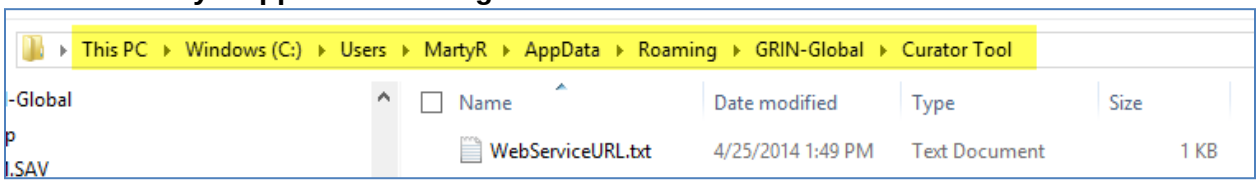
C:\Users*username*\AppData\Roaming\GRIN-Global\Curator Tool (see the ***Reports Mapping*** section).

WebServiceURL.txt

When the CT is initially installed, a copy of this file is installed at **C:\Program Data\GRIN-Global\Curator Tool**. The default file contains the following setting:



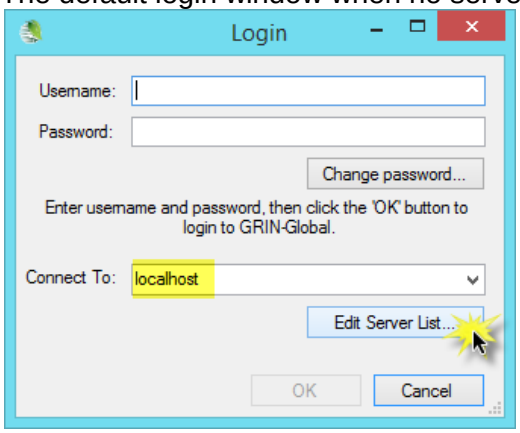
When the user starts up the CT, a copy of this file is copied to the user's CT folder, such as: **C:\Users\MartyR\AppData\Roaming\GRIN-Global\Curator Tool**



When starting the CT, this file controls what GRIN-Global servers display in the Login window. If the user has previously added a server to his list, he can select it from a list of servers.

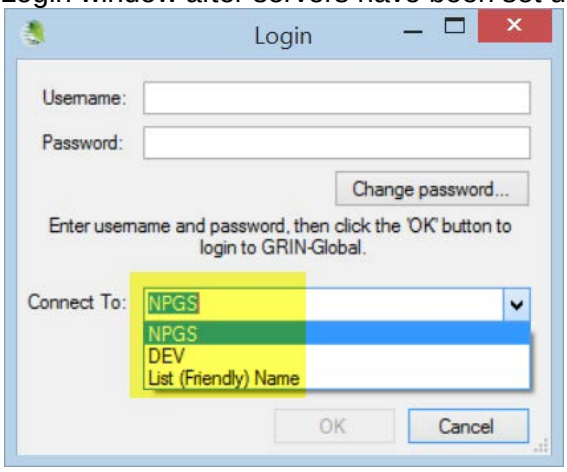
A GG administrator can copy this **WebServiceURL.txt** file onto each user's PC instead of the user or the administrator manually setting up the server connections on each PC via the User login screen.

The default login window when no servers have been set up:

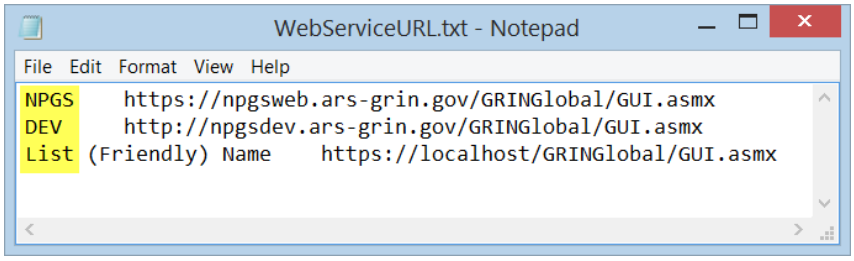


(Click on the **Edit Server List** button to set up.)

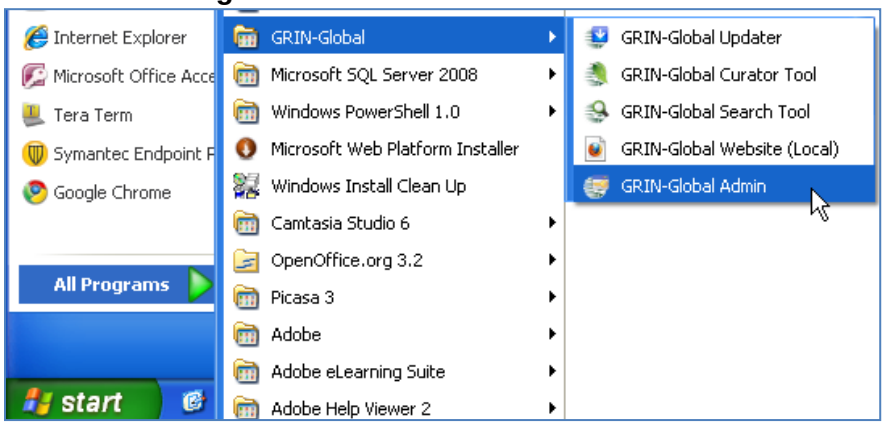
Login window after servers have been set up:



Corresponding **WebServiceURL.txt** file in the user's folder (such as **C:\Users\MartyR\AppData\Roaming\GRIN-Global\Curator Tool):**



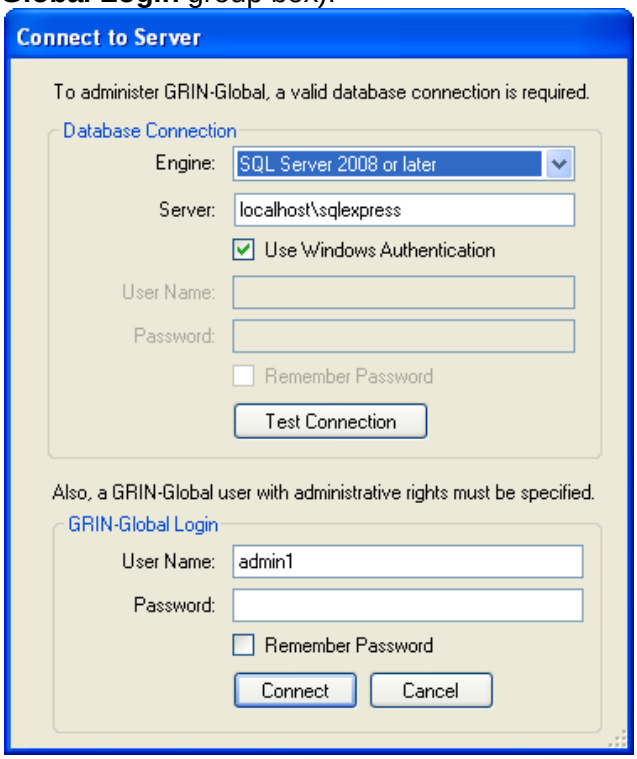
Admin Tool Overview
In Windows 7, when the Admin Tool is installed properly on your PC, its menu option will display on the **Start Programs** menu:



In Windows 8.1, there is a tile for the start screen.

Opening Screen
The Admin Tool's initial screen requires two logins:

- one used for authenticating against the database engine (in the **Database Connection** group box)
- one to authenticate against the security records within GRIN-Global itself (in the **GRIN-Global Login** group box).



Database Connection

Currently, the only GRIN-Global users allowed to use the Admin Tool are those users who are members of the group named “Administrators.”

For the database engines other than non-SQL Server, you should probably provide the same UserID/password that was specified when the database engine was installed. Typical passwords are:

- mysql root
- postgresql postgres
- oracle SYS

Note

It's not required to be the superuser/root userID, but those are always created at install time for those engines.

Server

This is the server name that hosts the database engine. Some engines, such as SQL Server, expect an “instance name” (shown below as “sqlexpress”) as part of the server name. For other engines, the port must be specified. Examples:

SQL Server:	localhost\sqlexpress
MySQL:	localhost:3306
Oracle:	localhost:1521
PostgreSQL:	localhost:5432

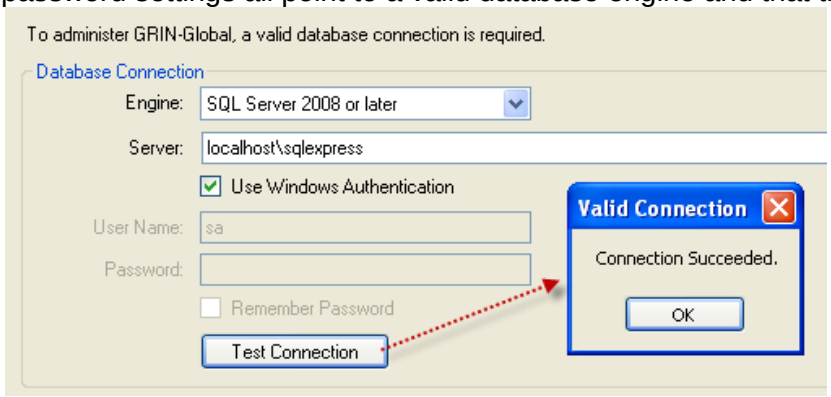
Windows Authentication

Windows authentication should be used wherever possible.

However, only Microsoft SQL Server can use Windows Authentication. All other database engines must specify a username and password that is specific to that database (i.e. not the same as their Windows login).

Test Connection

Click the **Test Connection** button to validate that the specified engine / server / username / password settings all point to a valid database engine and that the credentials are correct.



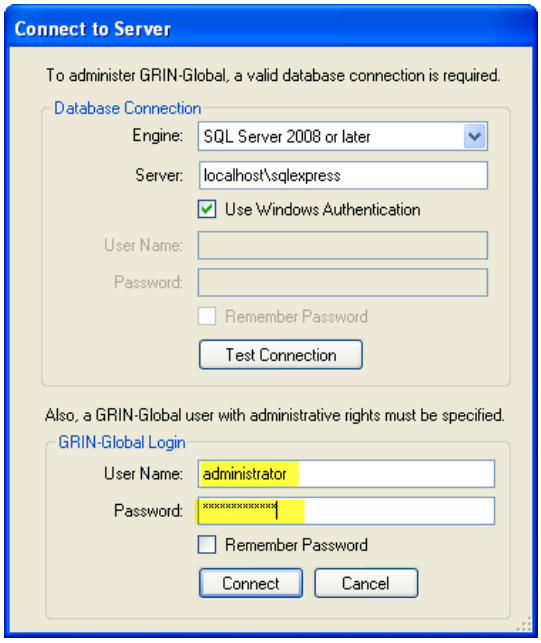
If you did not receive a “Connection Succeeded" message, then either the server name or the UserID/password combination was incorrect.

Verify that the servername and instance (or port) is correct. A SQL Server connection typically has an "instance name" with the server whereas the other three supported database engines include a port (see “Server” on page 27).

See also for detailed connection trouble-shooting suggestions.

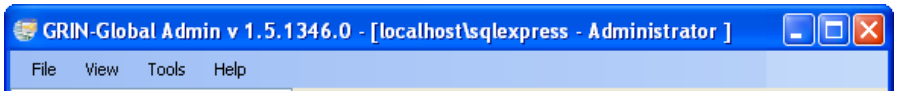
GRIN-Global Login

Use **administrator** for both the user name and password.



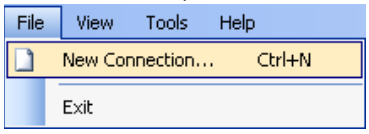
Admin Tool Menu

The Admin Tool menu items are:



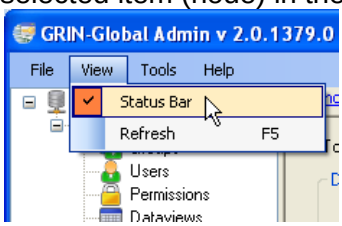
File | New Connection

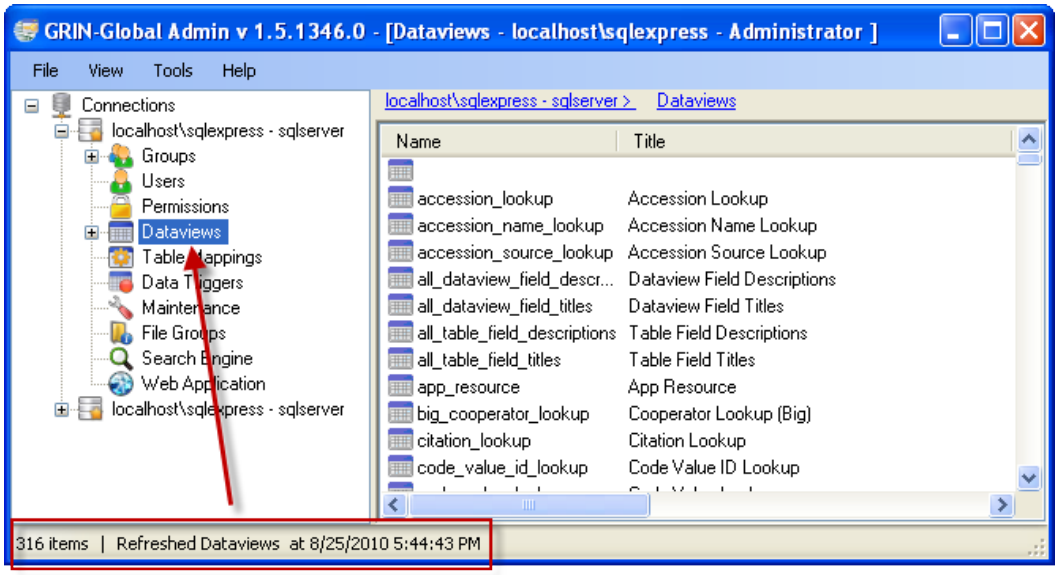
Use **New Connection** to point the Admin Tool to a different server. The Admin Tool defaults to localhost (the topmost node in the treeview on the left). If you make a new connection to a remote server, a second node will be added.



View | Status Bar

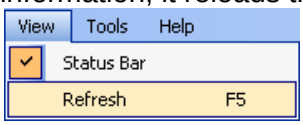
When **Status Bar** is checked, the status bar displays information pertaining to the currently selected item (node) in the treeview.



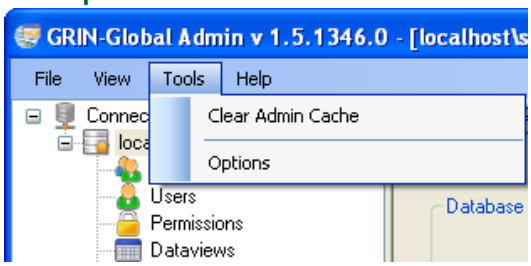


View | Refresh

Refresh (F5) redisplays your current form. For example, if you're viewing Permissions information, it reloads the permission info – if you are reviewing dataview info, it reloads that.



Tools | Clear Admin Cache

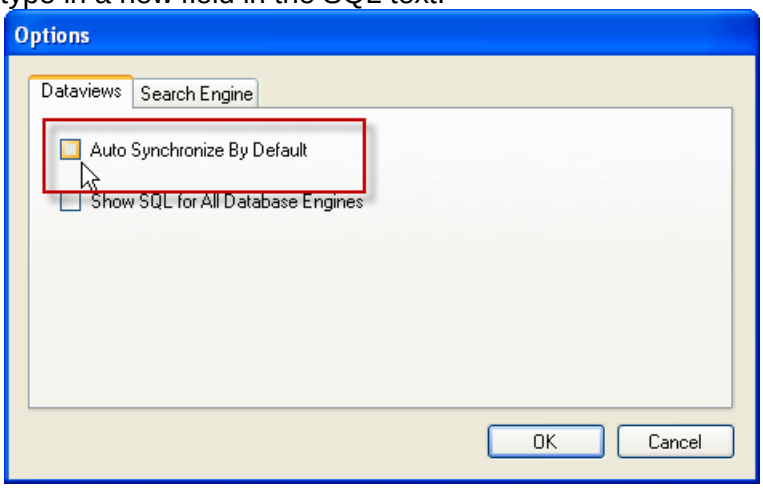


The **Clear Admin Cache** option clears the internal cache used by the Admin Tool. (See **Delete Cached Files** on page 15.)

Tools | Options

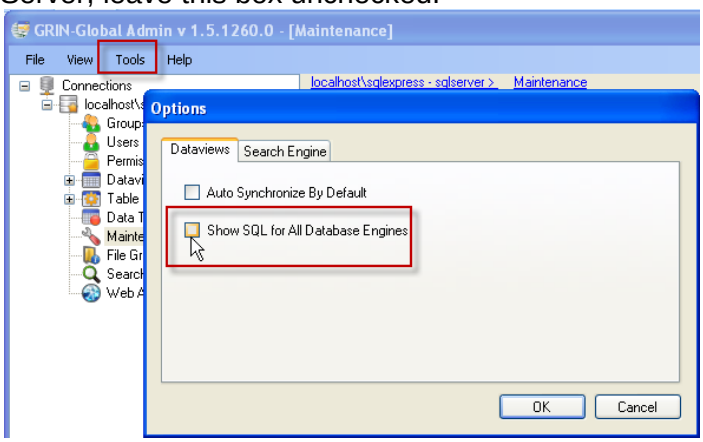
Auto Synchronize By Default Checkbox

Under **Tools | Options | Dataviews**, select the **Auto Synchronize By Default** checkbox if you want the changes in your SQL to be reflected in the gridviews automatically such as when you type in a new field in the SQL text.



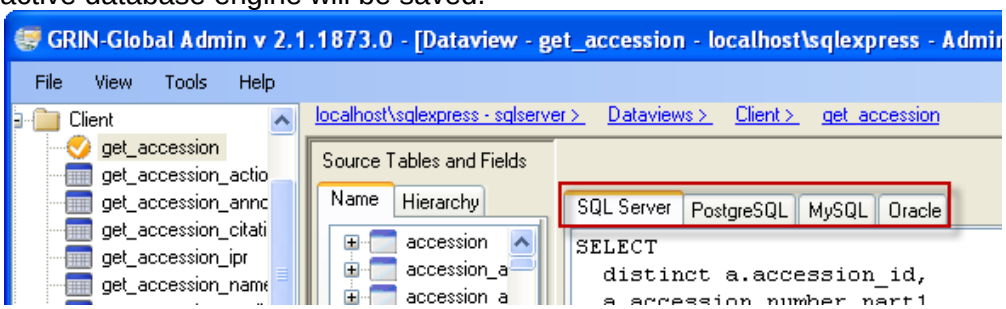
Show SQL for All Database Engines Checkbox

Under **Tools | Options | Dataviews**, select the **Show SQL for ALL Database Engines** checkbox to write SQL statements for the four database engines. If you will only be using SQL Server, leave this box unchecked.



Note

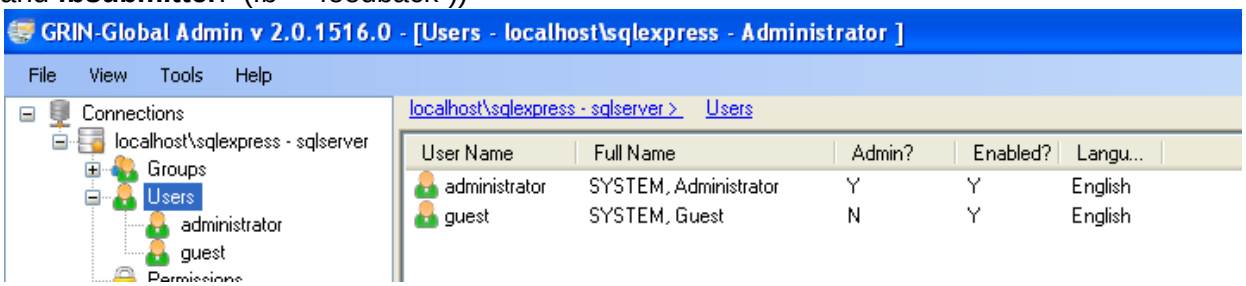
If your organization will be working with more than one database engine, then ensure that this checkbox is checked. Otherwise, if a dataview is changed, only the SQL for the active database engine will be saved.



Adding Users

By completing the various **Users** windows, you as the administrator can indicate what permissions the UserID will have and in which groups the UserID will be included. Within the User management windows, you can indicate the User's ID, password, contact information, etc.

Initially there are two users: **administrator** and **guest**. (In later AT versions, the installation procedure began including two additional users which currently are not being used: **fbowner** and **fbsubmitter**. (fb – "feedback"))



Default Users

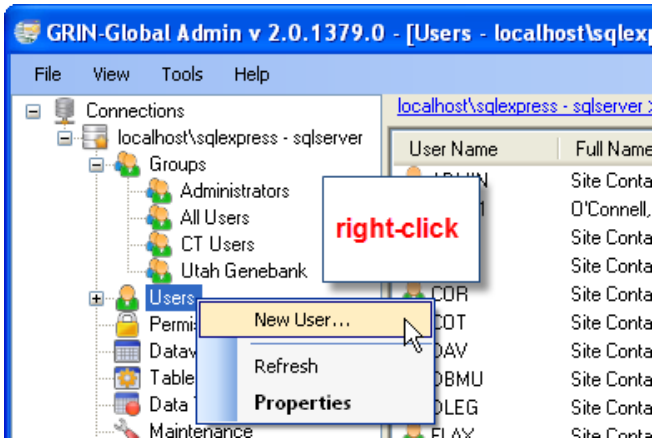
- **administrator -** full access to the GRIN-Global system. The administrator can add/edit users, etc.
- **guest -** needed for the Public Website; enables users who do not register or login to use some of the basic functions of the GRIN-Global Public Website.

  (Also, visitors can create new logins via the web site's Register functionality. To keep web-based user logins separate from actual GRIN-Global logins, an entirely separate user table exists. This means registering a user via the web site does NOT create a valid login that can be used via the CT.)

Tip ***Do not delete the default users!***

Adding a New User to GRIN-Global

To add a new user, right-click on **Users** in the left panel of the Admin window; select **New User…** from the menu:



Adding Users

At a minimum, you should establish/select the following when adding a new user:

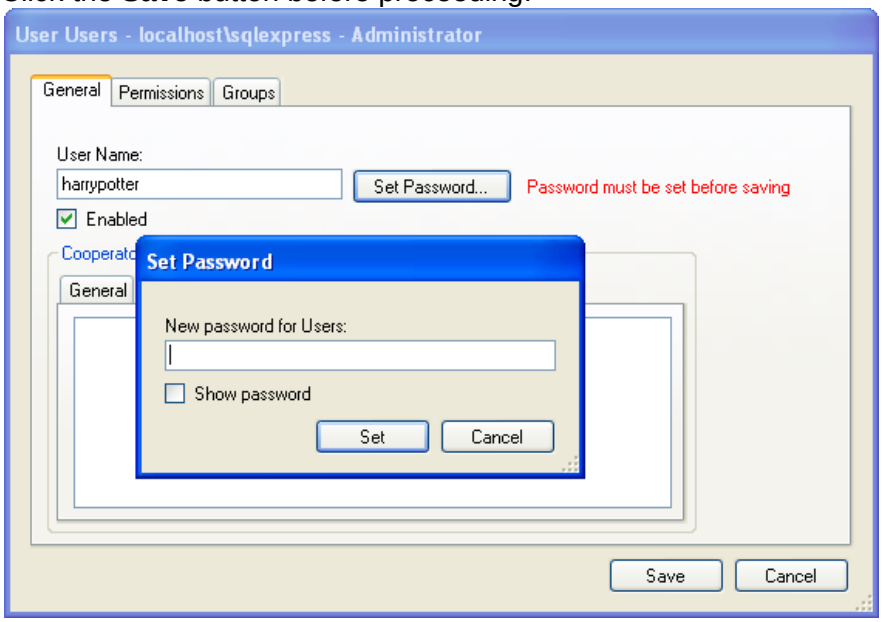
- User Name and Password
- Geography and Site codes
- Enabled (checkbox)
- Active (checkbox) (if this user will be an active cooperator)
- Groups: CT Users

General and Contact Information

Creating a User Name and Password Recommendations

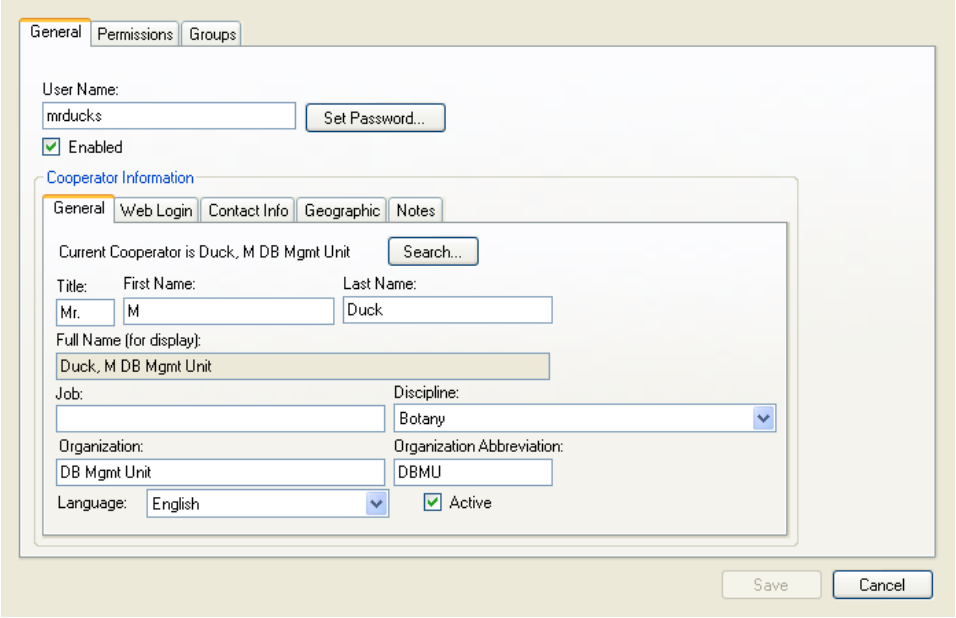
First step in creating a new user is to assign a **User Name** and **Password**. We recommend not using any whitespace or special characters in the User Name. Each organization has enforces different password rules – setting password parameters is explained in the Web Application section.

Input a **User Name**; click the **Set Password** button.
Click the **Save** button before proceeding.



Complete the fields displayed on the **General** and **Contacts** tab with the user's relevant information. (**Web Login** will be discussed later.) Use the **Search** button if there is the

likelihood that the new user is already in the system as an existing cooperator.



Enabled

When selected, the “Enabled” checkbox indicates that the user will be allowed to login to the Curator Tool.

Active Users

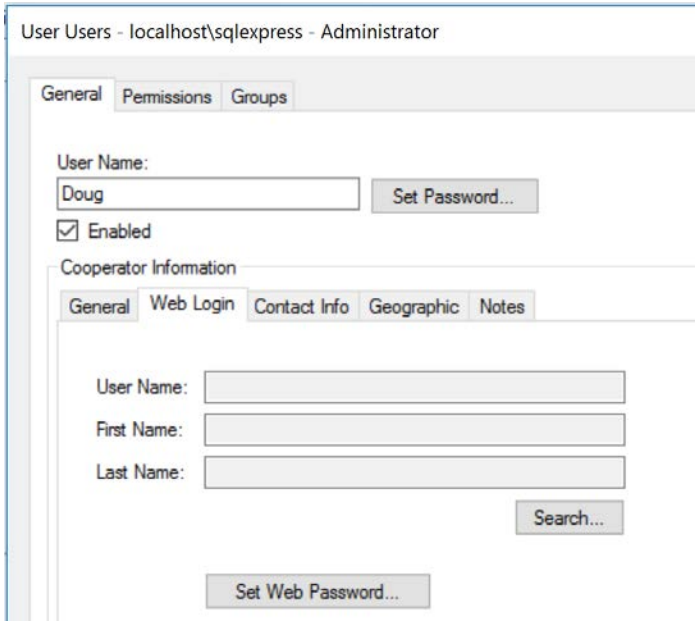
The **Active** checkbox indicates the UserID is associated with an active cooperator, meaning it is not a historical one – any data created or modified by this user will be tagged by his **CooperatorID**.

Web Login Tab

At the **Web Login** tab, you can perform two main tasks: (1) you can associate a Public Website UserID with a Curator Tool UserID and (2) reset a password for a Public Website user.

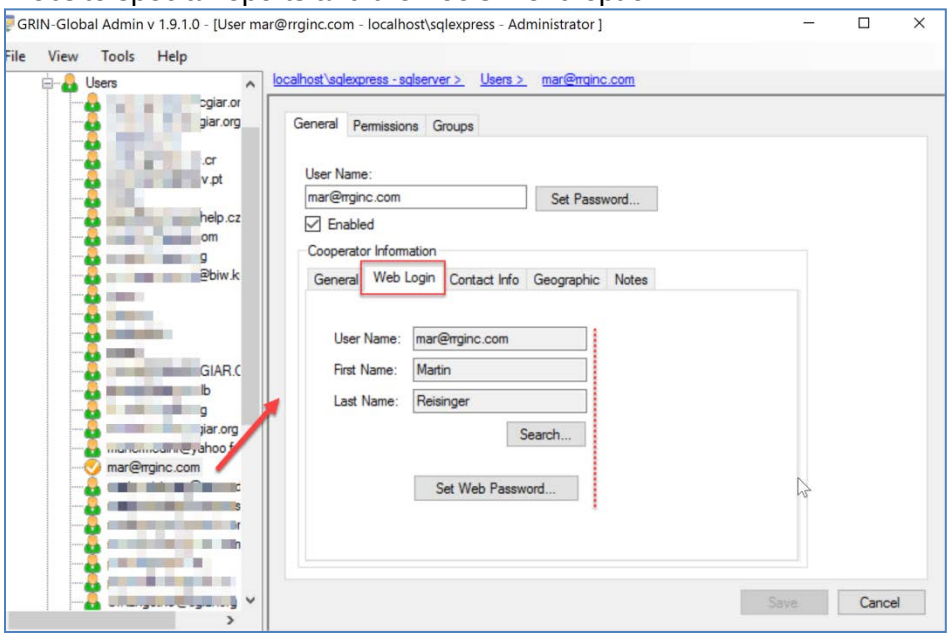
Initially, the Web Login information is grayed out:

Adding Users



Assigning a Web Login for Internal (Genebank) Users

The Public Website was designed for users who need to search for accessions and perhaps order them, typically external general users, researchers, breeders, etc. Genebank staff will also use the Public Website to search for accessions and display descriptors information, taxonomy, etc. Over time the Public Website has been modified to include additional features which are only appropriate to internal users, that is, users working in the genebank. When a CT user's User Name is configured with a Web Login, that user can then access on the Public Website special reports and the Tools menu option.

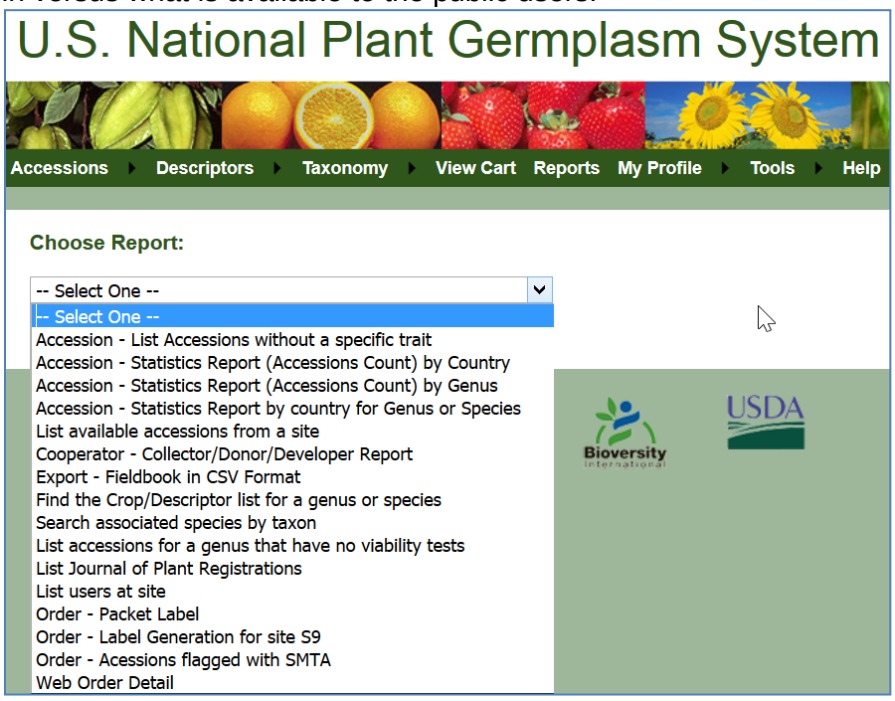


The GG Administrator can complete this screen after the user has created his Public Website account, or can create it when creating or modifying the Curator Tool account.

Before the CT user has logged in:

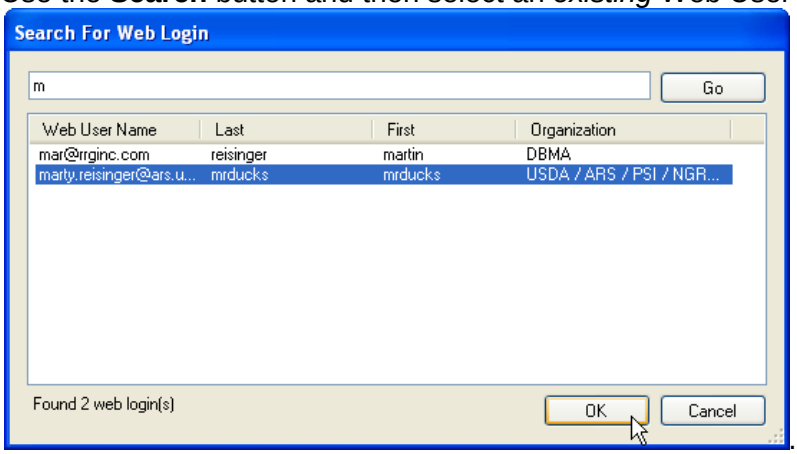
After she has logged in:

At the National Plant Germplasm System, these are the reports currently available when logged in versus what is available to the public users:



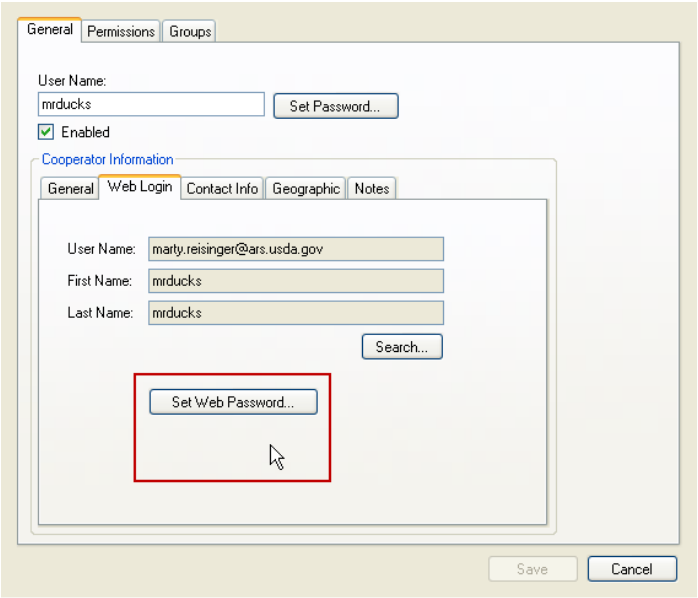
To associate a Web Login with a CT User account

Use the **Search** button and then select an *existing* Web User from the list of possible matches:

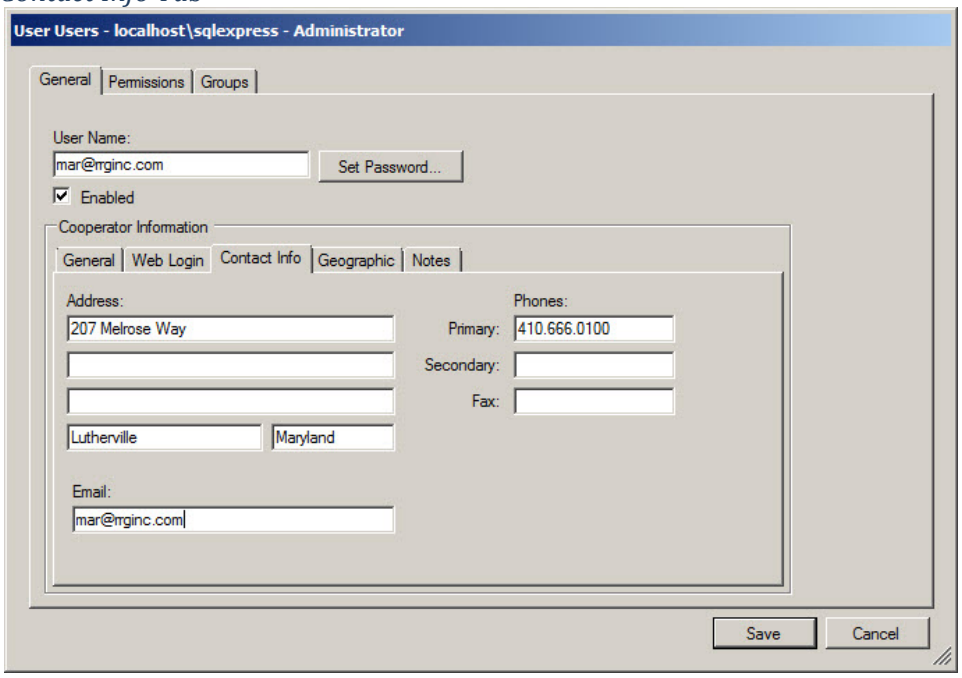


To Reset a Public Website User Password

First select the desired **UserName** if not already selected, and then click the **Set Web Password**… button:



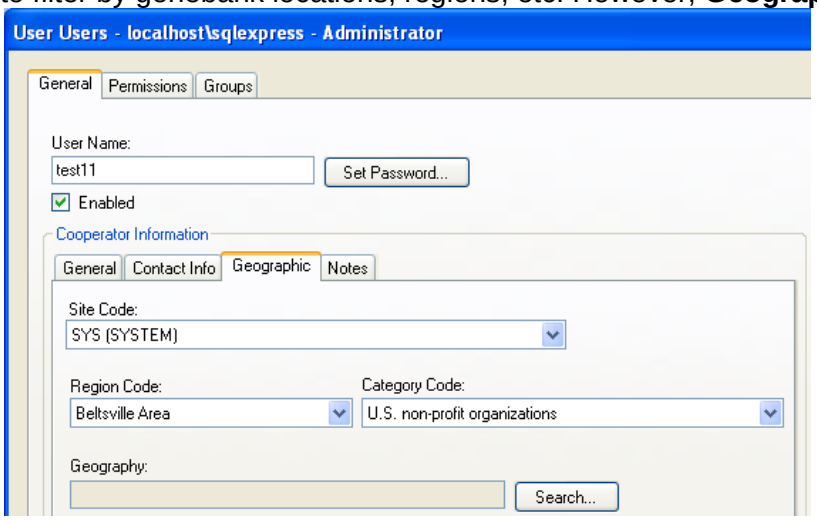
Contact Info Tab



Geographic Tab

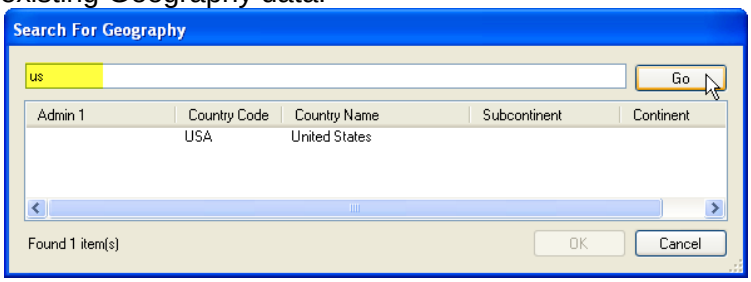
The **Geographic** tab has several text boxes available for classifying sites by **Site Code**, **Region**, **Category Code**, or **Geography**. Some organizations, especially smaller ones, may not have a need to use these classifications, whereas others may find that these can later be used

to filter by genebank locations, regions, etc. However, **Geography** is a required field.



GRIN-Global is installed with only one Geography record. The exception to this occurs when during the database installation via Updater, the GG administrator invokes the (Shift + Ctrl) option to install the GRIN Geography table.

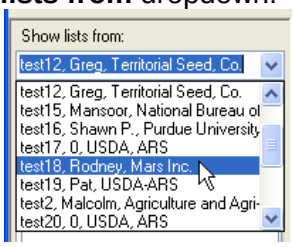
Click the **Search** button; on the **Search For Geography** window, input text to search for existing Geography data.



Site Code

Site Codes are optional but are frequently used, especially in larger institutes with multiple physical locations. However, site codes can be used to separate users into teams or virtual groups. See the guide

Users with the same **Site Code** can share lists within the Curator Tool. (They select another user's lists via **Show lists from** dropdown.) Also, when necessary, a Curator Tool user can copy a list from one account to another, simply by selecting a different account from the **Show lists from** dropdown:



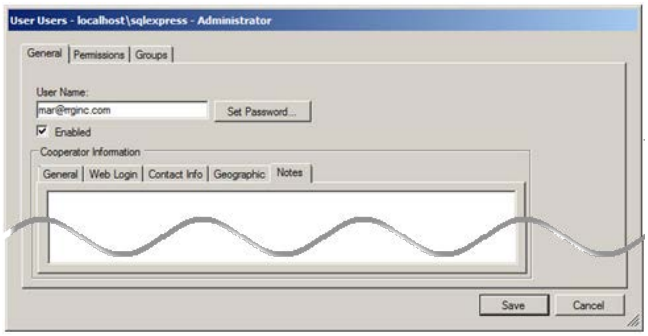
Initially, when GRIN-Global is installed, only one **Site Code** is installed: **SYS (System)**. Site records can be added at any time after the initial installation via either the Admin Tool's Import Wizard or the Curator Tool.

Tip

If your organization allows individuals to have multiple user accounts, a user with multiple accounts could set up his or her accounts to have certain defaults. For example, one account could be set up to display Order-related dataviews only. The same user could use that account for ordering, and another account for organizing accessions.

Notes

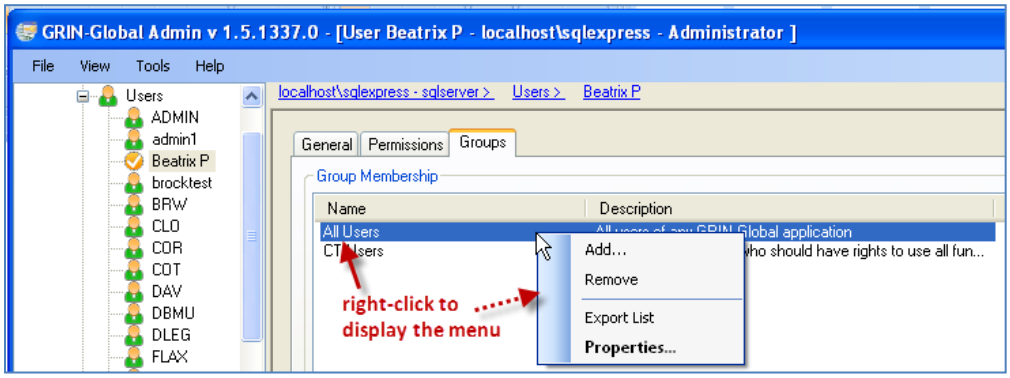
Use the **User | General | Notes** tab to input any user notes.



Adding Users to Groups

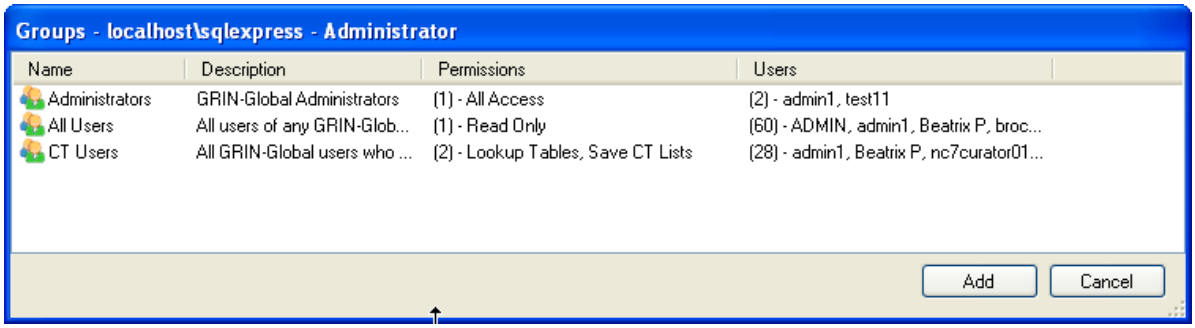
Users may be added to a group at any time; they also may be assigned to a group when the **User Name** is being created. (See the ***Groups*** section for more details.)

To assign the user to a group, click on the **User Name**; click on the **Groups** tab; right-click in the list; then select **Add…**.



Select the groups to which the **User Name** will be added (use Ctrl- or Shift- to select multiple groups); click the **Add** button.

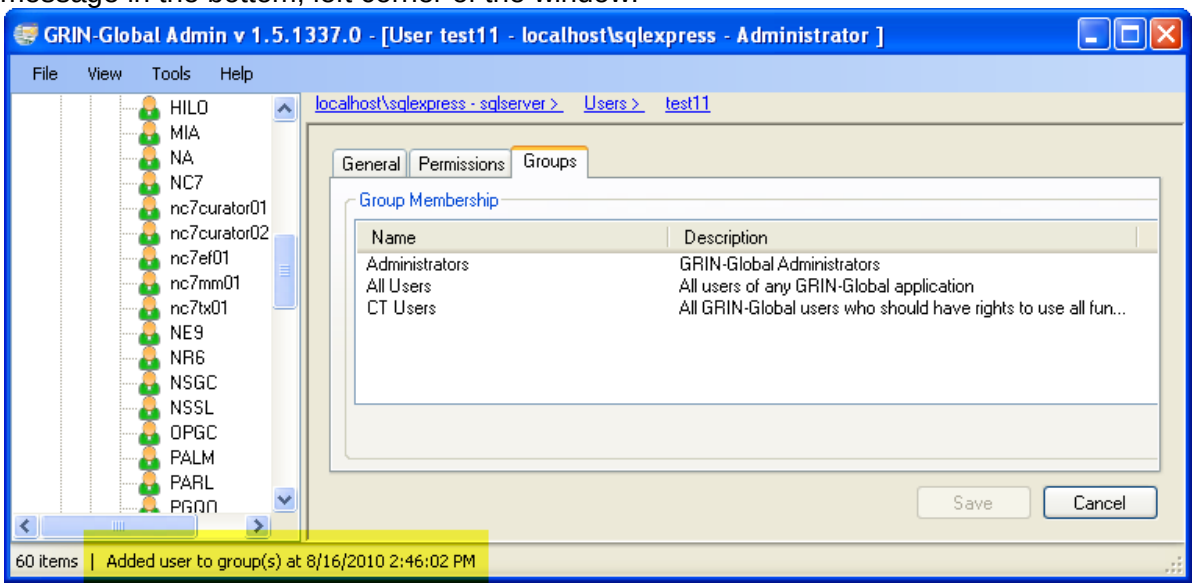
Groups



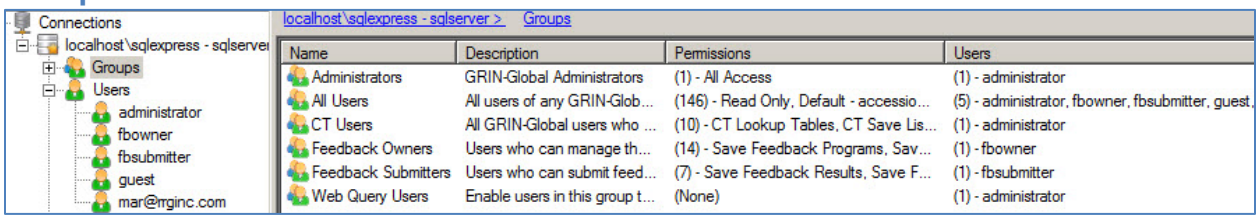
Note

The web application (Public Website) allows a user to be anonymous. That account is the "**guest**" account. It should never be a member of the **CT Users** or **Administrators** groups.

When a user has been successfully added to a group, the program will display a relevant message in the bottom, left corner of the window.



Groups

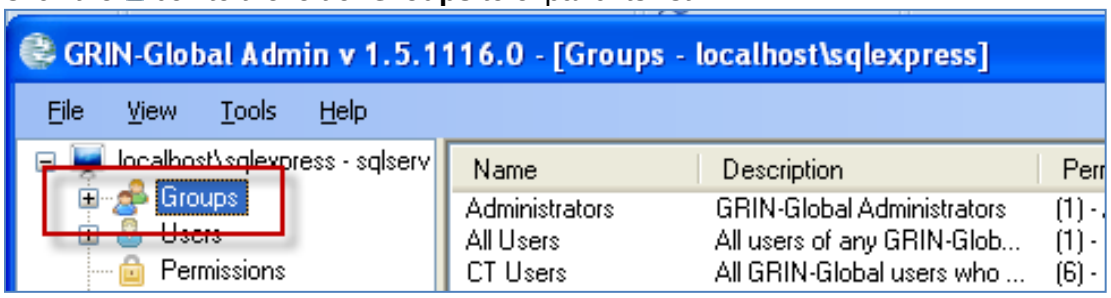


With the Admin tool's **Groups** feature, you establish (or edit) user groups who will have similar security privileges ("assigned permissions"). At installation, six default groups are created:

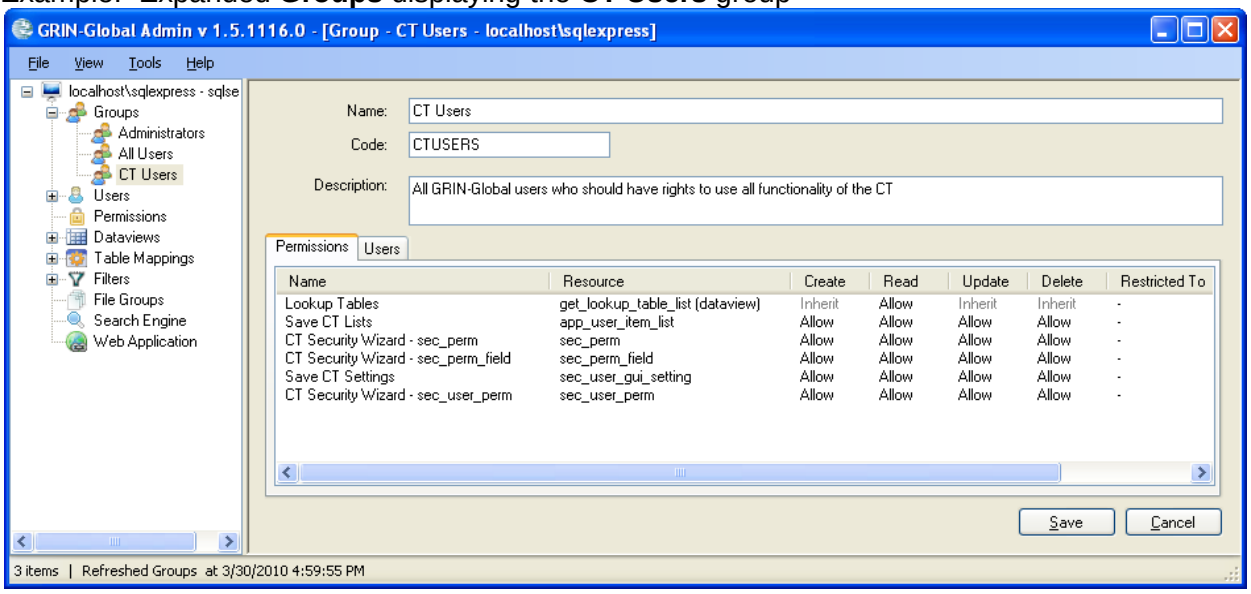
1. **Administrators** – users with full, unrestricted access to all GRIN-Global resources
2. **All Users** – every GRIN-Global user will be a member of this group

3. **CT Users** – all GRIN-Global users who will have access to run the Curator Tool
4. **Feedback Owners** – not functional yet
5. **Feedback Submitters** – not functional yet
6. **Web Query Users** – users who are given access to the Web Query feature located under the Tools menu option on the Public Website

Click the ⊞ box to the left of **Groups** to expand its list.

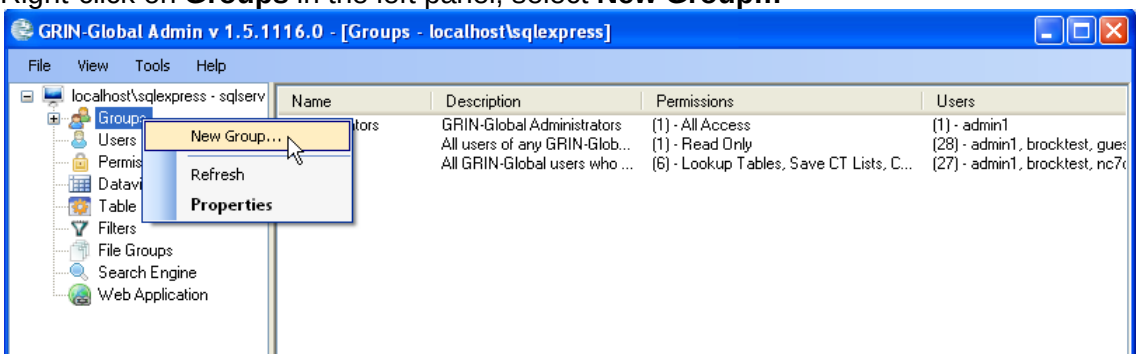


Example: Expanded **Groups** displaying the **CT Users** group



Creating a New Group

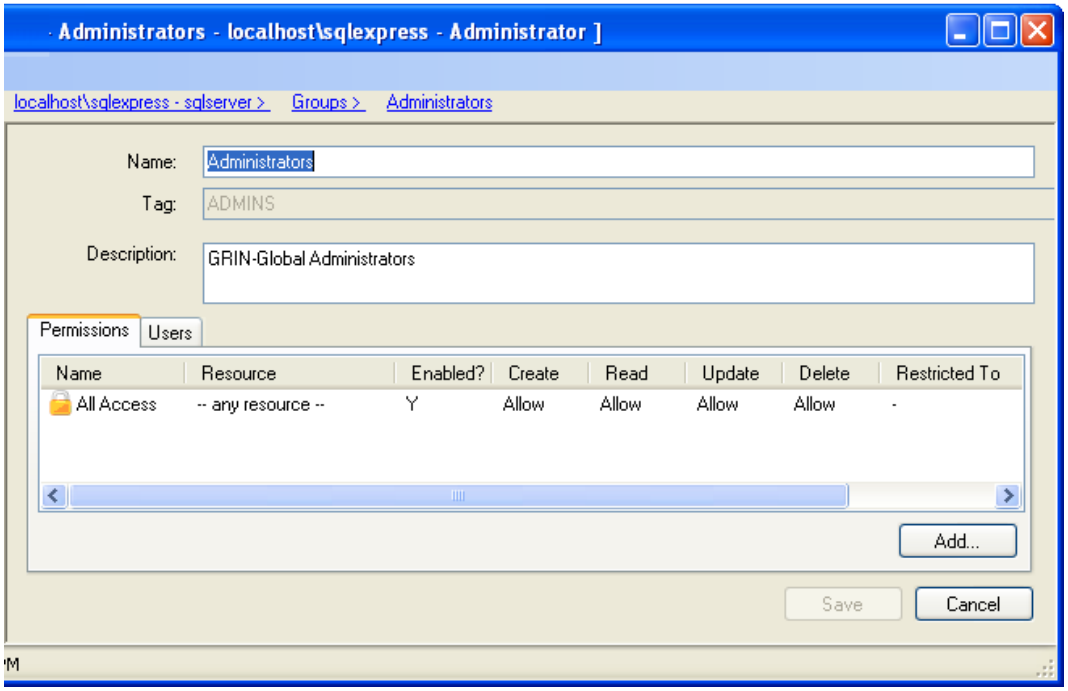
Right-click on **Groups** in the left panel; select **New Group…**



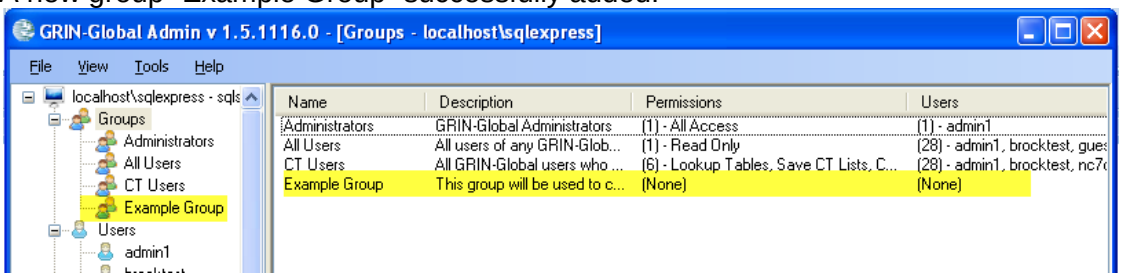
Tip

Virtually every tree or list in the Admin Tool has a context menu associated with it. To perform primary functions, get more information, etc., right-click on any item in the left panel to review its context menu and available options.

The **Group – Groups** window displays. Input a meaningful **Name**, **Tag**, and **Description** for the group. In general, any **Tag** property on any form in the Admin Tool is simply a value that may be programmatically used by code. The **Tag** value here is used internally by the middle tier to grab "ADMINS" group to see if a user has administrative rights. (This is something hardcoded in the middle tier to grab the list of administrators.) The CTUSERS and ALLUSERS default groups have those tags. (ALLUSERS is used by the Admin Tool when creating a user –every user is added to ALLUSERS group). The maximum length for the **CODE** is 10 characters. Click **Save**.



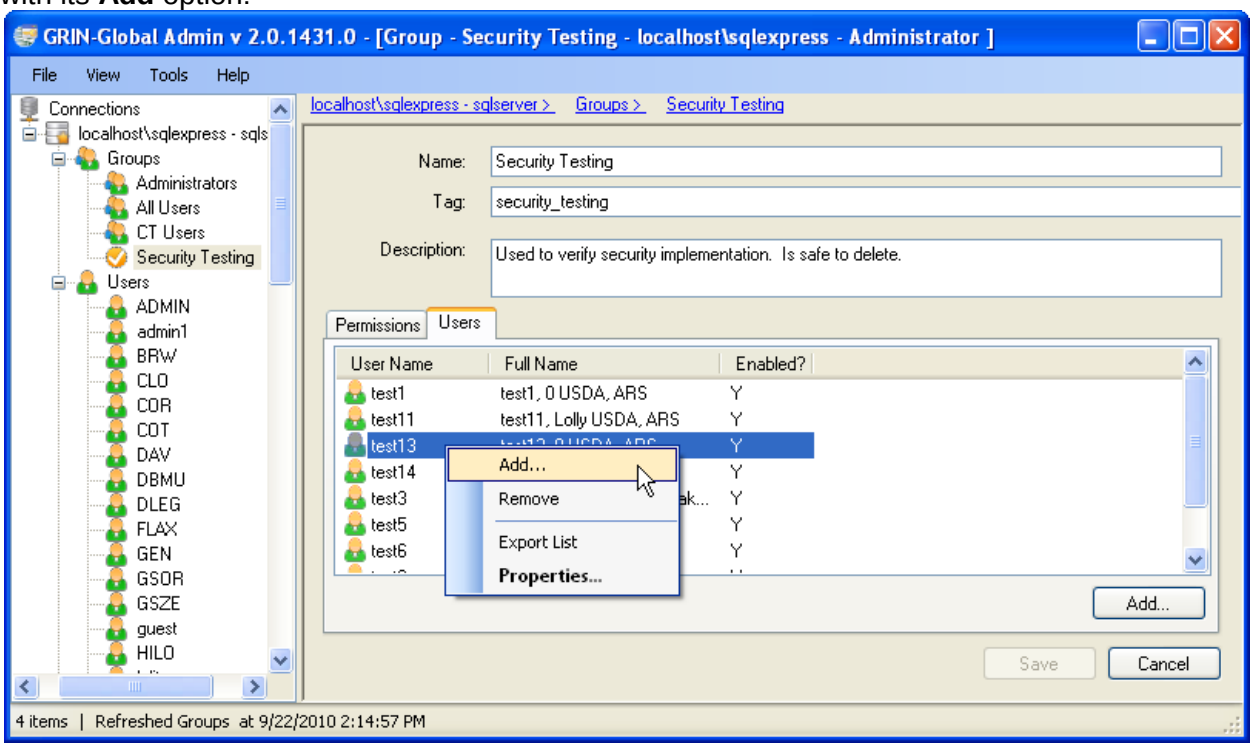
A new group “Example Group” successfully added:



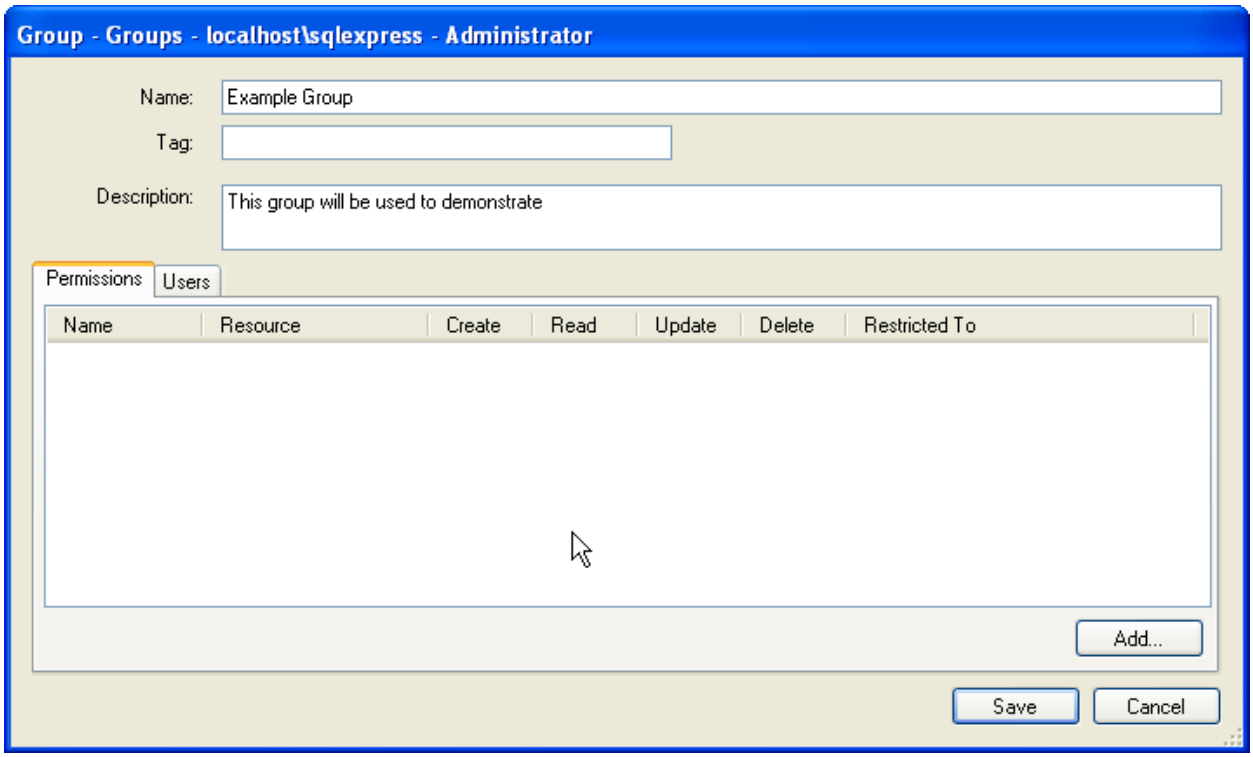
Note

Individual UserIDs are assigned to a Group either when the member is added as a new **User** or anytime after the Group name has been saved. (See **User | Groups** for details.)

The following screen example illustrates adding users to a new group. The administrator selected the **Example Group** in the left panel and then right-clicked to display the context menu with its **Add** option:



Groups


Group - Groups - localhost\sqlexpress - Administrator
Name:
Example Group
Tag:
Description:
This group will be used to demonstrate
Permissions
Users
Name
Resource
Create
Read
Update
Delete
Restricted To
Add...
Save
Cancel

Security

Tip

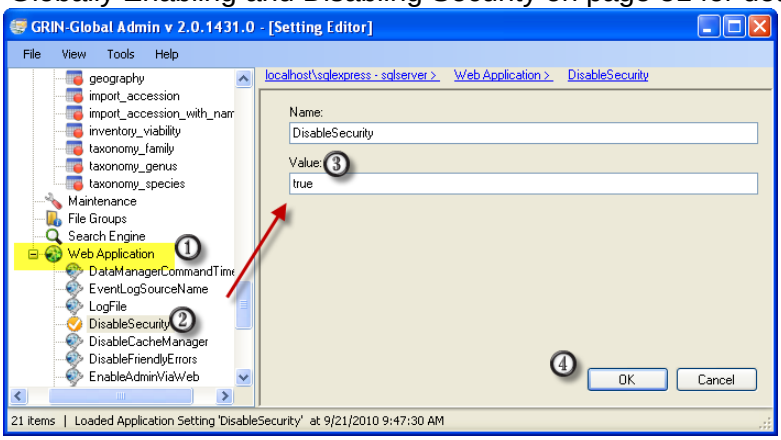
Typically, for records they own, Curator Tool users can establish permissions and assign ownership rights to other users. In many cases, these security parameters in the Curator Tool are adequate for an organization's security needs.

The security code written in the middle tier has never been fully exploited (or tested) and it is suggested that the Curator Tool be primarily used for implementing permissions at the record level.

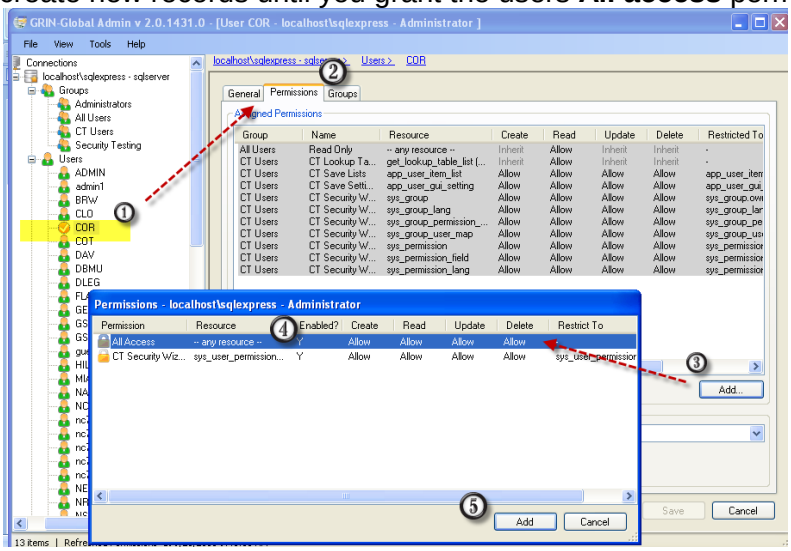
After reading this security section, GRIN-Global administrators should read the text in the separate document Adding Users [http://www.ars-grin.gov/npgs/gringlobal/docs/gg_adding_users.docx]. Ownership and the permission settings at the record level are handled in the Curator Tool and are fully explained in the Curator Tool Users Guide, under "Security."

Administrator Security Settings

1. Security is enabled by default. If you prefer to globally disable all security restrictions, in the Admin Tool, select **Web Application**; double-click on **DisableSecurity**; select **True**. (See "Globally Enabling and Disabling Security on page 51 for details.)



Alternatively, if you keep security enabled (recommended), then individual logins cannot create new records until you grant the users **All access** permission.



Security Overview

Typically, the administrator will assign each user to the **All Users** and the **CT Users** groups.

The administrator could also create less inclusive groups to handle permissions to handle unique situations. Besides modifying permissions by *group*, an administrator can modify permissions by *user*. However, as mentioned previously, in the typical GG environment, users decide who will be able to update or delete records.

When discussing security, two broad concepts must be understood:

- permissions
- ownership

Ownership

The concept of ownership is fully explained in the Curator Tool Users Guide, under "Security." Briefly, a person who created a record is typically the owner. However, this is not always the case.

In some cases, database rules have been established so that the owner is not the creator of the record. For example, a person who creates an inventory record is not necessarily the owner of the record. When it is created, an inventory record must have its required fields filled in, including the **Inventory Maintenance Policy** field. The owner of the **Inventory Maintenance Policy** record selected for the inventory record becomes the owner of the inventory record. This can be disconcerting to a user who may have just created the inventory record and made a typing error during the process – he may not be able to edit that record unless the owner of the **Inventory Maintenance Policy** is contacted to either change the ownership or grant the user permission to edit the new inventory record.

Permissions

For each user, permissions are applied to:

- a specific table
- a specific dataview

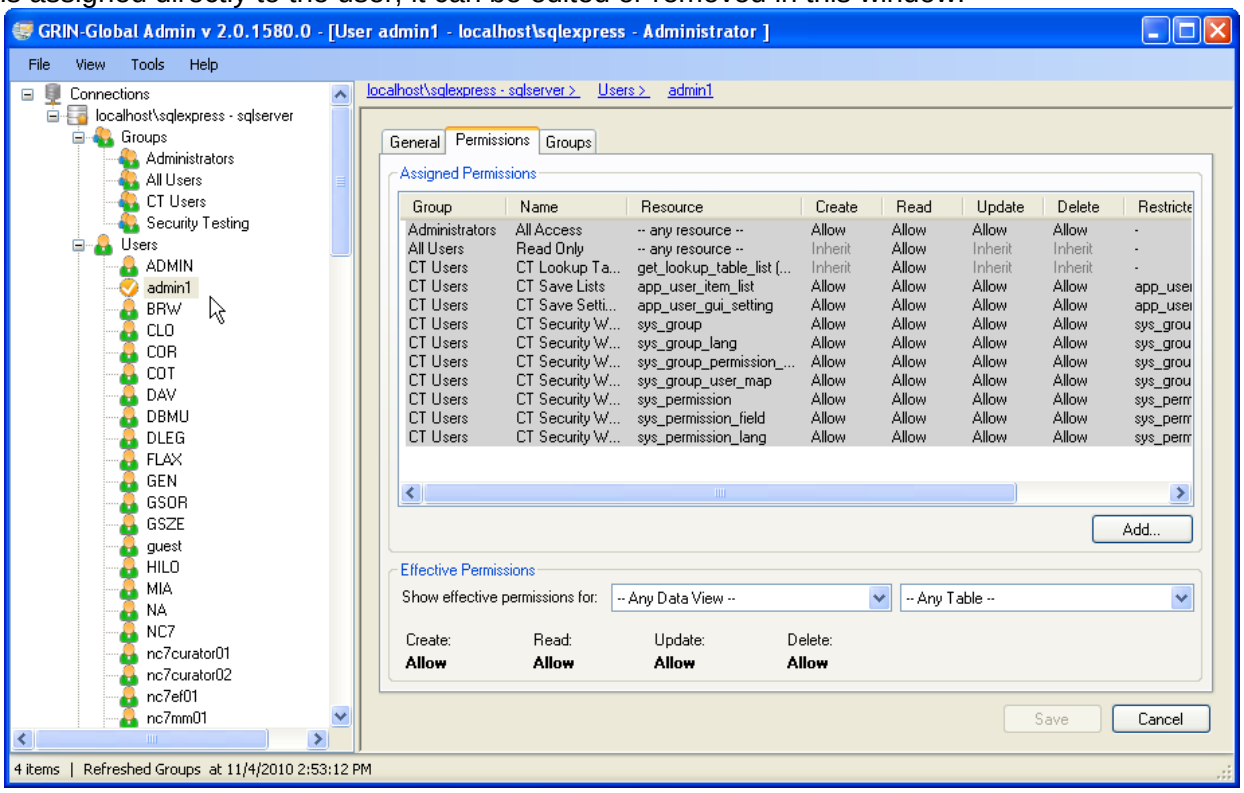
The **All Access** permission applied database-wide gives Create/Read/Update/Delete rights. The **Administrators** group has this permission. (One of the default accounts, **administrator**, is a member of this group.)

In the example below, **admin1** has been selected. Because **admin1** belongs to the **Administrators** group, it has permission to do everything.

Admin1 is also a member of the **All Users** and **CT Users** groups. Since those permissions are also used when calculating Admin1's rights to a resource, they are displayed on this form.

Grayed-out rows indicate permissions assigned to a group in which the user belongs. Therefore, in this window, the permission cannot be edited or removed. White rows indicate the permission

is assigned directly to the user; it can be edited or removed in this window.



Permission Defined

A permission restricts or grants access to a resource in GRIN-Global. A resource is typically defined as a specific row (record). A permission defines four kinds of rights:

A permission of type:	Has the ability to:
Create	Insert new data
Read	Read existing data
Update	Update existing data
Delete	Delete existing data

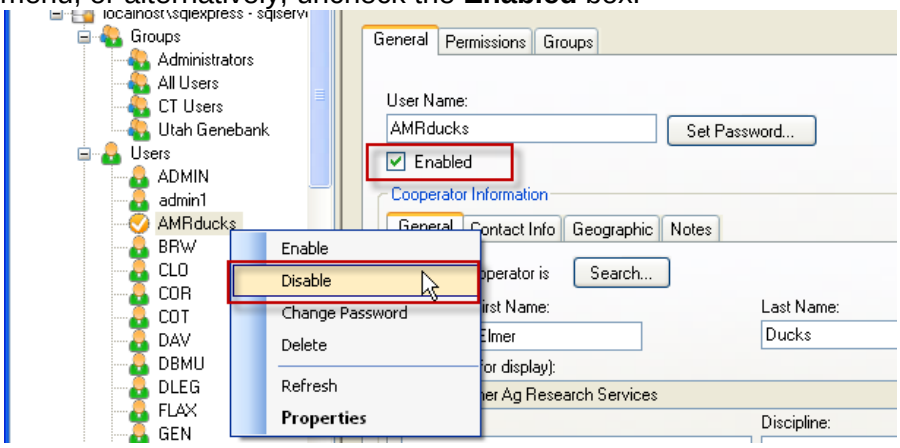
Each right can have one of three values:

Value	Description
Allow	Allows access
Deny	Denies access
Inherit	Neither allows nor denies access; access is situational; it is inherited from a previous definition

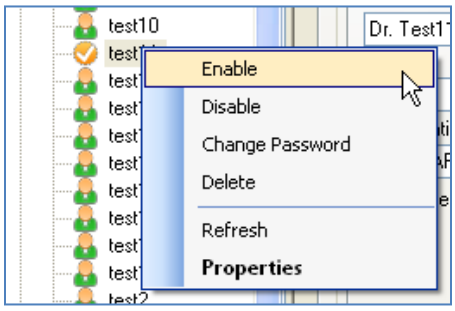
Disabling Permissions

Permissions can be complex. Sometimes it is simpler to disable permissions and then test security, instead of deleting the user(s) and then recreating them again.

To disable a user's permission settings, select the user, right-click, and select **Disable** from the menu, or alternatively, uncheck the **Enabled** box.

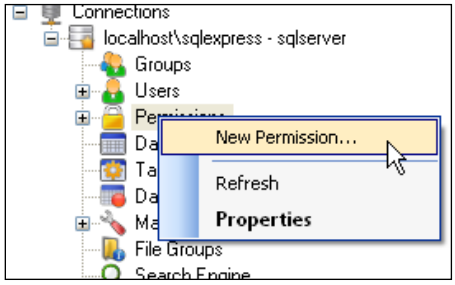


Conversely, when permissions have been disabled, select the username and then either select the **Enabled** checkbox on the **Permissions** window, or right-click and select **Enable** from the context menu.

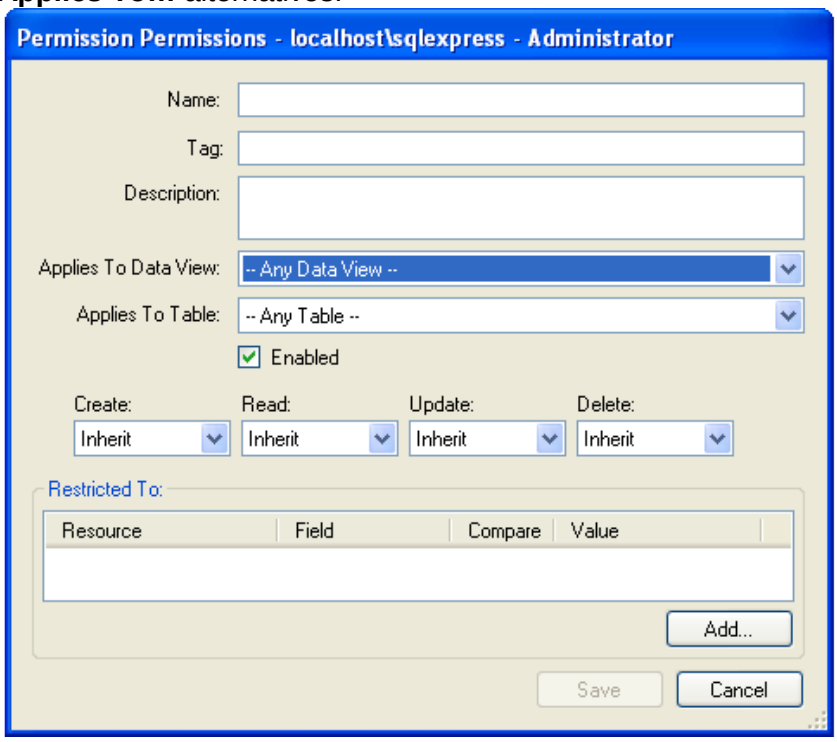


Setting Up a New Permission

To create a new Permission, right-click on the **Permissions** node; select **New Permission…**



Complete the **Name** field (**Tag** and **Description** fields are optional); select from one of the **Applies To…** alternatives.

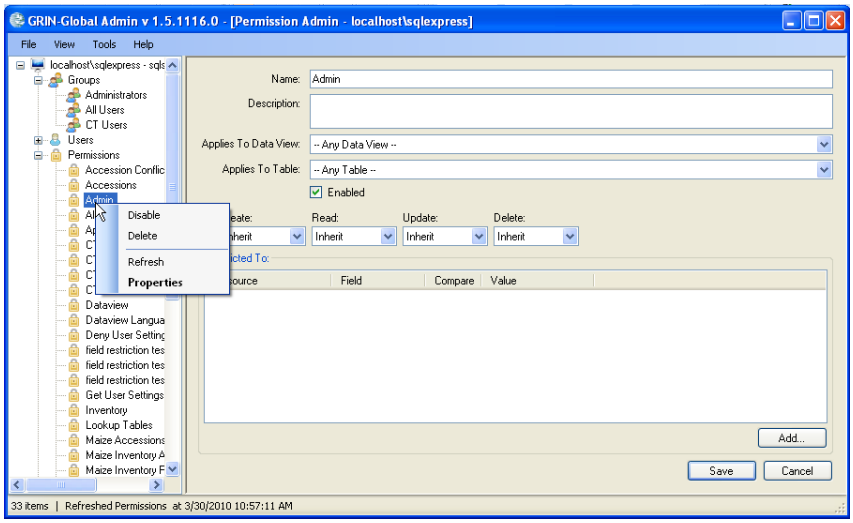


"Applies to Dataview" vs. "Applies to Table"

These two alternatives provide two basic levels of controlling access. Usually the "applies to table" will be used, but "applies to dataview" is useful in certain situations.

"Applies to dataview" denotes the permission affecting a dataview–if the user has an ALLOW permission to read the cooperator *table*, but a DENY permission on the get_cooperator *dataview*, he won't obtain any data when he tries to retrieve data via get_cooperator.

"Applies to table" affects the permission to the table regardless of what dataview is trying to access it.

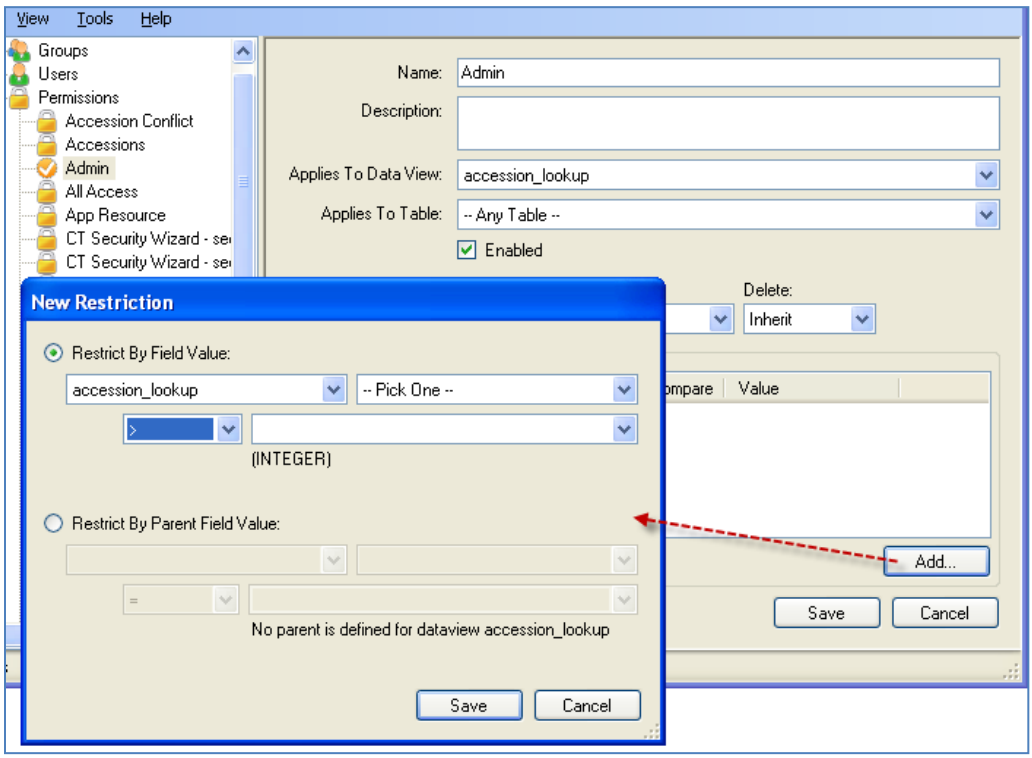


Setting Restrictions at the Row Level (Add ...New Restriction)

Note

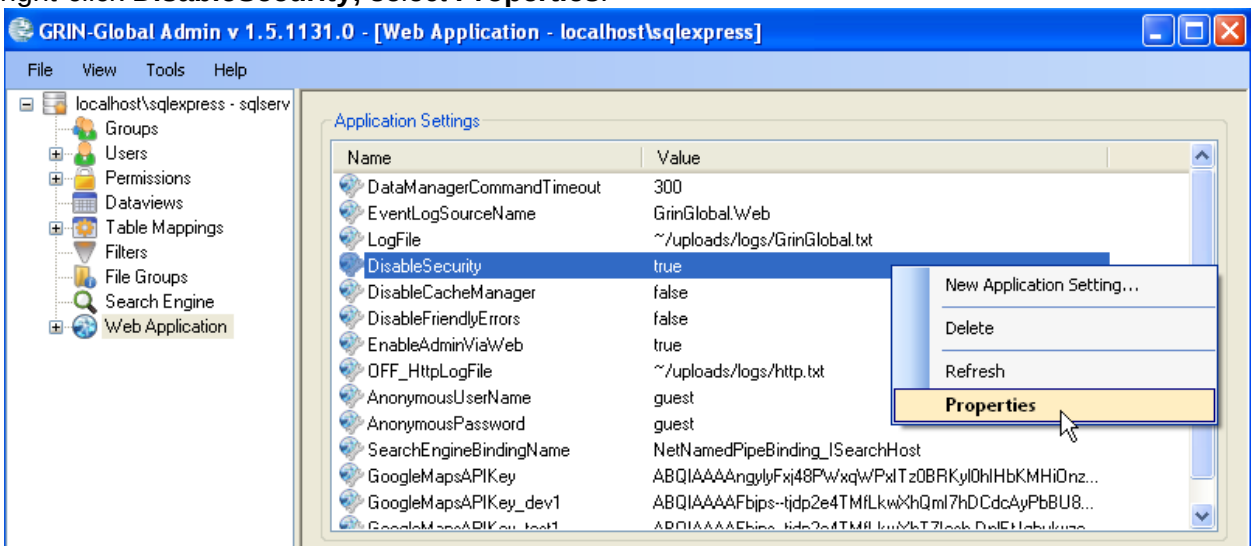
The following text is for reference; in reality administrators can typically avoid setting up restrictions, but rather let the users configure their own permission rules.

Permissions can be assigned not only at the table level, but also at the row level. For instance, student A should only be able to update accessions owned by Curator 1. This would be done by adding a restriction to the permission (i.e., you're restricting the permission to just rows that match the criterion specified) saying that accession.owned_by = <cooperator id for Curator 1>)

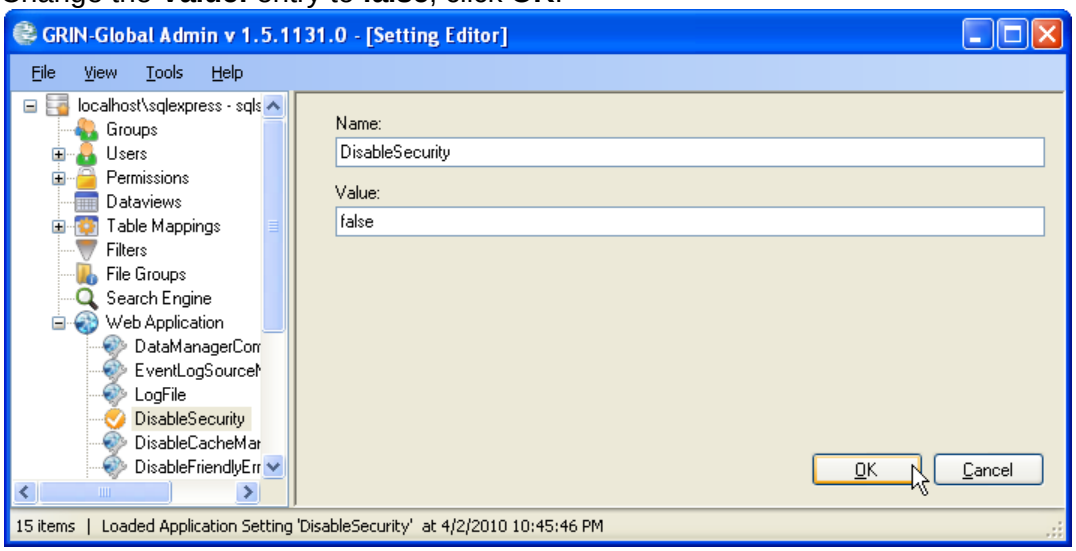


Globally Enabling and Disabling Security

Only the administrators of the installation can enable security. On the web application node, right-click **DisableSecurity;** select **Properties**.



Change the **Value:** entry to **false**; click **OK**.



Same Computer Requirement for Certain Administrative Rights

For editing anything in the Data Triggers, Search Engine, or Web Application nodes, the Admin Tool must be run on the same computer where the web application / search engine are installed. (Some of the properties that you are changing require administrative rights to do so, and that requires a local login.) If you try to edit any of those and you're not on the same physical box, a message box will display indicating that you must be on the same computer and prevent you from changing (or possibly even viewing) anything.

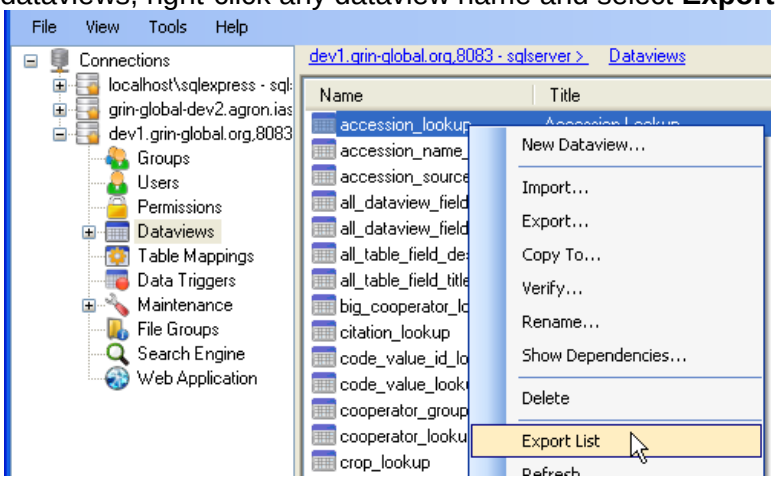
To edit any of the other nodes, only a database connection is needed.

Dataviews

Dataviews: Overview
GRIN-Global dataviews are a means for dynamically adding new queries to the Curator Tool. A GRIN-Global Dataview is essentially a stored SQL Query with some peripheral information. A dataview consists of the following components:

Component	Description
SQL Query Definitions	A SQL statement for each of the supported database engines -- SQL Server, MySQL, PostgreSQL, and Oracle (this statement was true for version 1.0.7, but is not currently valid for later GG releases.)
SQL Parameter Definitions	For passing user specified values to the SQL queries at run time
Field Mappings	For associating a dataview field with a specific table and column in the database, so that data in that field can be updated by the user
Language Information	Language-specific Title and Description for column headings displayed in the Curator Tool, Search Tool, and Import Wizard
Flags and Settings	Denotes if the dataview is read only, user visible, etc.

Using the Admin Tool's dataview editor, you can list and modify any dataview available in the system as well as create new dataviews and delete existing ones. To produce a list of all dataviews, right-click any dataview name and select **Export List**.



Note: The Dataview editor is used for editing *dataviews*, not *table schema* properties. Use the database's associated database management tool, such as SQL Server Management Studio, to make changes to the actual schema.

Tip: Occasionally you will need to edit a dataview. Before making changes to any existing dataview, it is recommended that you first export the dataview that will be edited. Keep this as a backup in case later it needs to be restored. Consider including the date as part of its filename when you export it.

Dataview Fields

Dataview fields are mapped to provide additional information, such as language-specific column headers, read-only or editable status, foreign key lookups, etc.

Fields are updatable via a dataview only when the following is true:

1. They are mapped to a specific table and field
2. The Table Mappings for that table and field are not marked as read-only
3. The Primary Key for that table is included in the dataview
4. The field is not marked as read-only in the dataview

Dataview Example: get_accession_name

The GRIN-Global application was designed with many dataviews which meet most organizations' needs. Of course, each organization is unique and will most likely modify existing dataviews or design new ones.

The simplest way to become familiar with dataviews is to review an existing dataview. The following screen illustrates a typical dataview within the Admin Tool's dataview editor.

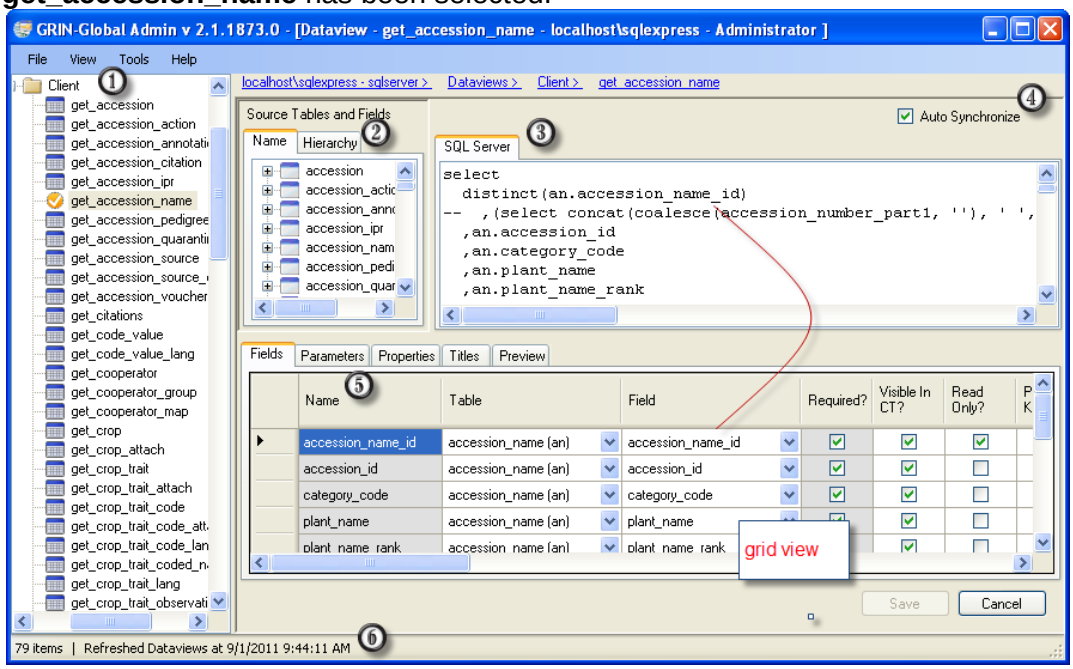
Note

The **Accession Name** dataview is being used as the example. The example is still relevant to this discussion, but the actual dataview name has been changed due to the schema changes after version 1.5. It was **get_accession_name**; the name has been changed to **get_accession_inv_name.**

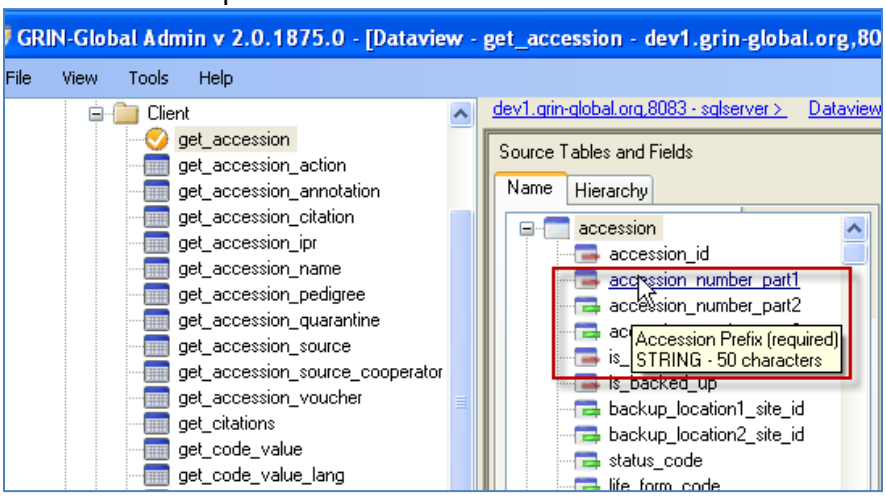
Note

Refer to Tools | Options the section *Tools* | *Options* on page 30 for information when your organization needs to edit SQL for more than one database engine. This condition is relatively rare; for most organizations this will never be the norm as most organizations will use just one database.

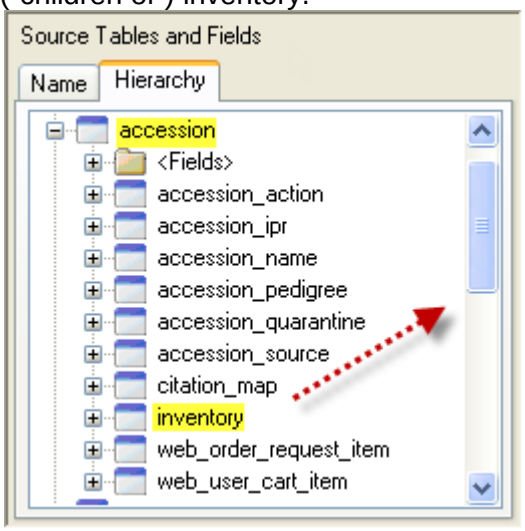
1. The left panel "tree view" lists the dataview names; in this example **get_accession_name** has been selected.

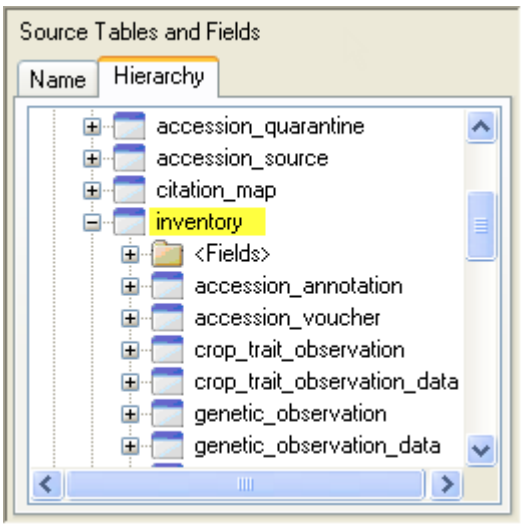


2. **Source Tables and Fields Panel**
   This panel lists all of the tables and fields in the application. (Note that it is not directly related to the dataview currently being displayed, but is here to assist an administrator in building new SQL. More on using this later – many administrators may never use this panel.) The **Name** tab lists all dataviews in alphabetical order; when scrolling over a fieldname, the friendly name is displayed as well as other field attributes; also, the red icons indicate required fields.



   The **Hierarchy** tab shows the parent child relationships between tables. The **Hierarchy** panel shows tables in their relationship hierarchy. If you expand accession, you'll see **accession_action** displayed and other accession "child" tables including **inventory** under it. If you expand **inventory**, additional tables are listed that are dependent ("children of") inventory.





3. The **SQL Server** Panel
   The **SQL** pane displays the SQL code that runs the selected dataview. Many administrators will not need to edit this SQL code. However, when an administrator needs to make additional edits to the SQL, such as adding a "WHERE" clause and possibly one or more parameters, she can edit the code in this panel or drag and drop fields or tables from the **Source Tables and Fields** Panel (see #2).

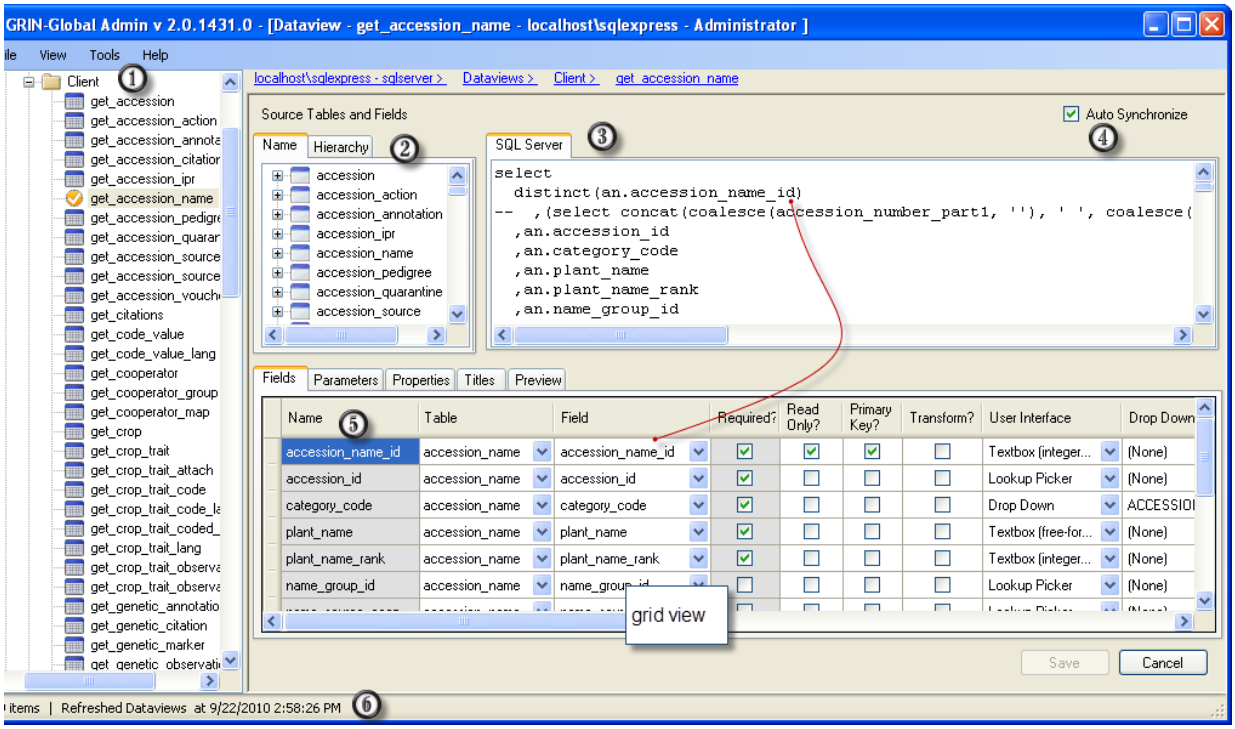
If you completely replace all the SQL in a dataview, the dataview loses all field information on a **Save** action. This was discovered when editing SQL outside the AT in order to test it using SSMS. One work-around is to *never completely* replace the SQL in a dataview. It is fine to replace the first half and then separately replace the second half, as long as you do use a single copy and paste operation.

When the administrator is satisfied with her SQL edits, in the bottom panel (#5) she would use the **Preview** tab to run the SQL against the database to ensure the dataview returns the desired information. If satisfied, the administrator saves the SQL. After it is saved, the dataview is available to the Curator Tool.

4. **Auto Synchronize** Checkbox
   This checkbox will automatically inspect the **SQL Server** pane and the grid view contents as edits are made and keeps both synchronized with each other. (To see this feature, edit the SQL directly and it will auto-inspect two seconds after your most recent keystroke.)

   To Change the Order of the Columns
   You can rearrange column order in the dataview by moving the fields in the SQL Select clause – be sure that you the comma separators are correctly placed. The code displayed on the **SQL Server** tab in the following illustration is indicating the order in which the columns will be displayed. The same relative order is shown under #5, in the grid.

5. The **Fields** grid view provides additional information about the dataview fields being returned by an SQL query. Here the administrator can review the dataview and determine what fields are required, are visible in the Curator Tool, or are read-only; what field is the primary key; and how the dataview fields will be displayed in the Curator Tool (including the field headings in the valid languages). (Refer to the Transform? and User Interface topics (below) for detailed explanations of those columns.)



6. **Status Bar** displays information pertinent to the most recent action or selected object.

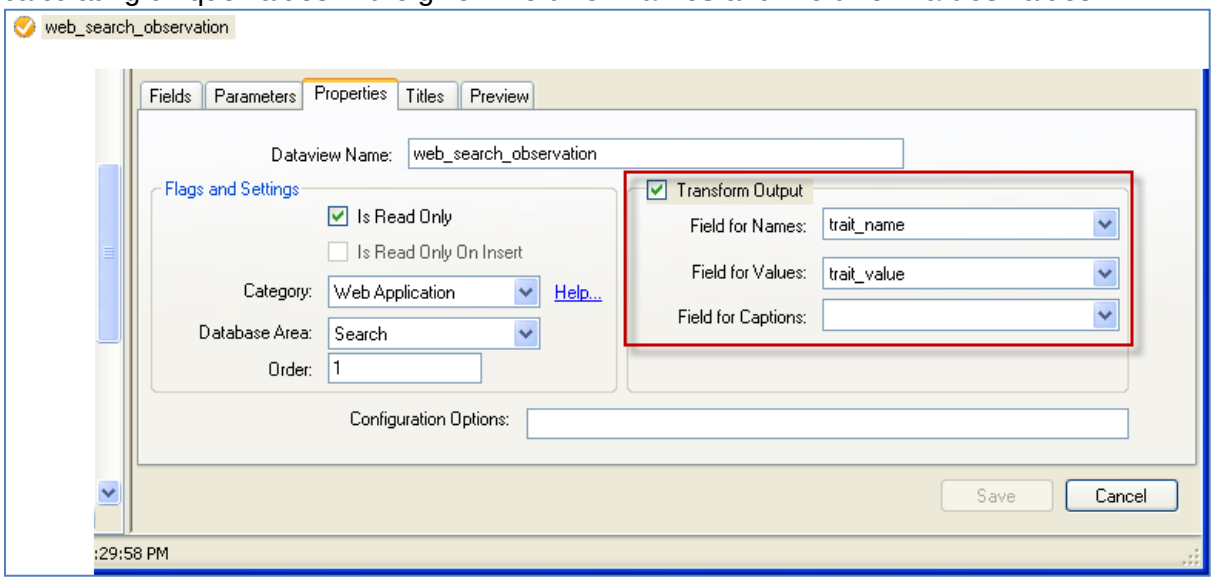
Transform?

This is rarely used. However, the **web_search_observation** dataview uses this functionality. (Open this dataview in the Admin Tool and inspect under the **Properties** tab the **Transform Output** section for an example.)

The search page on the GRIN-Global Public Website uses this feature to display observation data. (There is an option on the **Advanced Search** link. Users may want to see the traits column-wise instead of row-wise (they're stored row-wise). "Transforming" the dataview displays the data column-wise instead of row-wise.

In order for the transform feature to function properly, you must also specify which field(s) will be transformed (i.e., which ones do you group them by). The **Transform?** checkbox in the **Fields** tab serves this purpose. It indicates to the middle tier which fields to group records by when

calculating unique values in the given **Field for Names** and **Field for Values** values.

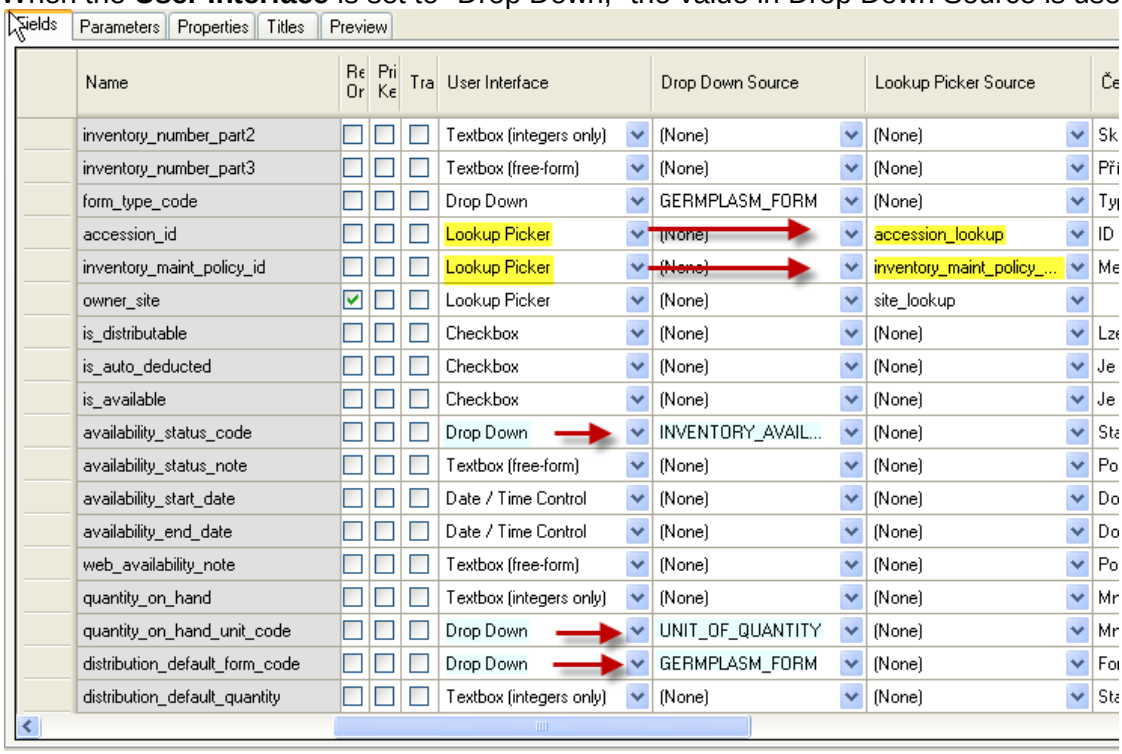


User Interface

The **User Interface** column indicates the type of, or the source of, the data for each field in the record. Three types of textboxes can be indicated (freeform, integers only, or decimals only) as well as **Checkbox** and **Date/Time controls** are possible options and are self-explanatory. The less intuitive options, **Lookup Picker** and **Drop Down**, are explained below.

Drop Down Source & Lookup Picker Source

When **User Interface** is set to "Lookup Picker," the "Lookup Picker Source" value is used. When the **User Interface** is set to "Drop Down," the value in Drop Down Source is used.



If there is an entry in either the **Drop Down Source** or the **Lookup Picker Source** column, but **Drop Down Source** or **Lookup Picker (**the respective choice) is not selected in the field's **User Interface** column, then the **-Source** column is ignored.

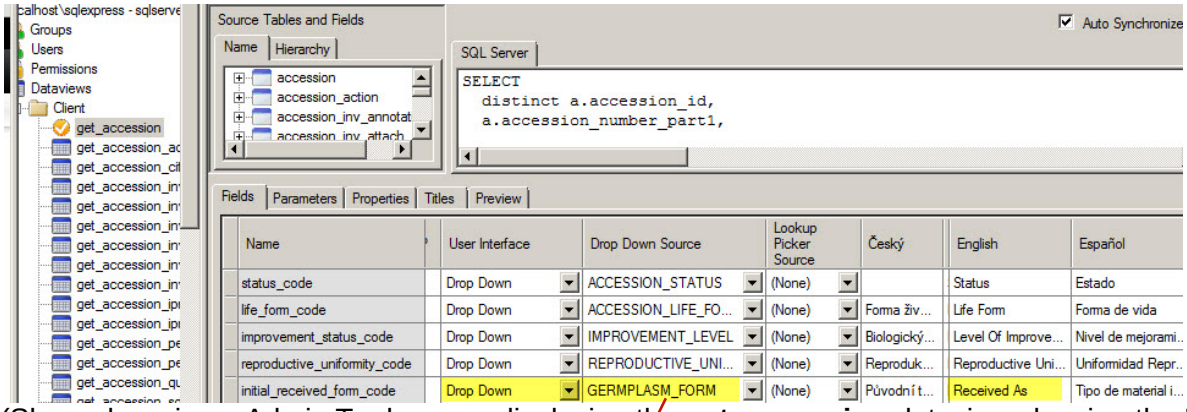
Fields marked with **Lookup Picker Source** are almost always foreign keys to another table. Also, these field names typically end with "**_id**."

The **Lookup Picker Source** column gets its list of data from running the "get_lookup_table_list" dataview. This list includes all the dataviews having "Client" as their category and "Lookup Table" for their database area.

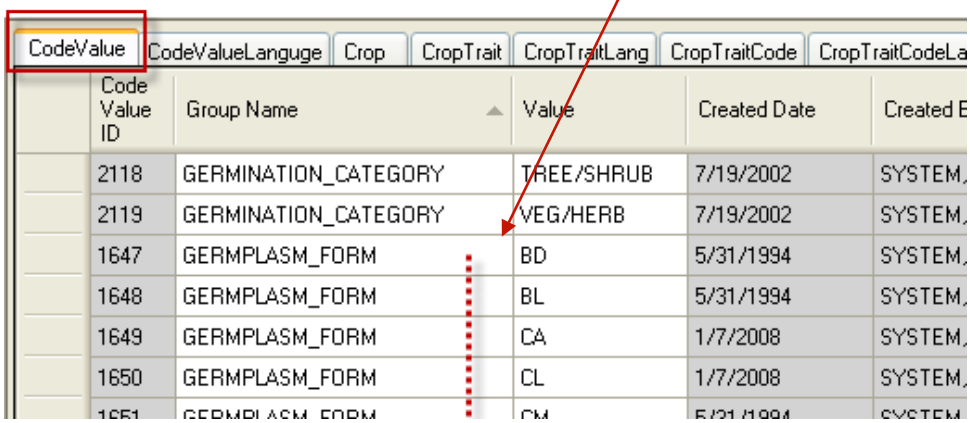
Orders | CropTraitCodeLookup | OrderRequest | CropTraitCodeLang | CodeGroup | SecLangLookup | SecDVFieldLangDV | LookupTableList

dataview_name	title	description	pk_field_name	table_name
accession_group_lookup	Accession Group Lookup Dataview	Accession Group Loo...	accession_name_group_id	accession_name_group
accession_lookup	Accession Lookup Dataview	Accession Number Lo...	accession_id	accession
accession_name_lookup	Accession Name Lookup Dataview	Accession Name Loo...	accession_name_id	accession_name
accession_source_lookup	Accession Source Lookup Dataview	Accession Source Lo...	accession_source_id	accession_source
big_cooperator_lookup	Big Cooperator Lookup Dataview	Big Cooperator Lookup	cooperator_id	cooperator
citation_lookup	Citation Lookup Dataview	Citation Lookup	citation_id	citation
code_value_id_lookup	Code Value ID Lookup Dataview	Code Value ID Lookup	code_value_id	code_value

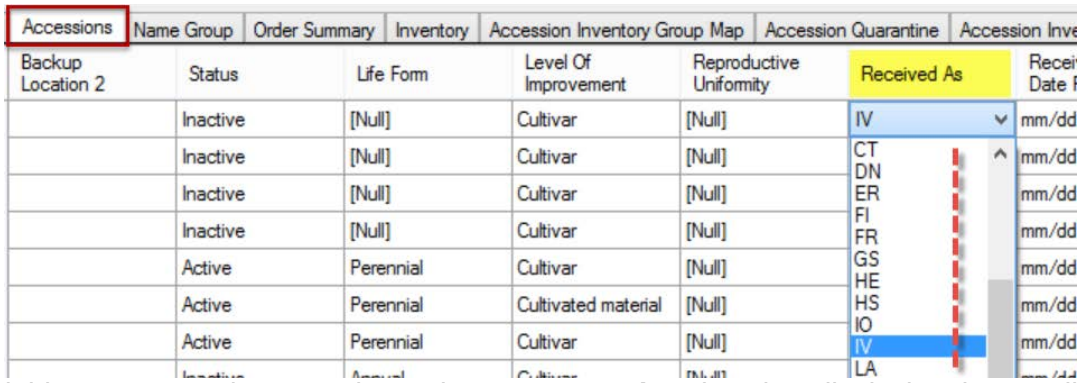
The **Drop Down Source** column gets its data from the **group_name** field **in the get_code_value** dataview.



(Shown here is an Admin Tool screen displaying the **get_accession** dataview showing the Drop Down Source, **GERMPLASM_FORM**, for the initial_received_form_code field.)



(This is a Curator Tool screen using the **get_code_value** dataview showing **Group Name** values that have been stored in the code value table.)



(This Curator Tool screen shows the **get_accession** dataview displaying the possible values for the **Received As** field ("Received As" is the English friendly name; the database name is "initial_received_form_code" field.))

Parameters

Parameters are values that must be specified at run time – that is, in order for the SQL query to run correctly, there are variables ("fill in blanks") that must be supplied with appropriate information.

On the **Parameters** tab, you can define the data type for each parameter. (This is required by the database engines and cannot be easily derived from the raw SQL).

Consider the following example.

Suppose you wanted the dataview to return all accession records that have a prefix of "PI."

Select * from accession where accession_number_part1 = 'PI';

If you also wanted to do the same process, but for those accessions starting with "BW," you would write:

Select * from accession where accession_ number_part1 = 'BW';

The only difference is the prefix value. We are essentially running the same query, but with different values for certain parts of the SQL than we did on the previous run. We can actually make the SQL run faster by using a parameter for this:

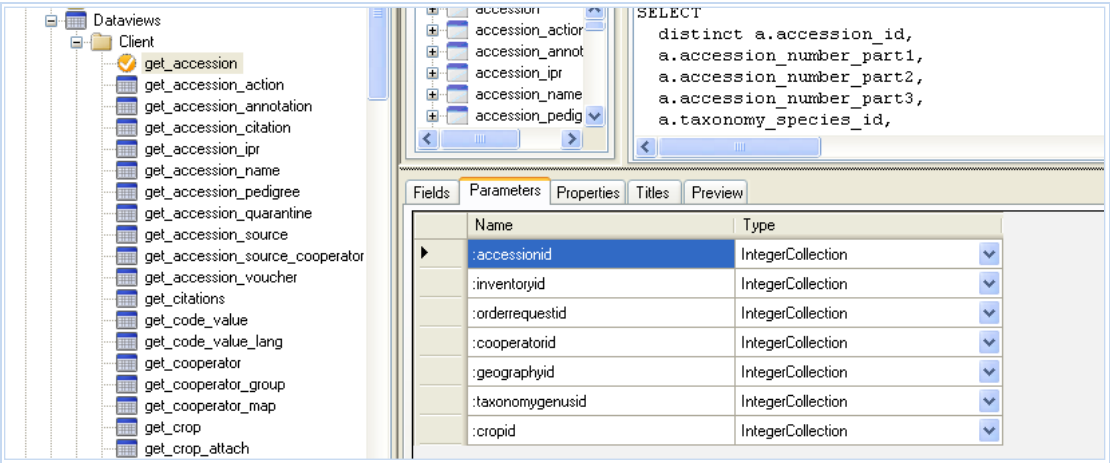
Select * from accession where accession_ number_part1 = :part1

NOTE: The parameter is always prepended with a moniker – colon (:), question mark (?), or at-sign (@). Different database engines use different ones, but GRIN-Global hides this variation so any can be used against any GG-supported database engine.

Using parameters also lets us prevent SQL injection attacks, as these typically work by providing carefully crafted values. For instance, if we don't use parameters and someone gives us the following value: '; delete * from accession; --

Select * from accession where accession_number_part1 = ''; delete * from accession;

The database engine would run a select, followed by a delete which would wipe out the table.



Data Types

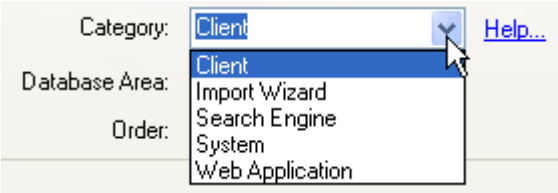
When parameters are used, the database needs to know which data type each is. The data types supported by GRIN-Global for SQL parameters are as follows:

Field	Description
String	Text
Integer	A whole number (i.e. no decimal point – 3, 7892, -12, etc)
DateTime	A date and optionally a time
Double	an approximate fractional number (i.e. decimal point – 3.222222222295, 4.01, etc)
Decimal	An exact fractional number (i.e. decimal point – 3.23, 40.9, etc)
IntegerCollection	A sequence of integers that should be put into the SQL as comma separated list. i.e. (47, 23, 99, 101)
StringCollection	A sequence of strings that should be put into the SQL as a comma separated list. i.e. ('hello', 'how', 'are', 'you')
StringReplacement	Performs an explicit replace on the SQL string itself. Necessary for providing values to sections of the SQL where the database engine does not allow a parameter, such as specifying a table name, a column name, a join type, etc. Should be used very sparingly.

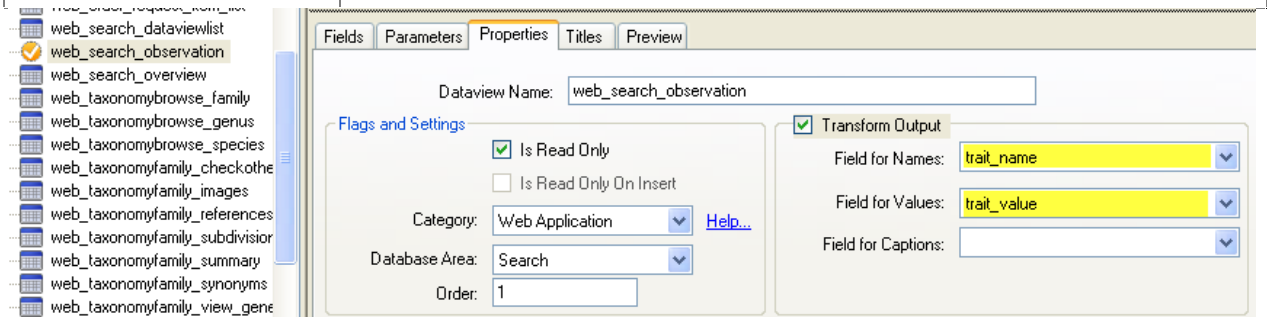
PostgreSQL is notoriously strict about the parameter types it accepts. MySQL is extremely permissive of types it accepts. SQL Server and Oracle are somewhere in the middle, as they generally will coerce types as needed. Recommendation: Always use the proper data type.

Properties

On the **Properties** tab, you can edit the dataview properties:

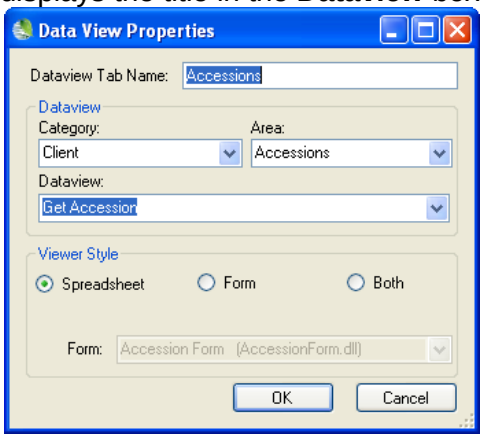
Item or Option	Notes
Dataview Name:	dataview name
Flags and Settings Is Read-Only	
Is Read-Only On Insert	This checkbox specifically supports the **Cooperator** dataview – the field should be updatable, but on initial record creation it should be read-only as its value is auto-populated by the middle tier based on business logic.
Category	The **Category** describes in a general sense how the dataview is being used**:**  **Database Area**: The search page uses this information to determine which dataviews will be shown in the Advanced Search link's dropdown for which

	"view to display results in" The **Order** determines the order in which they are displayed. 0 = first…
Transform Output	The search page on the GRIN-Global Public Website uses this feature to display observation data. (There is an option when on the Advanced Search link. Users may want to see the traits column-wise instead of row-wise (they're stored row-wise). "Transforming" the dataview makes it possible to show data column-wise instead of row-wise. This is rarely used. However, the web_search_observation dataview uses this functionality. Open it in the Admin Tool and inspect the Transform Output section for an example. See also Transform?

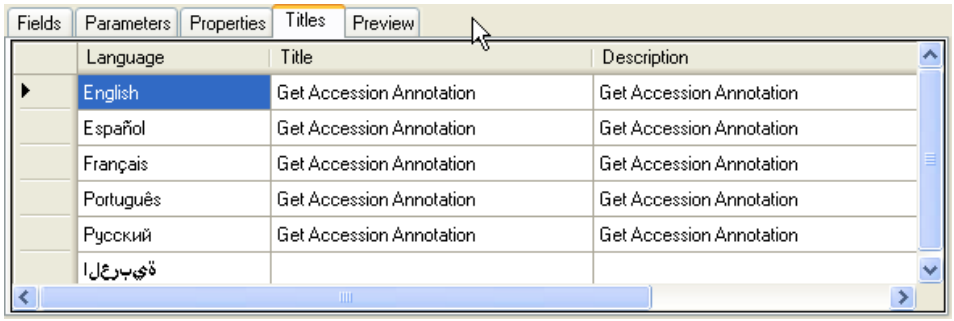


Titles and Descriptions

A title and a description can be given to each dataview for each language. The Curator Tool displays the title in the **Dataview** box in the Data View Properties window:



Title and **Description** provide additional information about the dataview itself in a language-aware manner.

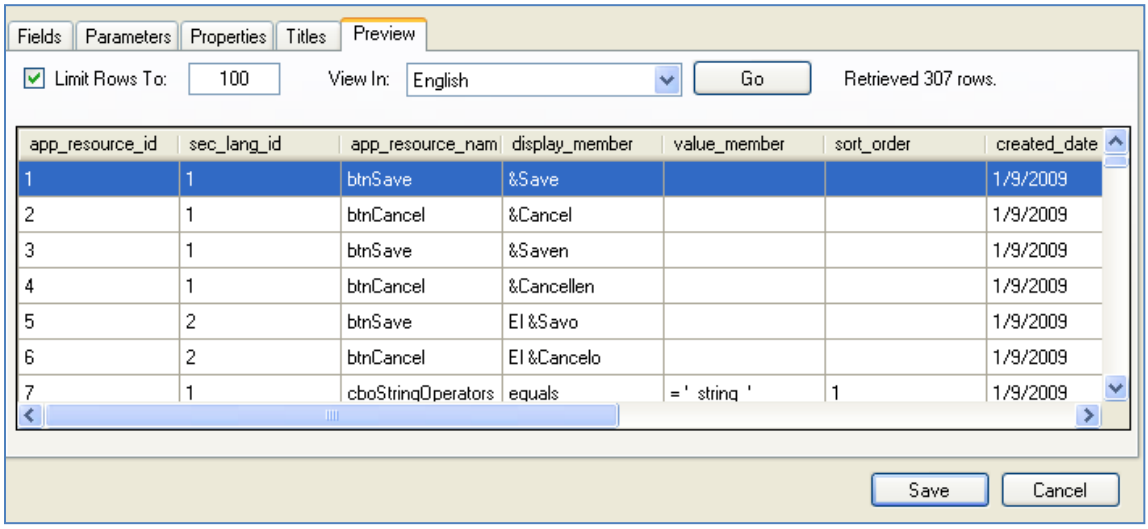


Preview

Using **Preview**, you can interactively test your new dataview settings without first saving them. Preview uses the SQL, fields, and parameters that are specified in the form and executes the query. If parameters exist, a dialog window will prompt for the value(s) to be entered.

If there is a problem running the query, a message box will display showing the raw error from the database engine itself.

Results can be displayed in whichever language is chosen. The default language is the same language as the current user's language.



Importing and Exporting Dataviews

This capability is very practical, especially when an organization is initially configuring its GRIN-Global environment and the organization's many dataviews. Exporting/importing provides a handy method for using a dataview from one environment as the basis in another. This feature is also useful when a dataview is mistakenly altered by the administrator and needs to be replaced. A copy of a working dataview could be imported and then used as the basis to overlay and correct an improperly altered dataview.

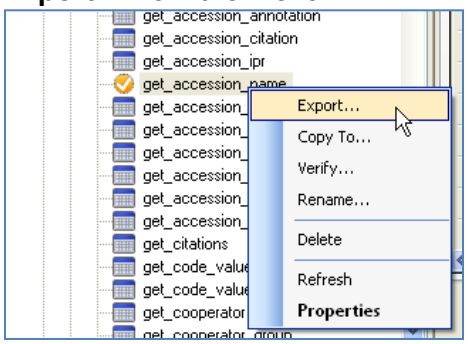
Tip

Any time you intend to use the GRIN-Global Updater to download any server component, first use the Admin Tool to export a copy of all of the dataviews for safe keeping. Similarly, immediately after updating server components via the Updater, export another copy of all of the dataviews. By following this tip, you will have a complete set of dataviews which will be available in case you need to restore a dataview that stops working after the software is updated.

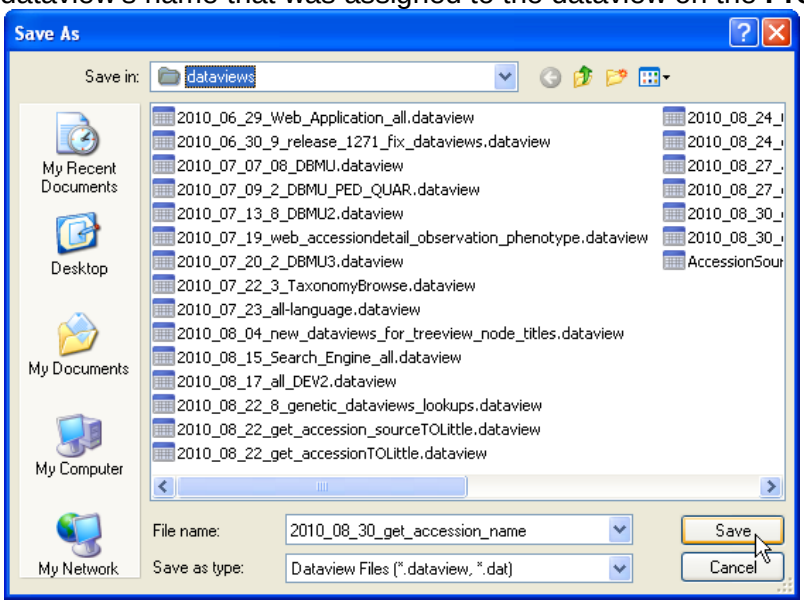
Exporting Dataviews

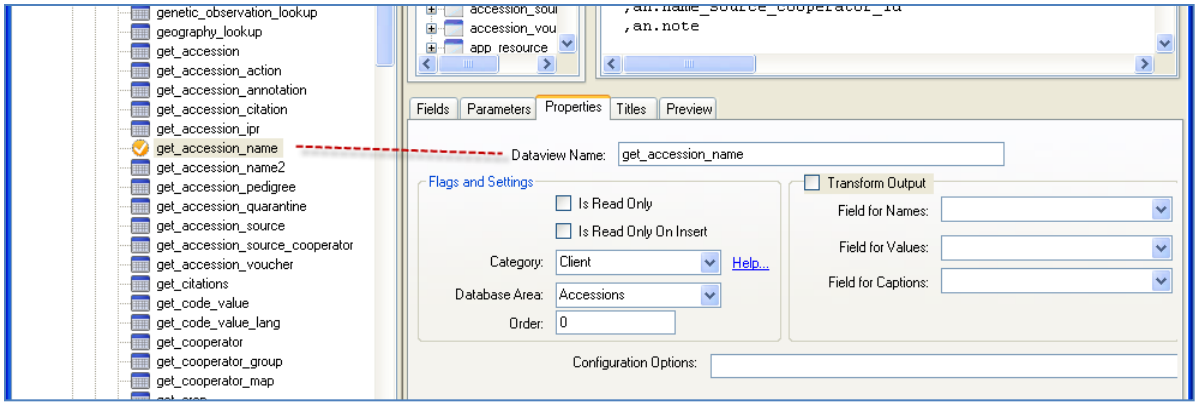
To export dataviews, right-click on either the **Dataview** node or one of the Categories below the Dataview node; select **Export All…** from the context menu.

If you are exporting a single dataview only, highlight the dataview name, right-click, and select **Export…** from the menu:



The suggested filename for the dataview will include the current date as the prefix as well as the dataview's name that was assigned to the dataview on the **Properties** tab.

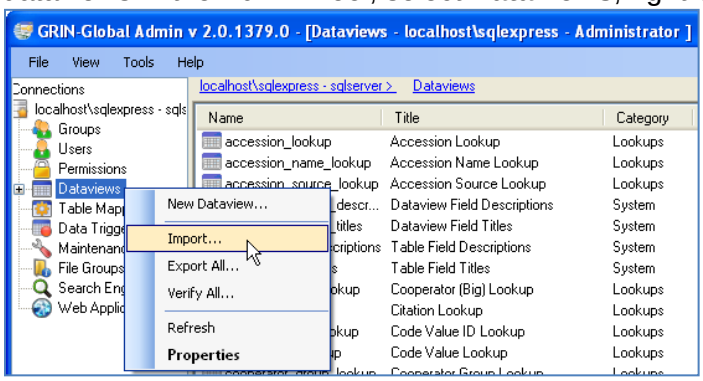




To select multiple dataviews, use **Ctrl** or **Shift** as needed.

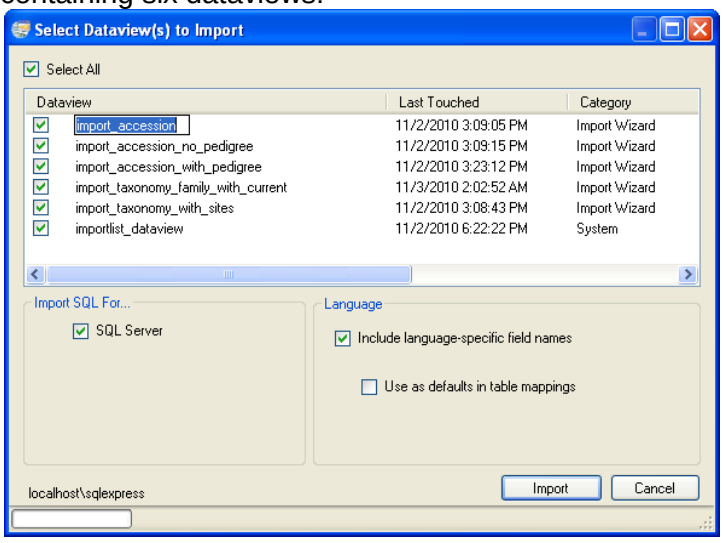
Importing Dataviews

When importing dataviews, you may be either importing just a single dataview or multiple dataviews in the Admin Tool, select **Dataviews**; right-click and select **Import…**



Importing a Dataview File Containing Multiple Dataviews

Some **.dataview** files will contain multiple dataviews. The following example illustrates one file containing six dataviews:



When importing, when appropriate, use **F2** to edit/rename the dataview name. By doing so, you can avoid overlaying an existing dataview name.

Language Checkboxes

As background, there are "friendly" names defined at two levels: at the *table*/field level and at the *dataview*/field level. The table/field friendly name is used as the default friendly name whenever a new dataview is created. For example, any dataview that points to accession.accession_id will automatically receive "Accession ID" for its English friendly name.

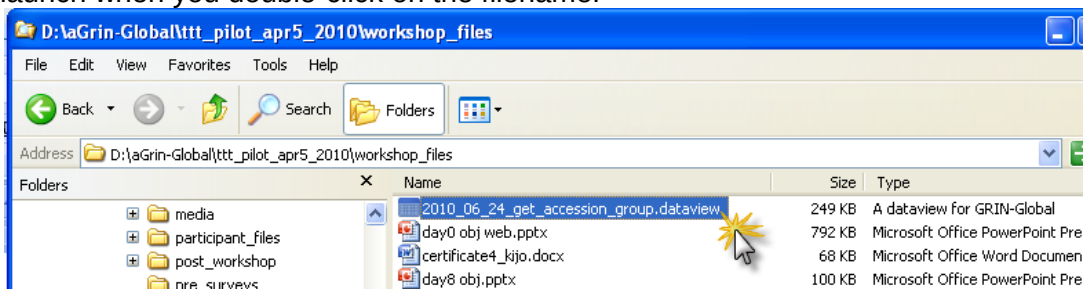
However, the person editing a dataview can change the friendly name to "'ACID" if desired and "ACID" will be stored as the friendly name used within that dataview. However, if another user creates a different dataview which also points to accession.accession_id, by default the field will be "Accession ID," not "ACID."

Why the duplication? The table/field friendly name is a "fallback" if nothing is specified at the dataview/field level. Almost all of the GRIN-Global dataviews currently use table/field as the friendly name source, so when we change the field's name essentially all dataviews are updated at the same time. The dataview editor compares the friendly name the user entered for each field, and if it matches exactly what's in the table/field friendly name, it writes nothing (so the table/field one is actually used).

Selecting the "**Include language-specific field names"** checkbox pulls in the "friendly" names you see in columns in the Curator Tool. When unchecked, it will skip them (so a user can pull in a new dataview definition without overlaying their own "friendly" names.)

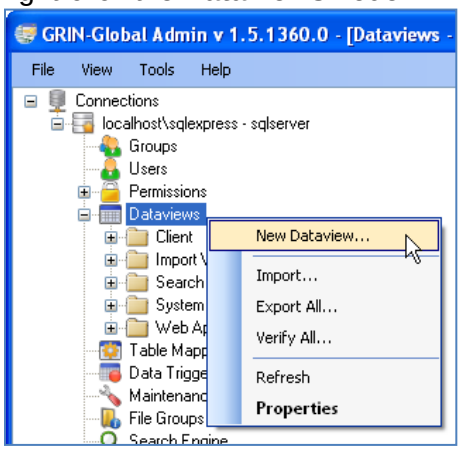
Selecting the "**Use as defaults in table mappings**" checkbox will write the table-level "friendly" name for each field. That is, the Admin Tool updates the default friendly name for any dataview which points at the field(s), not just the "friendly" name directly associated with the dataview. When unchecked, it does nothing at the table level for "friendly" names.

You can import a stored dataview when in Windows Explorer. The Admin Tool will launch when you double-click on the filename.

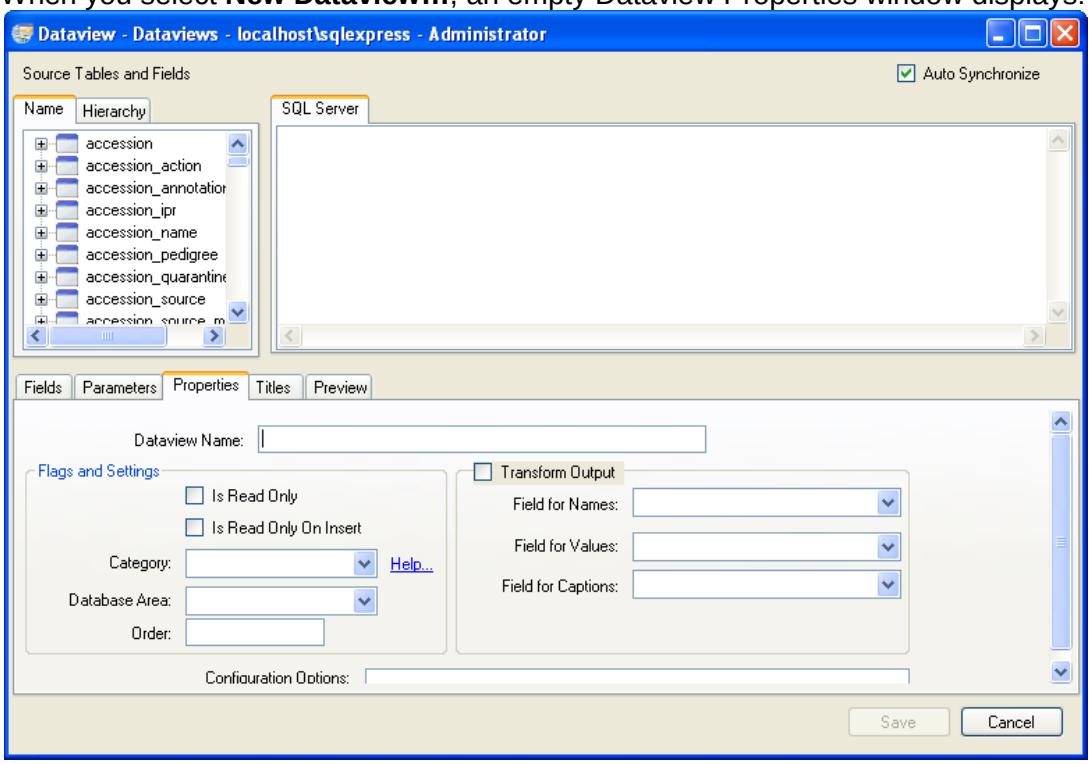


Creating New Dataviews

The recommended (and the easiest way to create a new dataview), is to copy an existing one and then edit it. However, a **New Dataview** option is available on the context menu when you right-click the **Dataviews** node.



When you select **New Dataview…**, an empty Dataview Properties window displays:



You would then proceed to complete the fields under the various tabs.

Determining Where Lookup Tables are Used

Dataview – Show Dependencies

You can determine what dataviews and tables use a lookup table by using the Show Dependencies option on the context menu.

In the AT, select **Dataviews | Lookups**. Then on the right-hand side, select the desired lookup dataview; right-click. A window will display the other dataview and tables in the schema that use this lookup table.

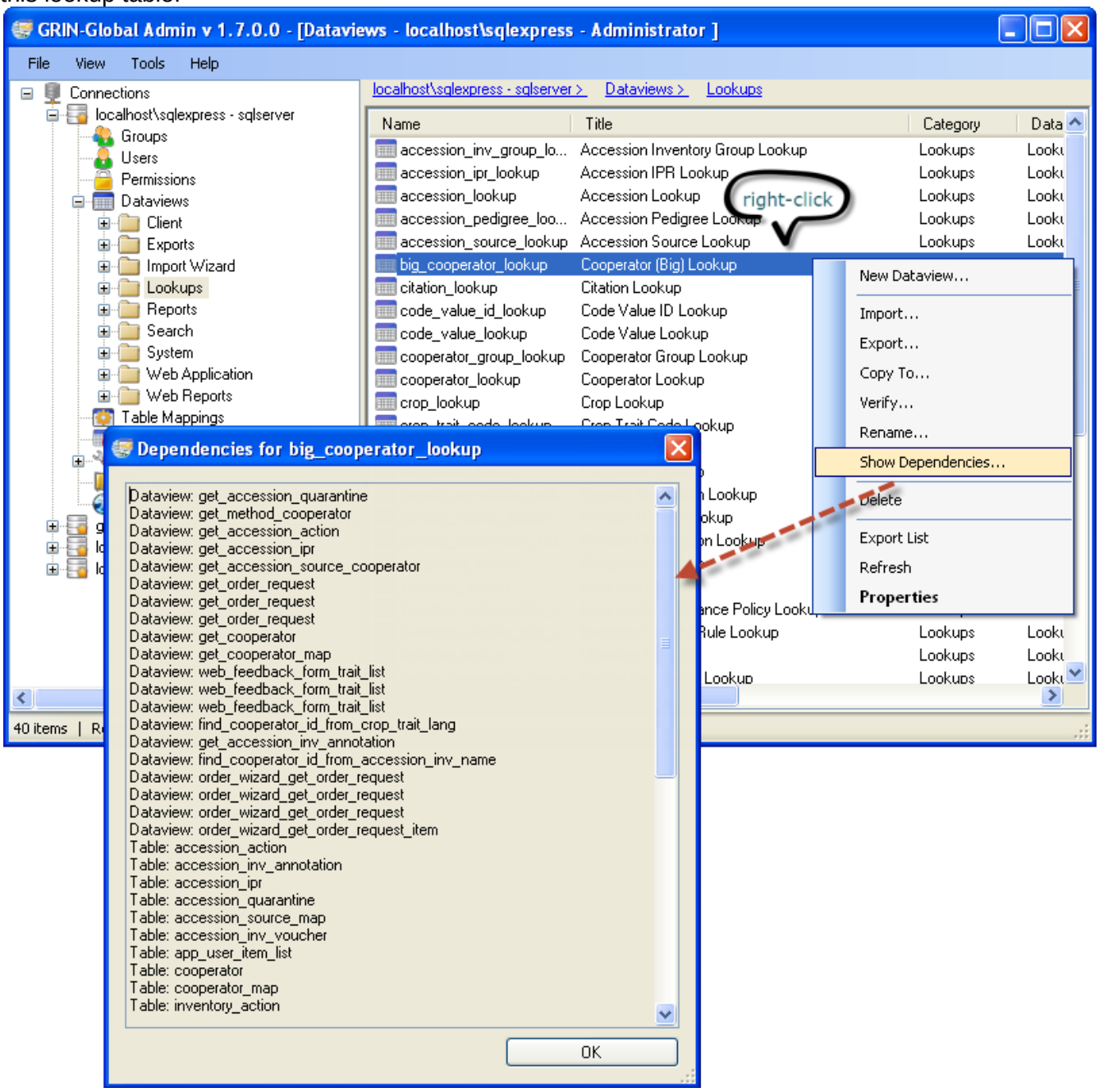


Table Mappings

Table Mappings Overview

Table mappings are used by dataviews to associate a dataview field to a table field for the purposes of showing friendly column names, allowing updates, etc. The dataview editor, as well as the middle tier, associate dataview fields with specific table/fields in the database. This association makes possible the updating of the table field via the dataview.

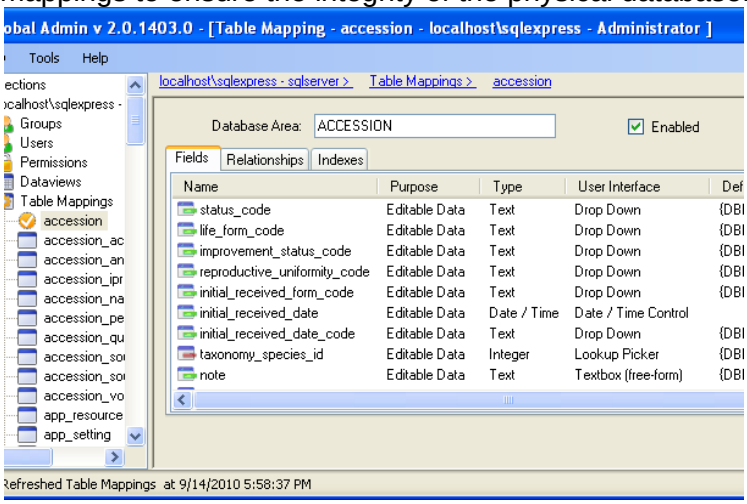
Since each database engine (SQL Server, My SQL, Oracle, PostgreSQL) stores information about its tables, constraints, foreign keys, and indexes differently, Table Mappings is a GRIN-Global utility which overcomes these individual differences and provides a common tool for the administrator.

Table and column Information, the “table mappings,” are required for every INSERT, UPDATE, or DELETE SQL statement issued to the database engine so that validation can be performed for data typing, maximum length, etc. Since these table mappings are costly to retrieve in real time with some of the database engines, part of the GRIN-Global database schema includes a place to cache table mapping information. These table mappings serve the following purposes:

Purpose	Description
Data Updatability	A field in a dataview must correspond to a given table mapping for that field to be updatable
Security	A permission that restricts down to the row level uses table mappings to specify the rows to which the permission apply
Language Information	Language-specific Title and Description values are used as defaults when defining a dataview

The Concepts of “Mapped” and “Unmapped”

In the Admin Tool, when a table is selected under **Table Mappings**, the **Fields** tab lists all the “mapped” fields for the selected table. However, there may be occasions when the GRIN-Global administrator has modified the physical database tables using the database management application such as SQL Server Management Studio. After doing so, the administrator should follow that activity by using the GRIN-Global Admin Tool to review and update the table mappings to ensure the integrity of the physical database.

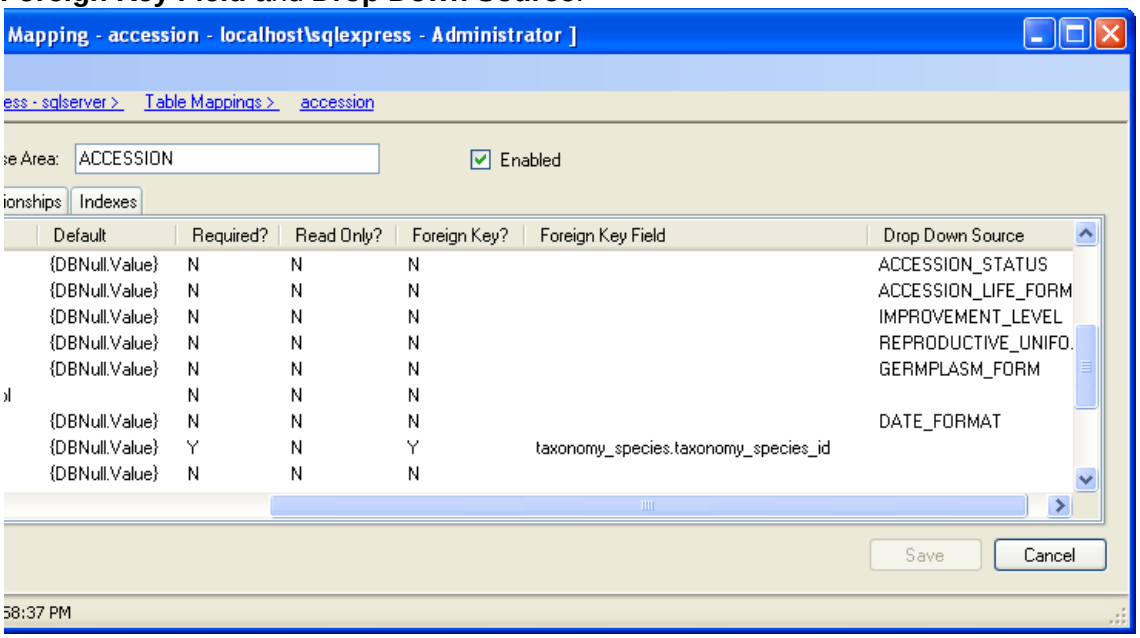


(partial screen capture; the right part of this screen is shown on the next page)

Table Mappings

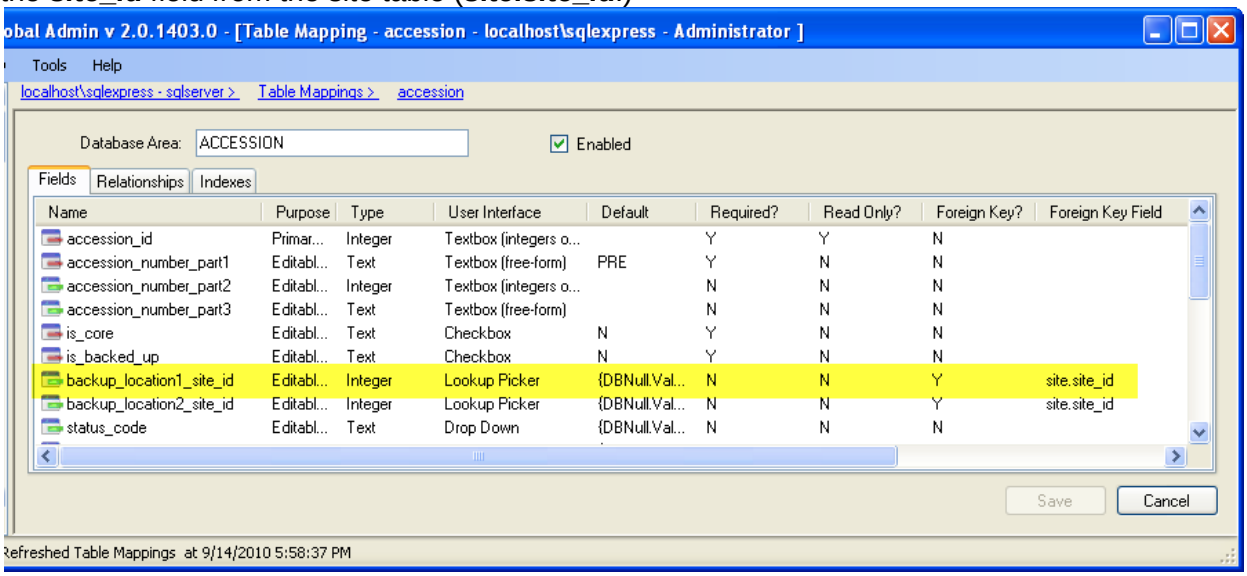
Typically, this list will match exactly the fields defined in the actual database schema for the table. *Fields must be mapped if they are to be available* for updating in GRIN-Global using dataviews.

This window provides additional information not readily available from the database management tool such as SQL Server Management Studio. For instance, this window displays **Foreign Key Field** and **Drop Down Source**.



(The Curator Tool uses this information to determine which dataview to run when displaying a dropdown, data for the LookupPicker, etc.)

For example, in the screen shown below, the Accession dataview was selected. When the Curator Tool user edits the **backup_location1_site_id** field, a lookup picker window will display the **site_id** field from the site table (**site.site_id**.)



In the Admin Tool, right-click on **Table Mappings** to display its context menu.

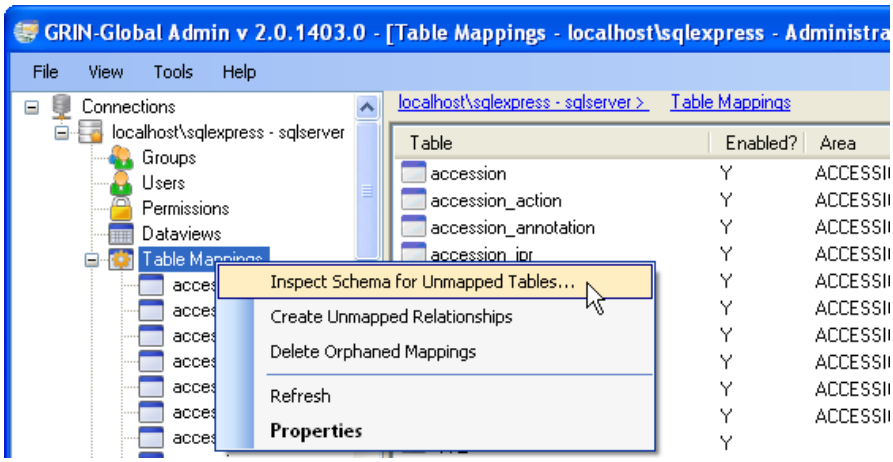


Table Mappings Node Context Menu Options

Menu Item	Description
Inspect Schema for Unmapped Tables	List all tables in the schema which are not currently mapped in Table Mappings
Create Unmapped Relationships	Use this option when a table has been modified in the database schema and has one or more relationships that have been modified, added, or removed, but which were not yet mapped in Table Mappings.
Delete Orphaned Mappings	Use this option to delete “Orphaned Mappings” – this happens when tables have been dropped from the schema, but are still mapped in Table Mappings. This will list all entries that are in Table Mappings, but which don't actually exist in the schema
Refresh	Refreshes the view in the Admin Tool with the latest data from the database.
Properties	See Refresh.

Field Context Menu Options (under Table Mapping)

Table Mappings Field Context Menu Options

Menu Item	Description
New Field…	Map a new field to GRIN-Global
Show Dependencies…	Displays a list of all instances where the field is referenced in dataviews
Delete	Deletes the field’s table mapping information
Export List	Generates a .csv file of the table mappings
Refresh	Ensures the display is current
Properties	See Table Mappings: Field Detail Window

Table Mappings

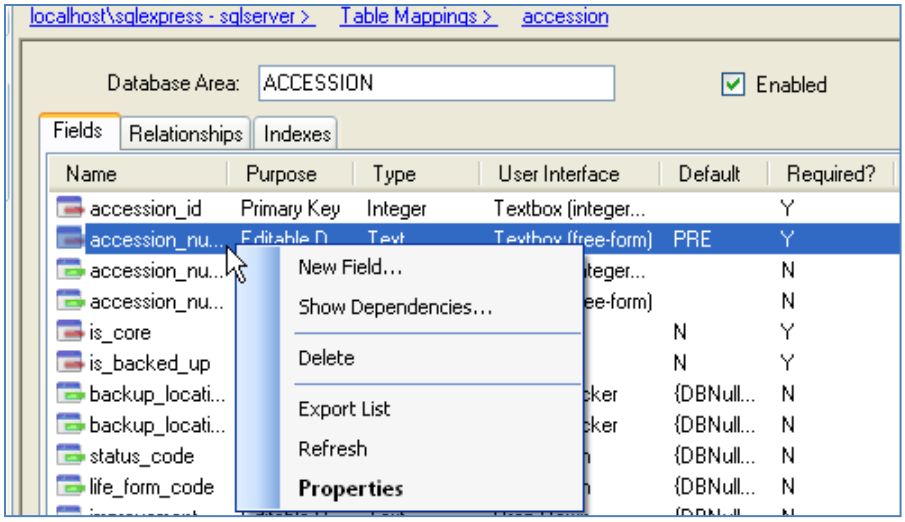
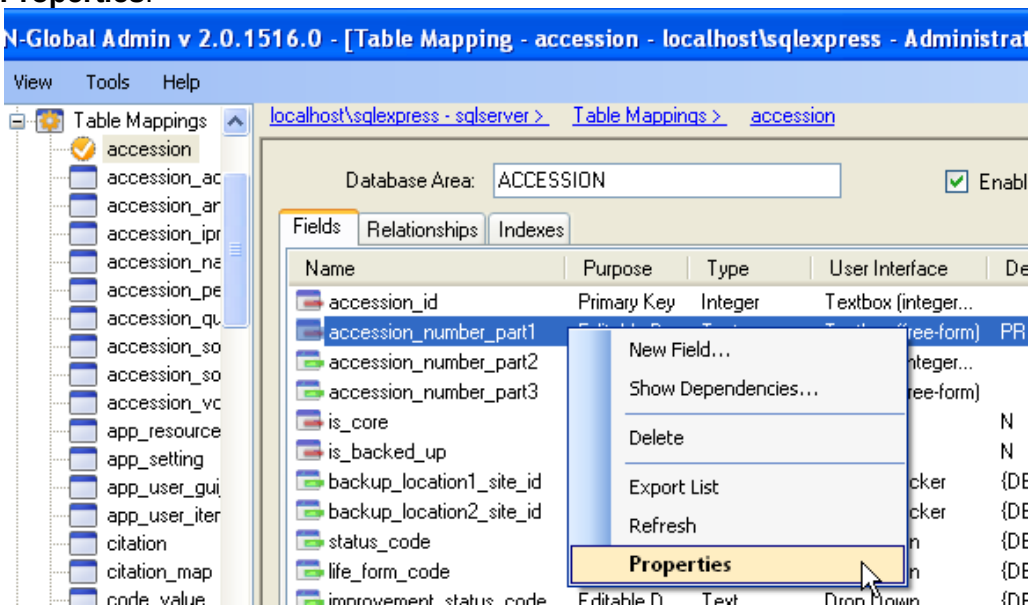
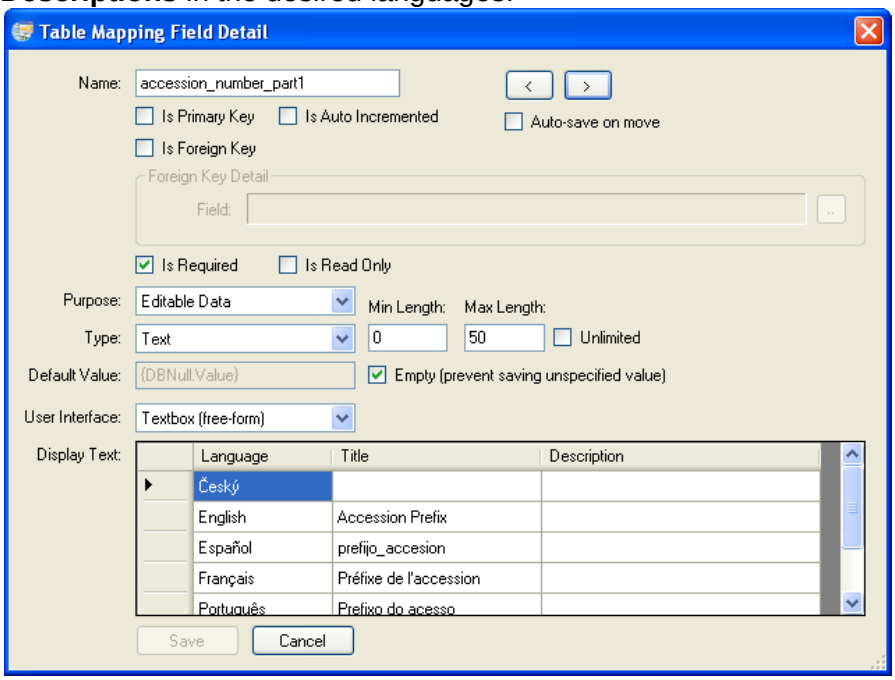


Table Mappings: Field Detail Window

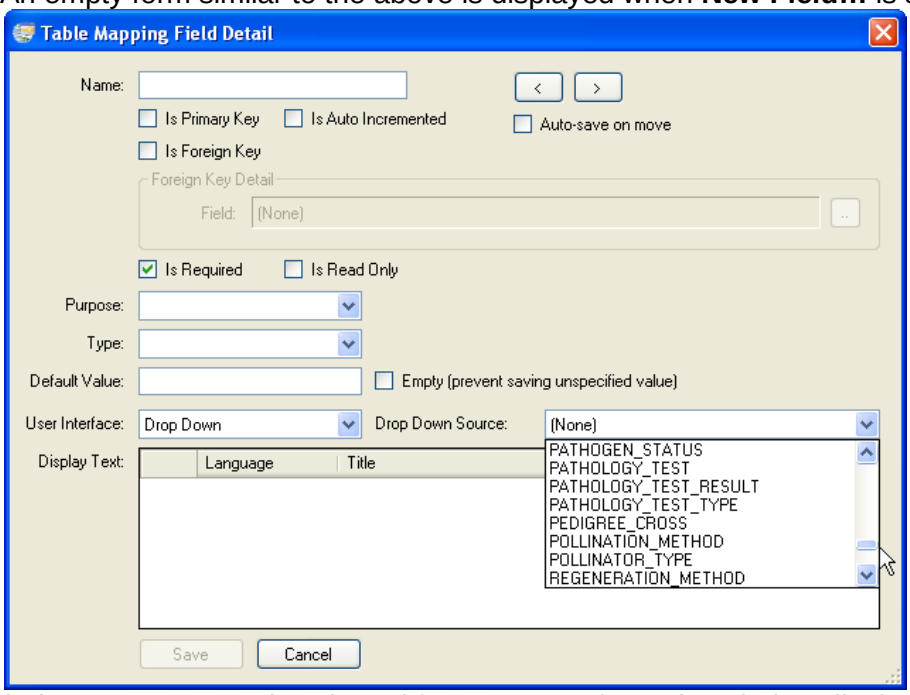
In GRIN-Global, you can have language-specific names at the *table*/field level in addition to the capability of having language-specific names at the *dataview*/field level. These table/field level names can serve as the friendly names and be used as defaults when creating a new dataview. In the Admin Tool, select **Table Mappings** | name of the table | field name; right-click and select **Properties**:



In the **Table Mapping Field Detail** window, complete the **Titles** and optionally input **Descriptions** in the desired languages:



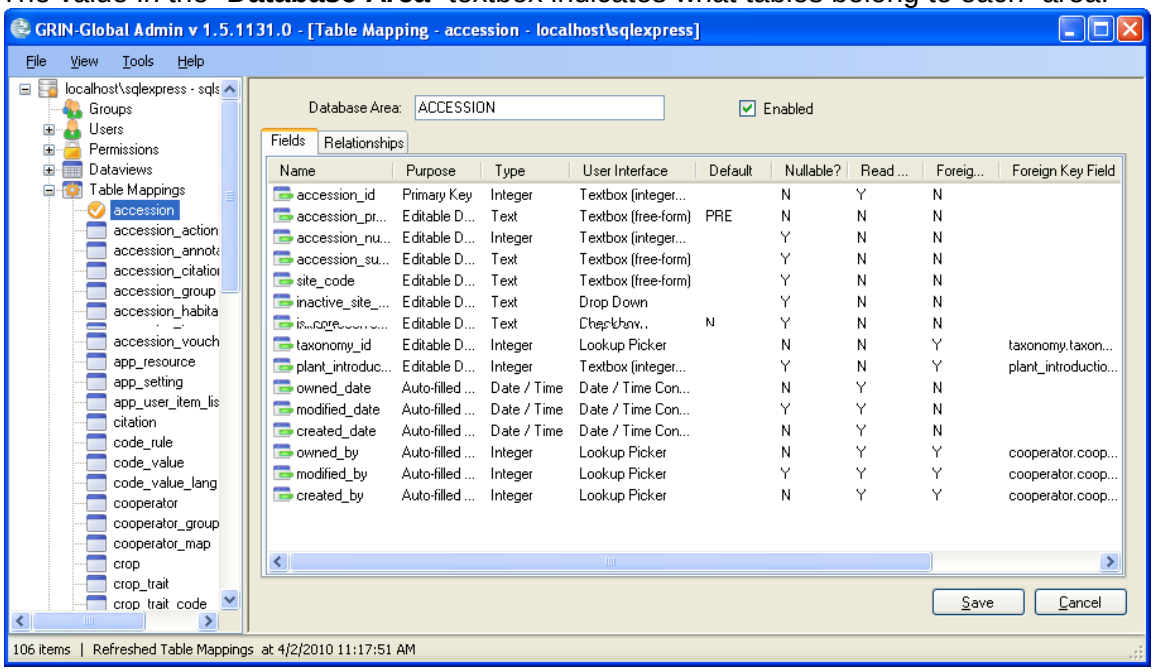
An empty form similar to the above is displayed when **New Field…** is selected:



(When “Drop Down” is selected for **User Interface**, the window displays the **Drop Down Source**: option.)

Table Mappings

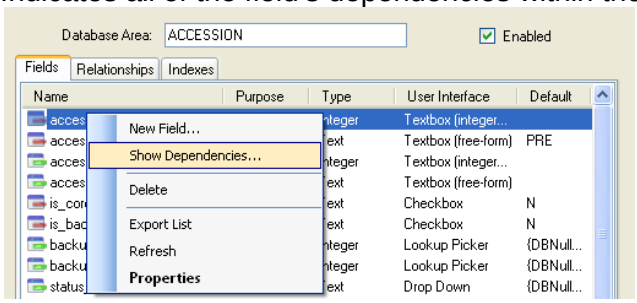
The value in the "**Database Area**" textbox indicates what tables belong to each "area."



The "area" field in table mappings is related to the "Area" of the Search Tool. (the checkbox list on the right hand side of it). If a table is marked with an area of "INVENTORY," selecting the "Inventory Area" checkbox in the Search Tool causes the search engine to add it to the list of areas to search in its indexes. Note that some tables do not have an "area" and therefore are never searched via the Search Tool.

Show Dependencies

Indicates all of the field's dependencies within the schema:



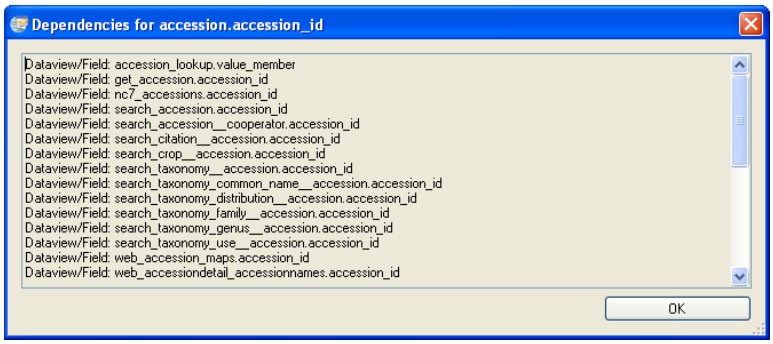
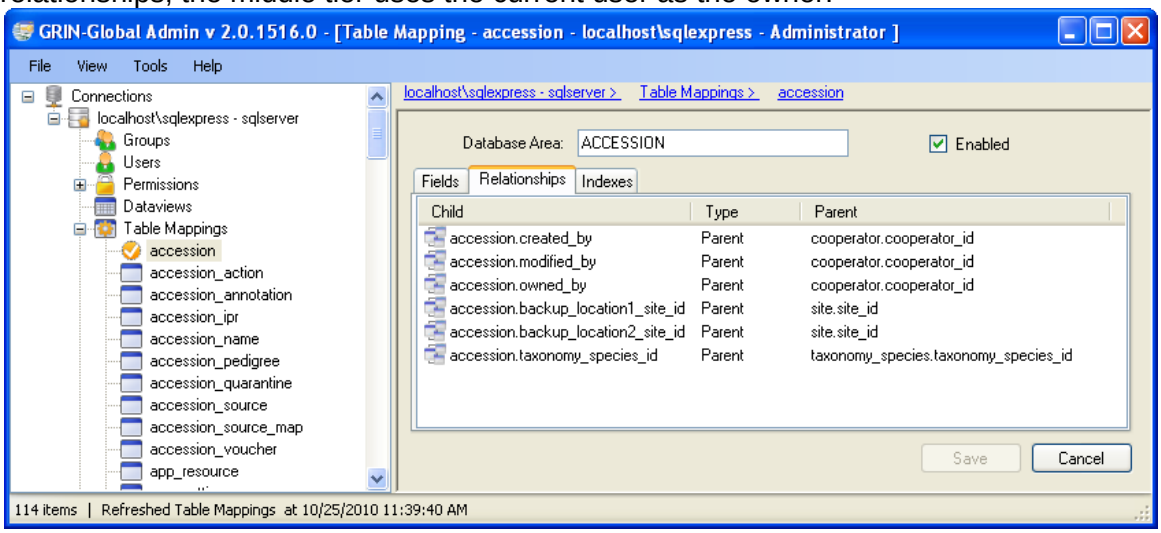


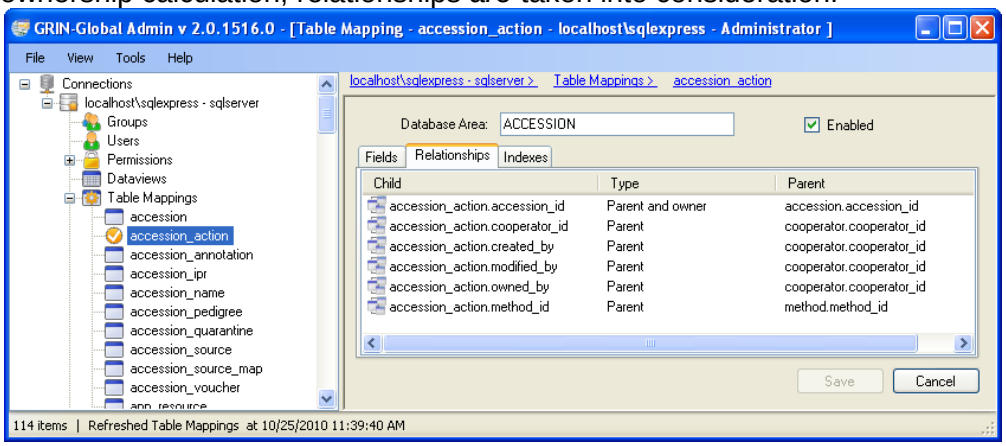
Table Mappings | Relationships
Similar to the way in which GRIN-Global stores information about fields, GRIN-Global also stores information about the relationships for a table. This information is used by the dataview editor to determine how to automatically generate SQL when a table or field is dragged onto the SQL pane.

It is also used by the middle tier to determine ownership of a database record when a new record is created. When a relationship exists with a relationship type of “OWNER_PARENT,” the record in the child table is also owned by the owner of the parent record. In all other relationships, the middle tier uses the current user as the owner.



Example
By default, whoever creates a record is the owner.

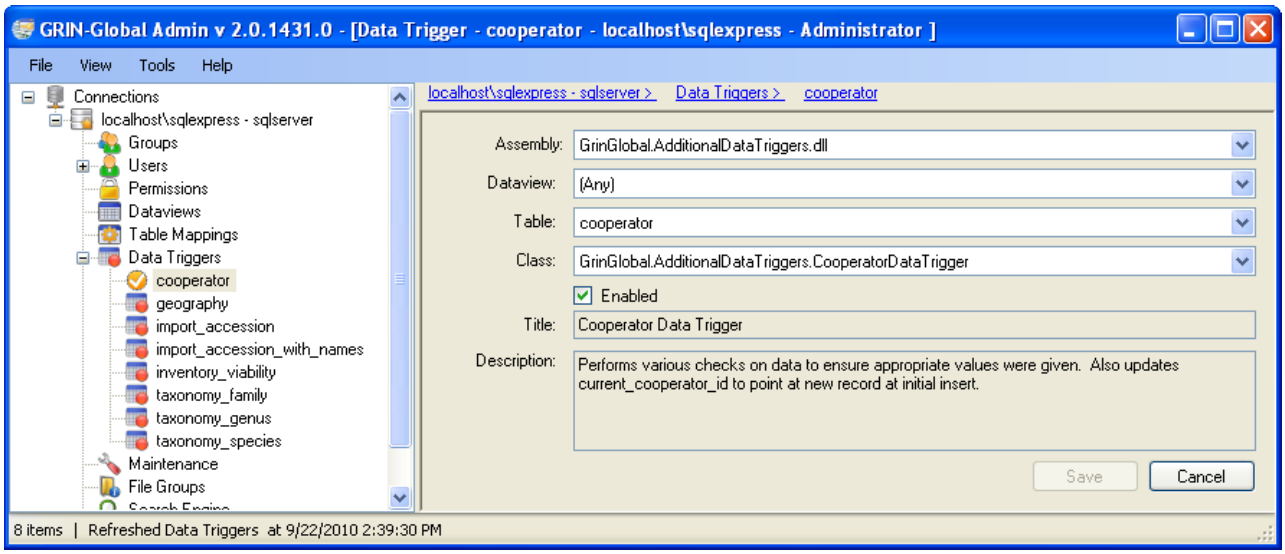
In the Admin Tool, relationships are mapped between dataviews. For instance, there is a relationship from accession_action to accession with the Relationship Type defined as "Parent and owner." This means if someone creates a record in accession_action, the owner will be marked the same as the owner of the parent record (the accession record's owner). If there are no relationships of "Parent and owner," then the record creator is the owner. So when doing ownership calculation, relationships *are* taken into consideration.



Data Triggers

A “data trigger” is a mechanism for developers to tap into the middle tier data pipeline (used for reading and writing data) and add custom business logic. Since each genebank may have different sets of rules, this was the most flexible and database engine-independent approach.

By implementing one of the specific interfaces (in GrinGlobal.Interface.DataTriggers namespace of the GrinGlobal.Interface.dll), and associating that “.dll” file with a specific dataview or table, custom business logic can be added without knowing any other GRIN-Global middle tier details. (A “.dll,” is also known as an executable file.)



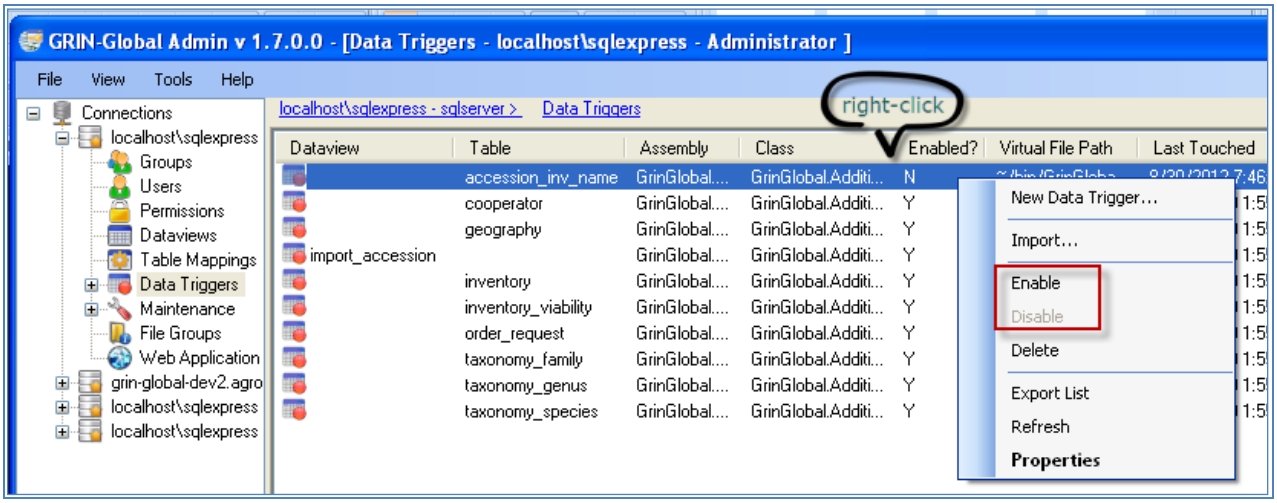
The following example illustrates how you can use this feature:

When saving cooperator information, there is business logic that states the first name and last name should be in “proper case.” “**Lutherville Maryland,”** as shown here, is in proper case**.** Using proper case is not necessarily universal. Some genebanks may prefer to use all capital letters. In order to implement special cases such as this, you can use a data trigger to run custom code. If a genebank wants their first name and last name to display in all caps, such as “**LUTHERVILLE MARYLAND**,” the genebank or the GRIN-Global support team could create a different assembly file. This custom logic would be easier to implement this way rather than changing the main body of code.

Note

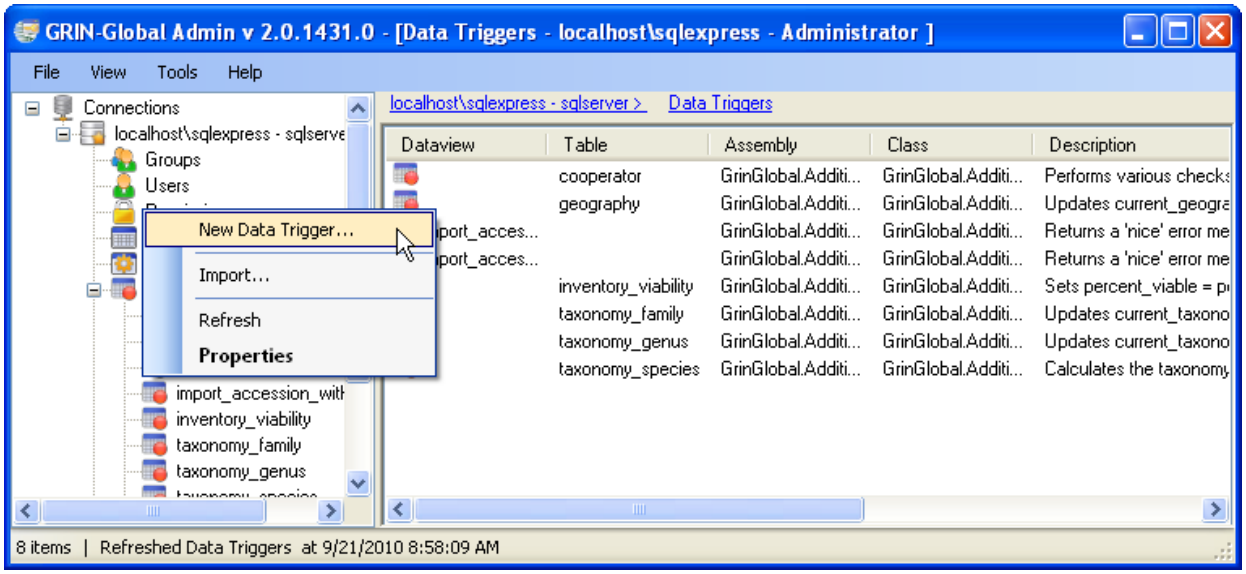
A datatrigger can be made invisible to the Admin Tool. In the current version, one that comes to mind is the datatrigger established to populate inventory records from their corresponding Inventory Maintenance Policy record. The developers decided that this should be a datatrigger that would not need to be disabled since you can easily override its impact by establish Inventory Maintenance Policies that indicate how the inventory data is updated when a germplasm order is made against the inventory.

GRIN-Global is bundled with some datatriggers. In the Admin Tool, the administrator can enable or disable them.



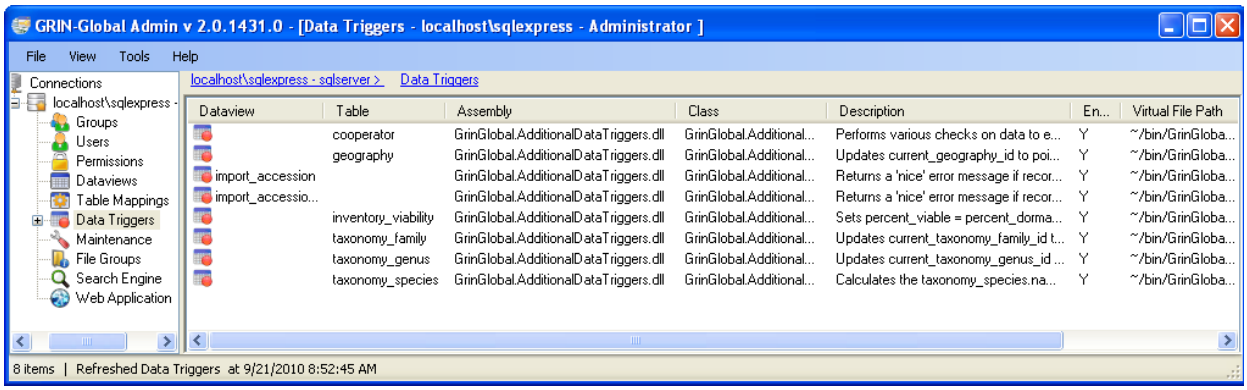
Import Option

When you right-click the Data Triggers treeview node, there is an **Import** option. (There is also a similar option when you right-click the listview on the right when Data Triggers node is selected.

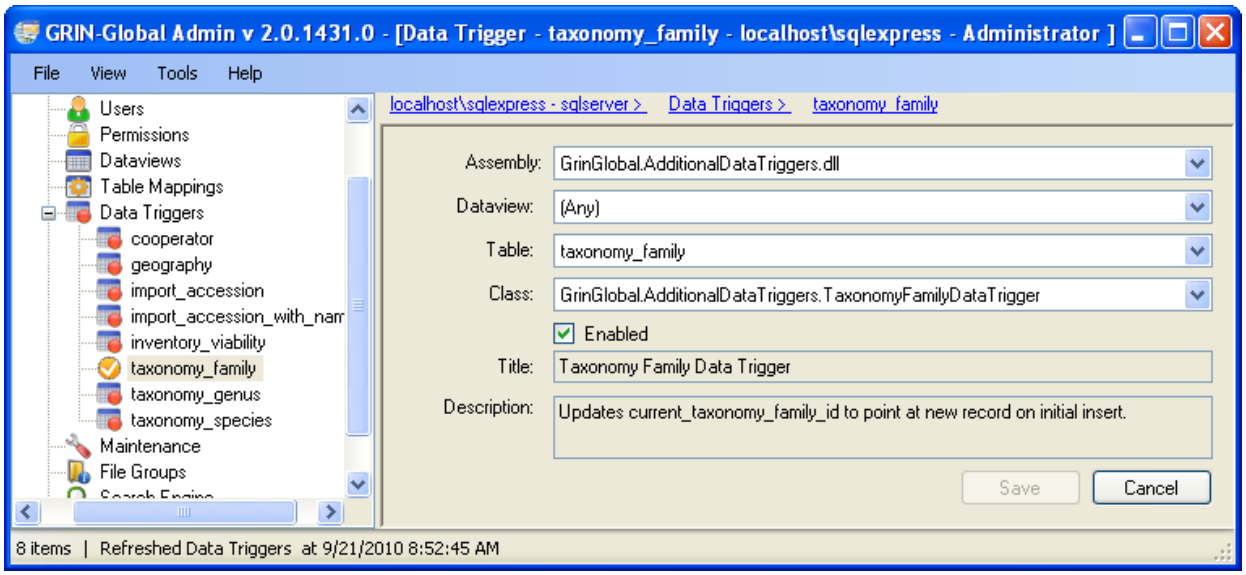


Data Triggers

When you select the .dll from the file system a listview pops up showing what is in it. Click **Import** and all of the mappings are created.

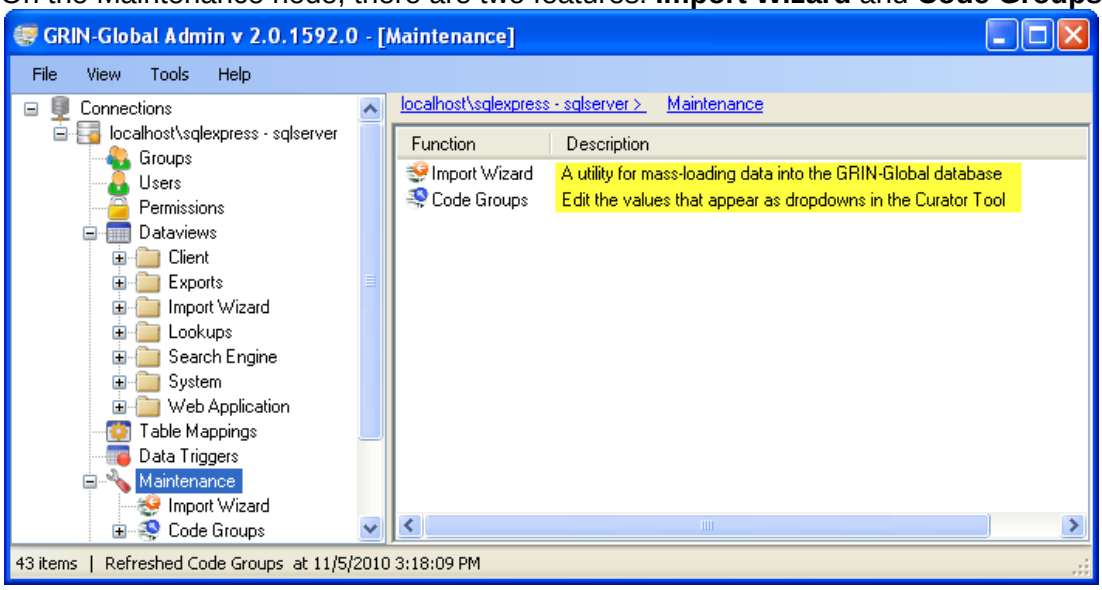


The **Data Trigger** form supports **Title** and **Description**. The **Description** is the long verbiage that is displayed in the **Import** form, telling the Admin Tool user what that particular trigger does.



Maintenance

On the Maintenance node, there are two features: **Import Wizard** and **Code Groups**:



Import Wizard

The **Import Wizard** is a utility for mass-loading data into GRIN-Global; it is explained in detail in the GRIN-Global Guide "Data Preparation Cookbook."

Note

The IW can be helpful creating records in a test system, but at this time, it is not really suited for production work.

Why not (for production)? Primarily for the following reasons:

- Since the Import Wizard (IW) is accessed within the Admin Tool (AT), ***it is only available to the administrator*** - while the data being uploaded is generally controlled by the Curator.

- Another disadvantage with the IW is that if you specify incorrect parent information, it will create that parent record for you. Furthermore, even when parent identifier is spelled correctly, it modifies the parent row and puts a new modified date on the row even if nothing was changed on the parent record.

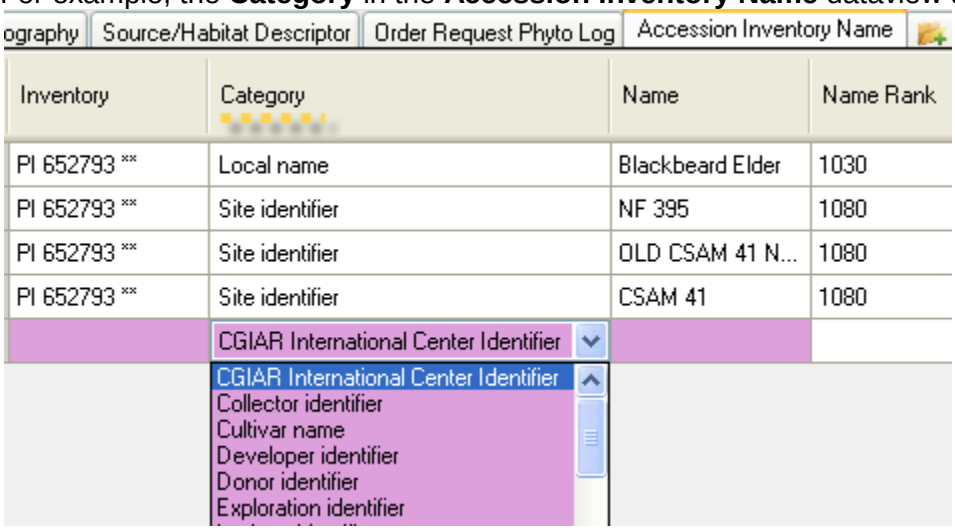
- Also, the Import Wizard code has not been updated – basically, since 2010. Most of its dataviews are old. This means that it matches the 1.0 schema, whereas most organizations will want to test newer versions.

Code Groups

Background Information

Many of the dataviews in the Curator Tool use dropdowns to assist the user in selecting a valid entry – the fields do not allow any random text data to be entered, but instead require a value from a pre-populated set of values. Various codes and data values are stored in the **Code Group** tables.

For example, the **Category** in the **Accession Inventory Name** dataview uses codes:

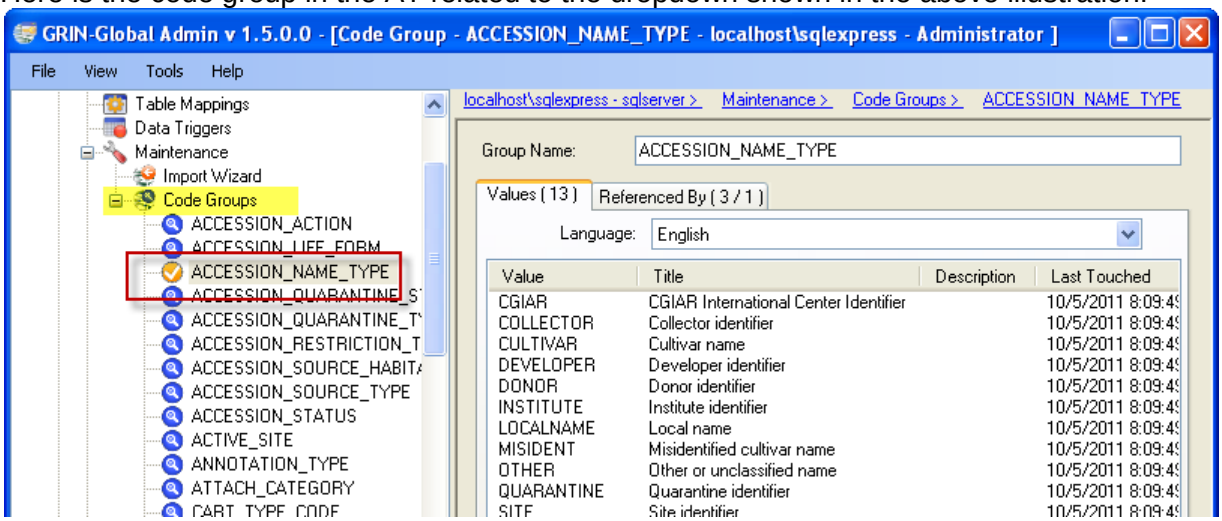


(In the "Database Code Groups & Values GRIN-Global" section of the **Setup Guide Cookbook**, there is a detailed example illustrating how one Code Group, "IMPROVEMENT_LEVEL," is set up in the Admin Tool and then used in the Curator Tool. Directions are also included there for initially bulk-loading **Code Groups** and their **Values**, using the Import Wizard.

Code Groups Editor

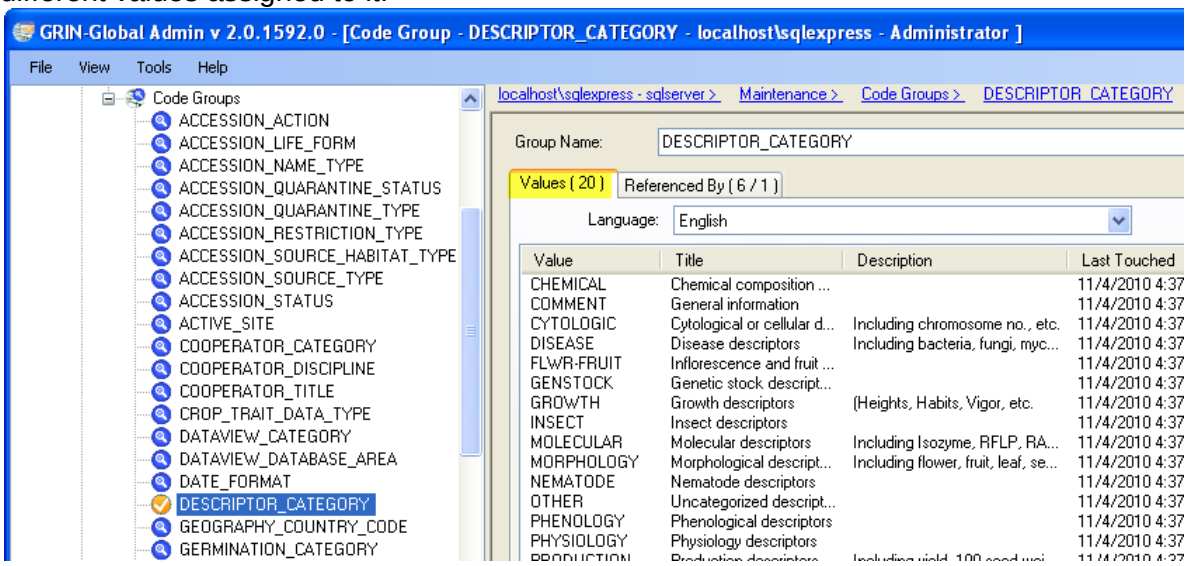
After an organization has initially bulk loaded its Code Groups, it most likely will need to edit, add, or delete certain Code Groups or their Values; the Code Groups editor is the utility for doing this and performing ongoing updates of the Code Groups.

Here is the code group in the AT related to the dropdown shown in the above illustration:

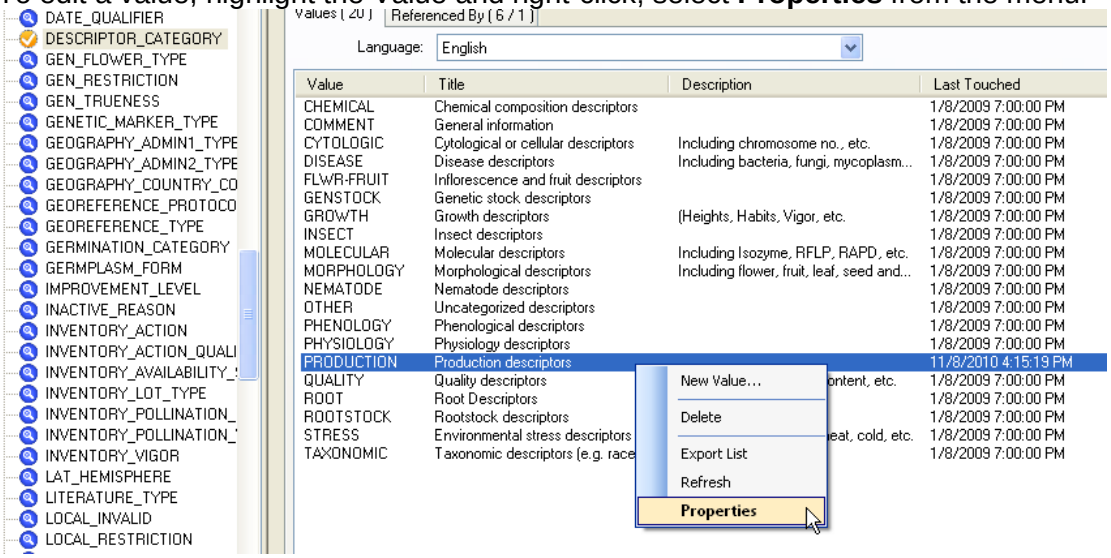


The AT Code Groups Editor does not allow deletion of a code if it is used anywhere in the database. Another advantage to using the Editor pertains to the editing of codes; when modifying codes, all GG database records are also adjusted to correspond with the newly-edited code values. Therefore, only use the AT Code Editor to delete or edit codes.

In this example, you can see that the Code Group **DESCRIPTOR_CATEGORY** has had 20 different values assigned to it:

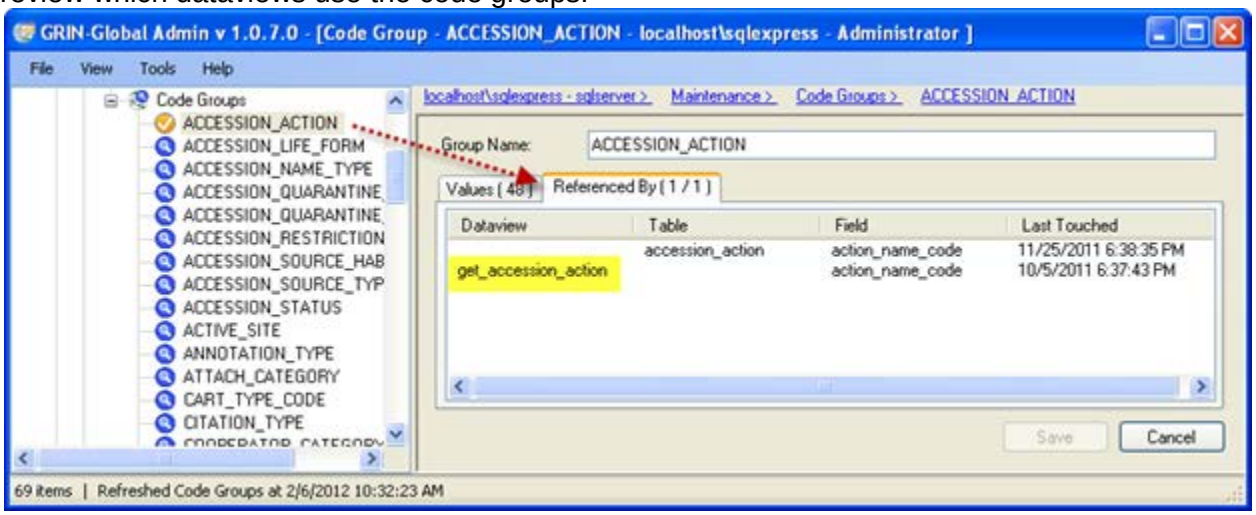


To edit a value, highlight the Value and right-click; select **Properties** from the menu.

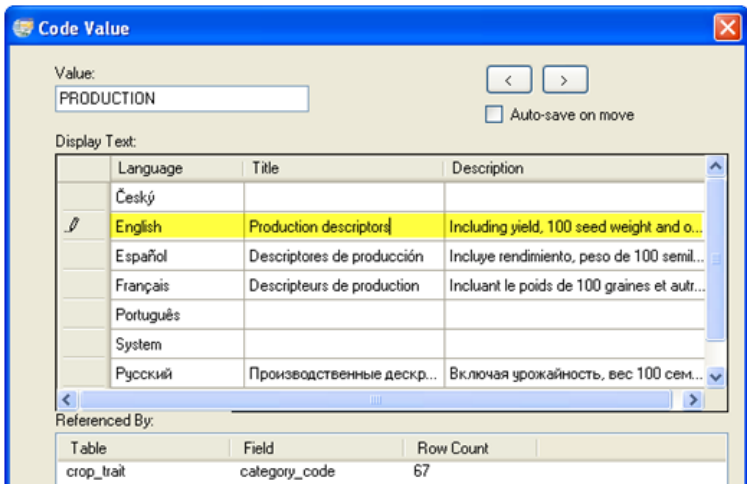


Determining Where the Code Groups are Used

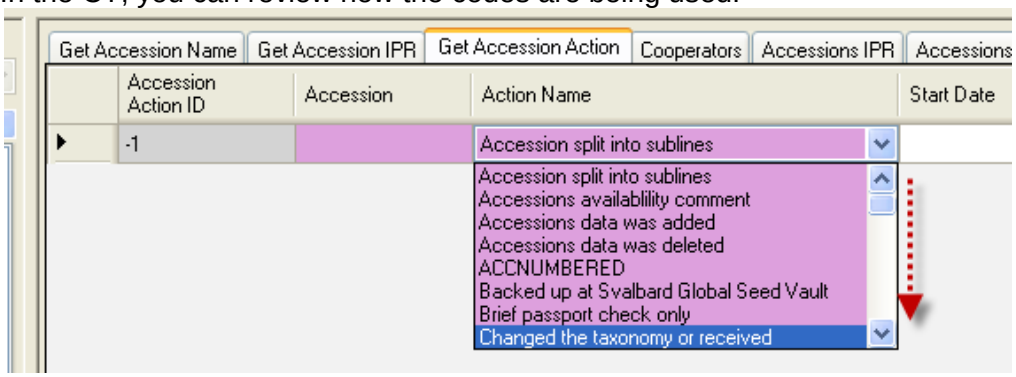
Use the Code Groups tool in the Admin Tool to review the values for each group and also to review which dataviews use the code groups:



In the AT, right-click on a **Code Group** name to edit the code's **Value**, **Title**, and **Description** cells. Use the right and left arrow icons on the Code Value wndow to move through the list of values; click **Save**:



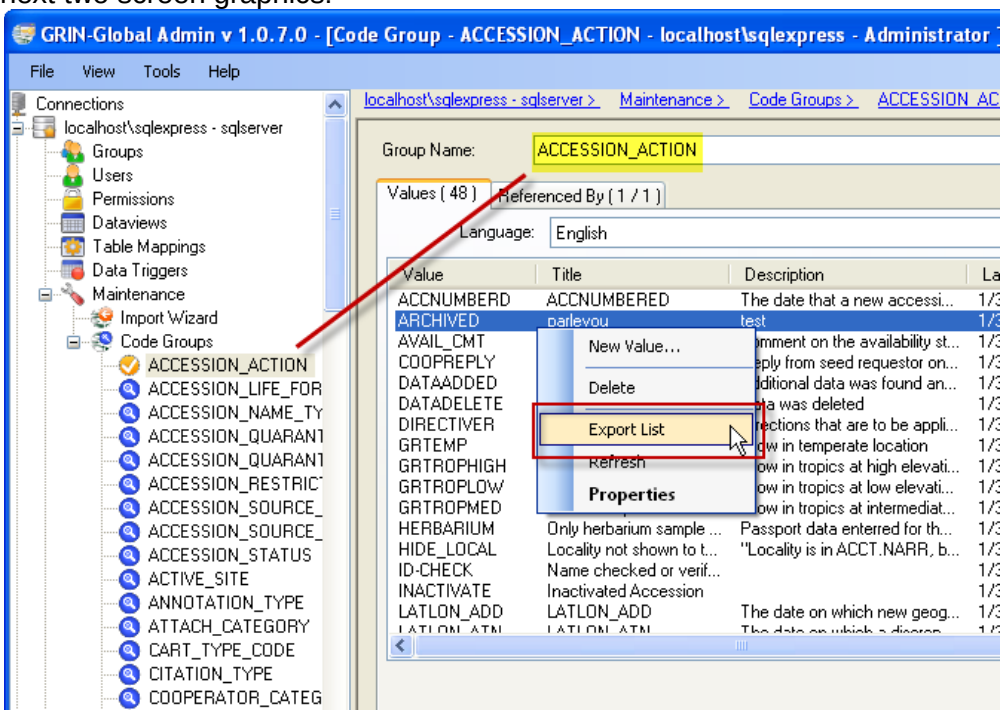
In the CT, you can review how the codes are being used:



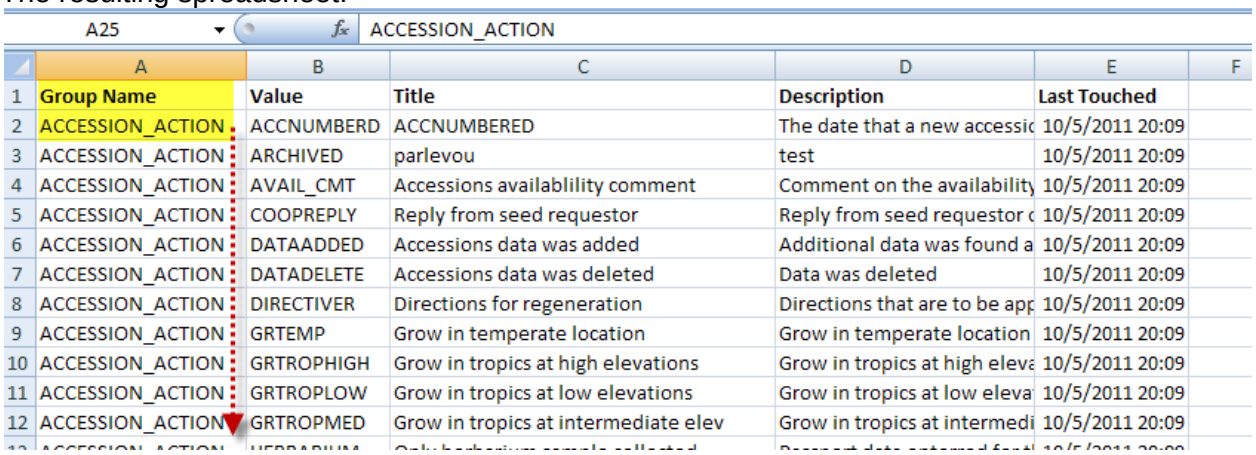
Extracting All Codes for a Specific Group Name

Use the Admin Tool Code Group Editor whenever you want to copy a complete list of code values to a spreadsheet. You can also use this technique to modify many code values at one time and then import them back into GG via the Import Wizard.

For example, **ACCESSION _ACTION**, one of the code groups, has been highlighted. The menu displayed when the user positioned the mouse in the right grid and right-clicked. Selecting the menu option **Export List** produces a comma-delimited spreadsheet (a CSV file). Refer to the next two screen graphics:



The resulting spreadsheet:



A25 | ACCESSION_ACTION

	A	B	C	D	E	F
1	Group Name	Value	Title	Description	Last Touched	
2	ACCESSION_ACTION	ACCNUMBERD	ACCNUMBERED	The date that a new accessic	10/5/2011 20:09	
3	ACCESSION_ACTION	ARCHIVED	parlevou	test	10/5/2011 20:09	
4	ACCESSION_ACTION	AVAIL_CMT	Accessions availablility comment	Comment on the availability	10/5/2011 20:09	
5	ACCESSION_ACTION	COOPREPLY	Reply from seed requestor	Reply from seed requestor c	10/5/2011 20:09	
6	ACCESSION_ACTION	DATAADDED	Accessions data was added	Additional data was found a	10/5/2011 20:09	
7	ACCESSION_ACTION	DATADELETE	Accessions data was deleted	Data was deleted	10/5/2011 20:09	
8	ACCESSION_ACTION	DIRECTIVER	Directions for regeneration	Directions that are to be app	10/5/2011 20:09	
9	ACCESSION_ACTION	GRTEMP	Grow in temperate location	Grow in temperate location	10/5/2011 20:09	
10	ACCESSION_ACTION	GRTROPHIGH	Grow in tropics at high elevations	Grow in tropics at high eleva	10/5/2011 20:09	
11	ACCESSION_ACTION	GRTROPLOW	Grow in tropics at low elevations	Grow in tropics at low eleva	10/5/2011 20:09	
12	ACCESSION_ACTION	GRTROPMED	Grow in tropics at intermediate elev	Grow in tropics at intermedi	10/5/2011 20:09	

Note

In GRIN-Global 1.0, column A with the **Group Name** is not included when the list is initially created. Insert a column before the **Value** column, type the heading **Group Name**, and then enter and duplicate the actual Group Name values. In this example the group name was **ACCESSION_ ACTION**. (The Group Name does is not required to be

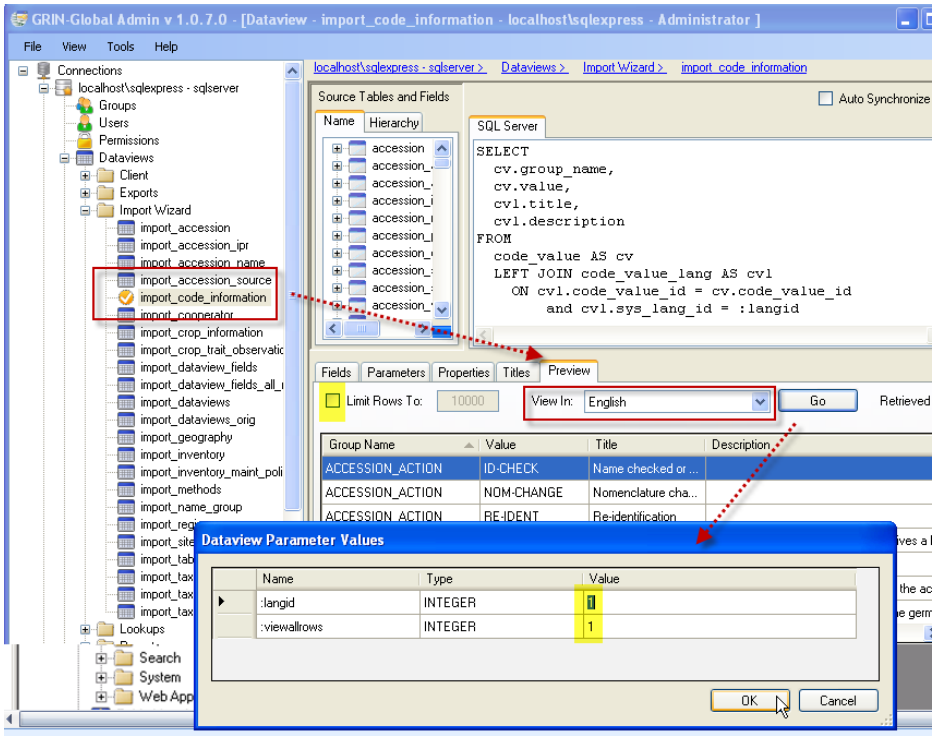
in uppercase, but that is the convention used in GG 1.0.)

If this file is saved as a Unicode text file, it can then be imported into GG via the Import Wizard. (Refer to the **Setup Guide Cookbook** for detailed directions.)

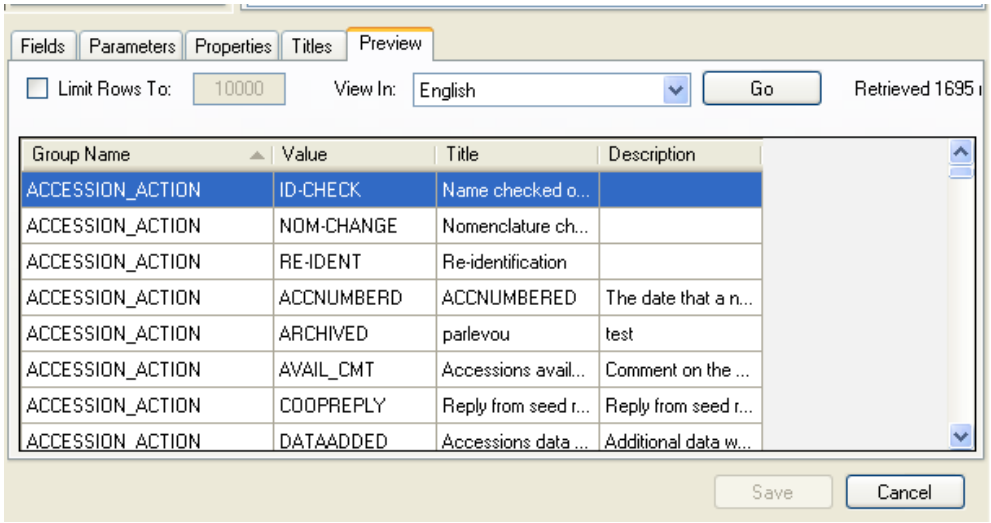
Extracting All of the Code Values

There are two quick ways to produce a spreadsheet that will contain all of the code values for a given language.

In the AT, you can preview the Import Wizard import_code-information dataview in the dataview editor.



Results:



This data can be copied into a spreadsheet. The alternative method for extracting the data is to use the Import Wizard to view the existing data for the language. (This method is fully described in detail the Import Wizard Cookbook.)

The read-only **Code Value** dataview in the Curator Tool can also be used to view the Code Groups and Values – the format for the Code Value field as shown below is **CodeGroup.CodeValue**

Get Sys Dataview Field Lang | Acc.Quar. | Taxonomy Crop Map | Crop Trait Observation | Code Value Language | Method | Accession Inventory Attach | Code Value | Get Code Value Language

Code Value Lang ID	Code Value	Language	Title	Description	Created Date	Created By	Modified Date
2120	ACCESSION_ACTION.ACCNUMBERD	English	ACCNUMBERD	The date that a	12/31/2009 7:00...	SYSTEM, (Defau...	11/5/2010 12:39...
5411	ACCESSION_ACTION.ACCNUMBERD	Español	ACCNUMBERD		9/30/2011 1:57 ...	SYSTEM, Admini...	10/5/2011 7:52 ...
7535	ACCESSION_ACTION.ACCNUMBERD	Français	ACCNUMBERD	Date à laquelle une	9/30/2011 2:03 ...	SYSTEM, Admini...	10/5/2011 7:59 ...
8516	ACCESSION_ACTION.ACCNUMBERD	العربية	المدخل المرقم	تاريخ حصول المدخل	9/30/2011 2:06 ...	SYSTEM, Admini...	10/5/2011 8:09 ...
10172	ACCESSION_ACTION.ACCNUMBERD	Русский	ACCNUMBERD	Дата получения	9/30/2011 2:09 ...	SYSTEM, Admini...	10/5/2011 8:04 ...
2121	ACCESSION_ACTION.ARCHIVED	English	Archived	Archived	12/31/2009 7:00...	SYSTEM, (Defau...	5/14/2012 5:28 ...
2590	ACCESSION_ACTION.ARCHIVED	Français	Archivé	Archivé	12/31/2009 7:00...	SYSTEM, (Defau...	4/10/2012 5:13 ...
5435	ACCESSION_ACTION.ARCHIVED	Español	Prueba		9/30/2011 1:57	SYSTEM, Admini	4/10/2012 5:13

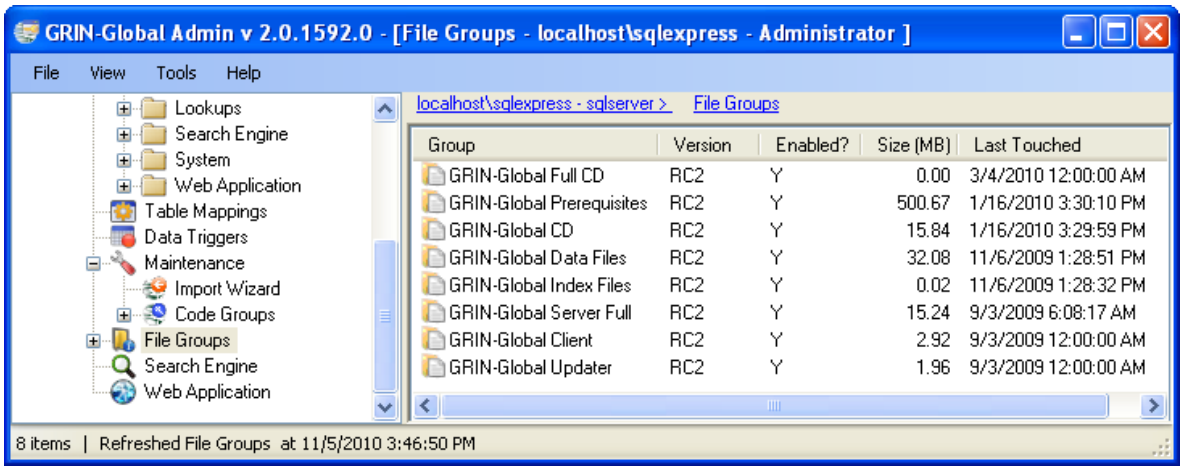
Also, the following SQL may be used in the Public Website Web Query tool to display the codes:

```
USE
  gringlobal;

SELECT
  cv.code_value_id as value_id,

  cv.group_name as group_name,
  cv.value as gg_value,
  cvl.title as cvl_title,
  cvl.description as cvl_description

FROM
  code_value cv

  inner join code_value_lang cvl
              on cv.code_value_id = cvl.code_value_id
   left join sys_lang sl2
                        on cvl.sys_lang_id = sl2.sys_lang_id
WHERE
  sl2.sys_lang_id = 1
/* and   group_name = 'inventory_action'  */
/* use and clause to search for a specific code */

ORDER BY
  group_name,
  gg_value
```

(in this example, sys.lang_id = 1 is for English)

File Groups

Overview
The **File Groups** are pointing to the files that the **Updater** sees. A file group corresponds to exactly one row in either of the Updater's gridviews. The file group can be comprised of one or several files. When a user clicks the "Download/Install" button, Updater will download all file(s) that belong to each group with a checkmark by it. Depending on the kind of file(s) it downloads, it will take the appropriate action (i.e., it will run an .exe, install an .msi, expand a .cab, and display all other files in explorer).



Using the Admin Tool, you can add a new file group, put those files on the server, associate them to that group, and someone pointing to Updater will see it as a new file group.

Note

Currently, the Mirroring functionality will NOT propagate new groups. It will download the files, but not insert the proper records into the database for them to show up as a new file group when others connect to that server. It is intended to do so, but has not been completely developed yet.

Web Application Settings

Note

While most of the **Web Application** settings relate specifically to the Public Website, not all do. For instance, the **DisableSecurity** parameter is a setting that affects the Curator Tool. These parameters are assembled together under **Web Application** because the code for the middle tier and web services are interrelated and stored together.

Public Website
The GRIN-Global Public Website is a web-based application. Using the website features, users can search for accessions by descriptors or by taxonomy and ultimately use the shopping cart function to order accessions. The user documentation for the PW, in a Help file format, can be viewed online.

The basic search feature is the same search as the text box search used by the Curator Tool. Organizations can make their public website visible to the public or only have it available to their internal genebank workers who will use it for searching accessions or taxonomy.

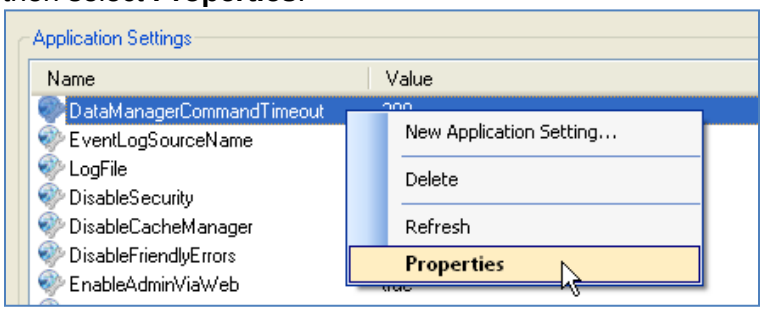
The website uses the familiar shopping cart model for handling orders. Users can easily add items to their cart directly from the search results or from the **Accession Detail** page. Users will be able to jump back and forth between the **Accession Detail** or **Taxonomy Overview** pages for any item in the cart.

Preventative measures will keep "unavailable" items from being added to the cart.

Parameters and Properties
The web application uses certain global settings that are described in detail, beginning on the next page. Note that the administrator may only access these settings from a local machine.

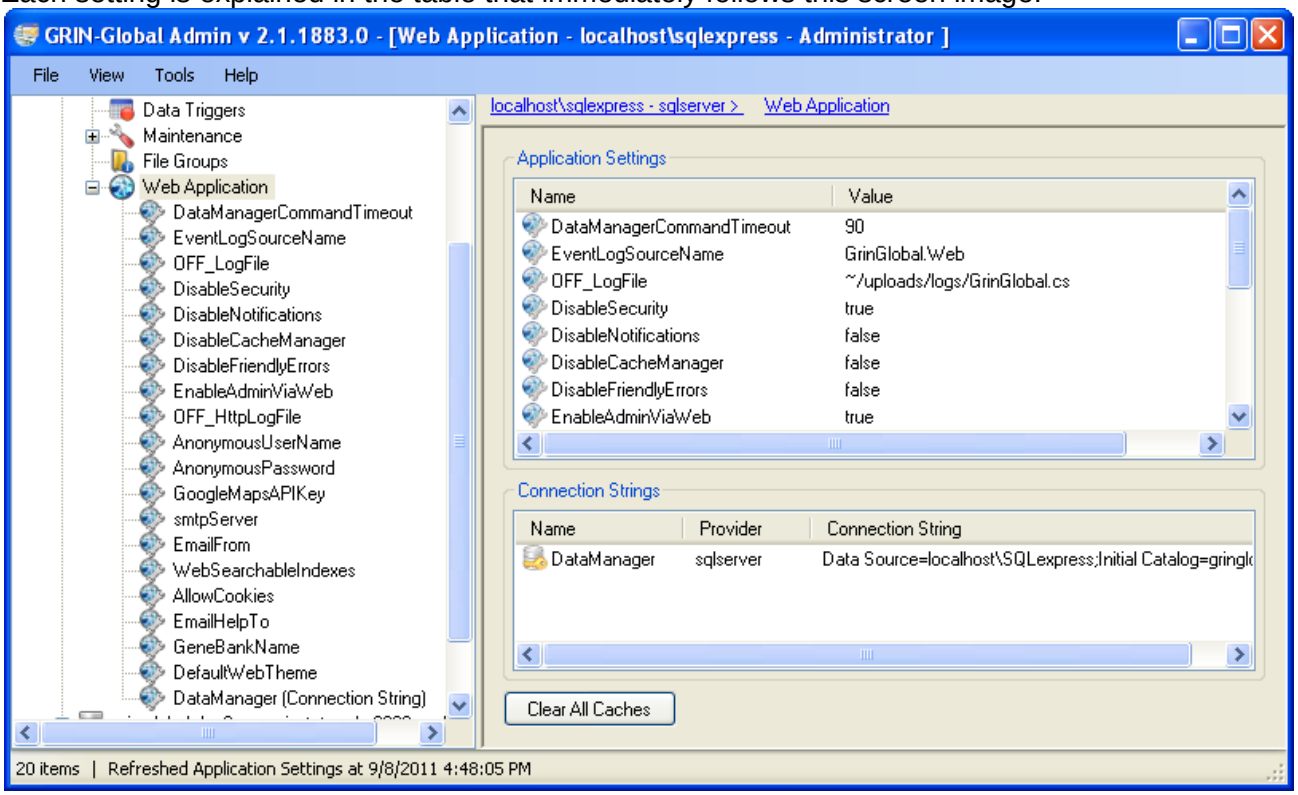
When the web application is installed, most of the web application settings and connection strings are set as needed. Most of these settings will never need to be modified; however they can easily be changed if necessary. There are some settings that do require editing; these are listed in bold in a table on the following page.

Each application setting can be reviewed or edited by right-clicking on the parameter name; then select **Properties**.



Web Application Settings

Each setting is explained in the table that immediately follows this screen image.



Note

Bolded parameters should be reviewed and modified to meet the organization's needs; it is recommended that those parameters not bolded should retain their default values.

Parameter	Description
DataManagerCommandTimeout	Maximum time (in seconds) to wait for a SQL command to execute before throwing an exception error Example: **90**
EventLogSourceName	Category name of all entries written to the Windows Application event log from the web application Example: **GrinGlobal Web**
Off_LogFile	Name of the file in which additional error information is logged. Should point to a file under ~/uploads/ path so that the web app has security rights to write to the file. Example: **~/uploads/logs/GrinGlobal.cs**
DisableSecurity	Either "true" or "false." With this setting, the administrator can globally enable or disable security in the *Curator Tool*. To disable global security, set to "True." When this parameter is "true," permissions are completely ignored. **False** is the default.
DisableNotifications	Inactive -- was planned to be used to disable sending queued notification email for the feedback program, but it is not implemented. **False** is the default.

DisableCacheManager	For troubleshooting purposes, all caching mechanisms can be shut off here. Valid values are "true" or "false." **False** is the default.
DisableFriendlyErrors	For troubleshooting purposes, the “nice” error page can be disabled so the raw error information is displayed. Valid values are "true" or "false." **False** is the default.
EnableAdminViaWeb	If "false," Admin pages on website will not function properly. Valid values are either "true" or "false." **True** is the default.
Off_HttpLogFile	Inactive - not used.
AnonymousUserName	This is the GG UserID that the web application uses as a login when the visitor has not yet logged in under his own credentials. Example: **guest**
AnonymousPassword	Password for the above Example: ???
GoogleMapsAPIKey	The Google Maps API lets you embed Google Maps in your own web pages. A passphrase is required by Google Maps for the mapping portion of the GRIN-Global web application to work properly. For localhost, this is already provided; for your GG server, you will need to access Google for the key. (See the **GoogleMapsAPIKey** section immediately following this table for detail.)
smtp Server	Address of the mail server sending messages. Default: **your-email-server-name**
EmailFrom	The “from” email address produced when the Public Website sends email messages. Default: **your-email-from-address**
WebSearchableIndexes	[not functional – needs to be removed from the AT] The search engine uses the records in the system sys_search_autofield table to find the results when doing a search on the public web site’s plain accession search and the in the Search Tool’s text box search. (Use the get_search_autofield dataview to display the records; records can be added to the sys_search_autofield table using SSMS to broaden the searches.)
AllowCookies	Valid values are either "**true**" or "**false**." If true, the application will use a computer cookie on the PC to store small data such as the language preference, etc. Deafault: **true**
EmailHelpTo	The email address that will receive the message when the user completes the Public Website’s “Contact Us” form. Example: **grin-global.feedback@ars-grin.gov**
GeneBankName	Indicates what text will be displayed in the Public Website’s header area Example: **GRIN-Global Release 1.7**
DefaultWebTheme	Designates which stylesheet to be used for the Public Website. The style sheets are stored in **C:\Inetpub\wwwroot\gringlobal\styles**

	Example: **theme1**
SafeImagePath	Example: <add key="SafeImagePath" value="http://www.ars-grin.gov;http://www.grin.gov"/>, If image has http path, only these are accepted for display ("approved image servers" – in addition to the GG server. (If a curator entered a URL such as http://bad-place.com/evil-file, the PW pages wouldn't display the link because it wasn't to a server on the SafeImagePath. Example: http://www.ars-grin.gov/;http:www.grin.gov
UnSafeFileExtension	Example: <add key="UnSafeFileExtension" value="exe;zip;asp;aspx;js;htm;html;shtml"/> Any file with these extensions will not be allowed to be uploaded by users to the server. Example: exe.zip;asp;aspx;js;htm;html;shtml
DefaultDownloadURL	Example: <add key="DefaultDownloadURL" value="http://distribution.grin-global.org/gringlobal/gui.asmx"/>: For a new machine, the first time the GG Updater is run, this will be the location where the Updater will look for the server elements. Example: http://distribution.grin-lobal.org/gringlobal/gui.asmx
WebHelpURL	Example: http://www.ars-grin.gov/npgs/gringlobal/HTML5/
WikiURL	Example: http:// www.ars-grin.gov/ index.php/Main_Page
EmailOrderTo	Example: <add key="EmailOrderTo" value="laura.gu@ars.usda.gov;admin@something.com"/> If genebank has one specific email to address, here is where to enter, so all on-line GG order emails are sent to this address, rather than any other curator email address.
HomePage	Example: http://www.ars-grin.gov/npgs/index.html
MakeReportsExternal	Default: false "False" requires users to login in order to see the Reports menu option. When set to "True," all users see reports even without logging in.

Parameter	Description	Example
SysChangePasswordRequiresSSL	Is an SSL connection required for the ChangePassword method. Recommended to set to "true" since the password is sent as plain text only during the password change.	FALSE
SysUserPasswordMinLength	The minimum length accepted for a new password.	12

Parameter	Description	Example
SysUserPasswordMaxLength	The maximum length accepted for a new password. (Note: ridiculously long passwords could take excessive server time to process.)	255
SysUserPasswordMaximumAge	The number of days a password is good for before a change is required. A setting of 0 means the passwords never expire.	60
SysUserPasswordMinimumAge	The minimum number of days before a password can be changed again.	1
SysUserPasswordWarning	The number of days before password expiration that the user gets a waning on login that the password will expire soon.	14
SysUserPasswordExpireLock	The grace period after password expiration that the user can still change the password themselves. After this an administrator is required to change the password. A setting of 0 means no grace period, a setting of -1 means infinite grace period to change the password.	30
SysUserPasswordMinDiffs	The minimum number of characters it is necessary to change for a new password.	3
SysUserPasswordHistory	The number of old passwords kept that can't be duplicated when making a new password.	24
SysUserPasswordReqDigit	The regex pattern for determining a digit character	\p{Nd}
SysUserPasswordReqPatt1Ccnt	The minimum number of digit characters that must be included in a new password.	1
SysUserPasswordReqLower	The regex pattern for determining a lower case character	\p{Ll}
SysUserPasswordReqPatt2Ccnt	The minimum number of lower case characters that must be included in a new password.	1
SysUserPasswordReqUpper	The regex pattern for determining an upper case character	\p{Lu}
SysUserPasswordReqPatt3Ccnt	The minimum number of upper case characters that must be included in a new password.	1
SysUserPasswordReqSpecial	The regex pattern for determining a special character	[\p{S}\p{P}\p{Z}\p{C}]
SysUserPasswordReqPatt4Ccnt	The minimum number of special characters that must be included in a new password.	1
SysUserPasswordSaltBytes	The number of random bytes added as a salt to the hash of the stored password.	6
SysUserMaxLoginFailures	The number of bad login attempts allowed for an account.	5
SysUserLoginFailPeriod	The number of seconds bad passwords attempts for an account are remembered.	900
SysUserLoginLockPeriod	The number of seconds an account is temporarily locked after exceeding the allowed login failures.	900

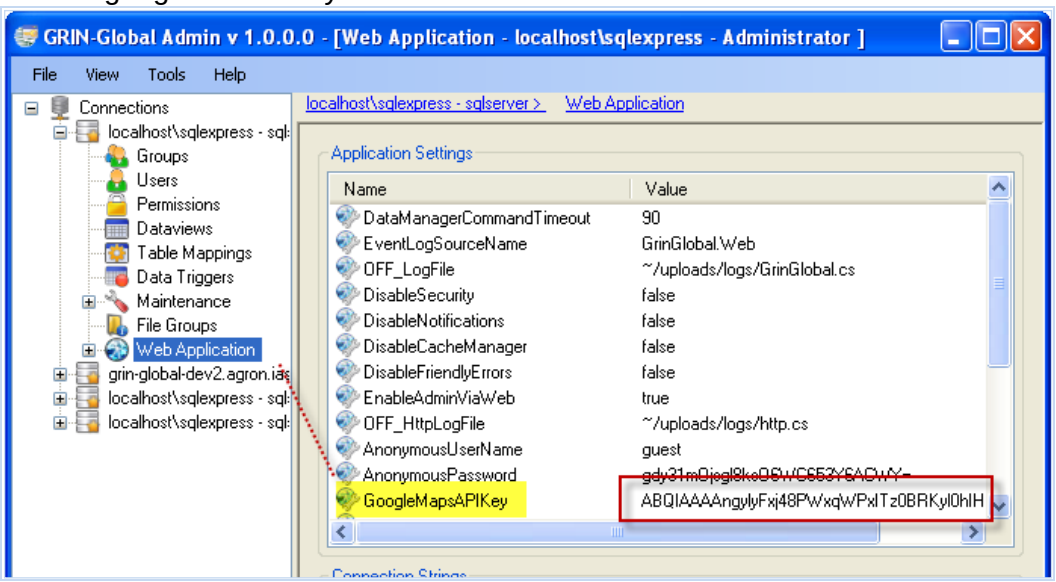
Parameter	Description	Example
SearchEngineFormattedTextPattern	A regular expression controlling the search engine's detection of formatted QBE queries like @crop.name = 'WHEAT'. Do not alter unless absolutely necessary. Currently customized to allow floating point numbers overlooked by the current SE internal pattern. ^\s*@\w+\.\w+\s*(?:<>|<=|>=|!=|=|<|>)\s*(?:[-+]?[0-9]*\.?[0-9]+|'([^\f\n\r\v']|'')+')\s*|^\s*@\w+\.\w+\s+(?:not\s+|NOT\s+)*(?:in|IN)\s*\((?:\s*(?:\d+|'([^\f\n\r\v']|'')+')\s*[,|\)])*|^\s*@\w+\.\w+\s+(?:not\s+|NOT\s+)*(?:like|LIKE)\s+'([^\f\n\r\v']|'')+'\s*|^\s*@\w+\.\w+\s+(?:is|IS)\s+(?:not\s+|NOT\s+)*(?:null|NULL)\s*	

GoogleMapsAPIKey

To obtain the Google Maps passphrase, in your browser, go to this site: http://code.google.com/apis/maps/signup.html. (Login to your google account. If you do not have a Google account, you must create one.)

Input your GG server site's address – a passphrase key will be generated. (For example, for the DBMU test site, we used http://test.grin-global.org as the site for generating the key.)

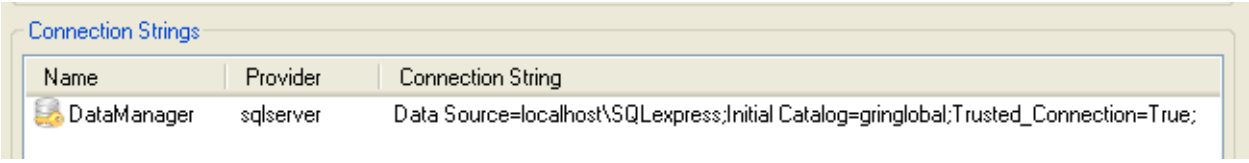
In the AT, under the **Web Application** node, select the **GoogleMapsAPIKey** sub node. Copy the Google-generated key value into the **Value**: box:



Connection Strings

This specifies how the web application is to connect to the database engine.

Parameter	Description
DataManager	The name required by the GRIN-Global web application so it can connect to the GG database. Can have a Provider value of "sqlserver," "mysql," "postgresql," or "oracle." Connection String value varies from one provider to the next.



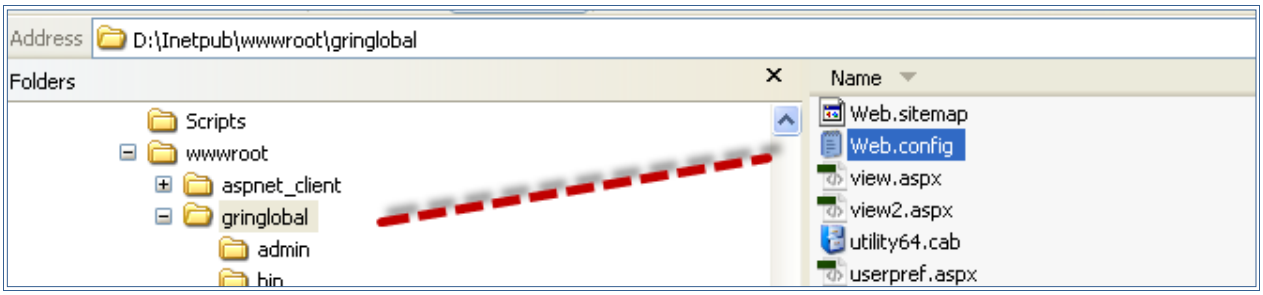
Other Website Configuration Settings
Besides the parameters listed under the **Web Application** node in the Admin Tool, there are additional items that can be modified to affect a customized version of the Public Website. Each organization may wish to include its own banner and graphical appearance to the site; there may be some additional titles or admin values for state or country that they may want to include.

Tools Menu Option on the Public Website

The **Tools** option on the Main Menu on the Public Website makes it possible for a user to submit Read-Only SQL queries. This option is not visible by default.

There are two approaches an Admin can take to make this visible. In both alternatives the user must be logged in and that the Admin has linked the PW account with the user's Curator Tool account. (See **Users | General | Web Login**.)

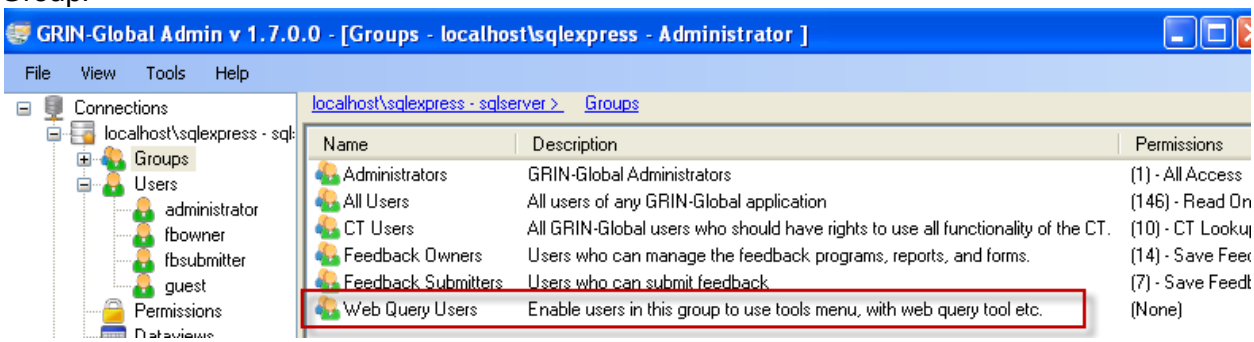
In the first approach, the key is to edit the web.config file which comes on a new install. This file is located on the server.



There is a section in the web.config pertinent to this discussion. The default is:

```
<location path="Admin">
  <system.web>
    <authorization>
      <allow roles="webtools" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

When the default is left this way, the GG Admin must add the user to the Web Query Users Group.



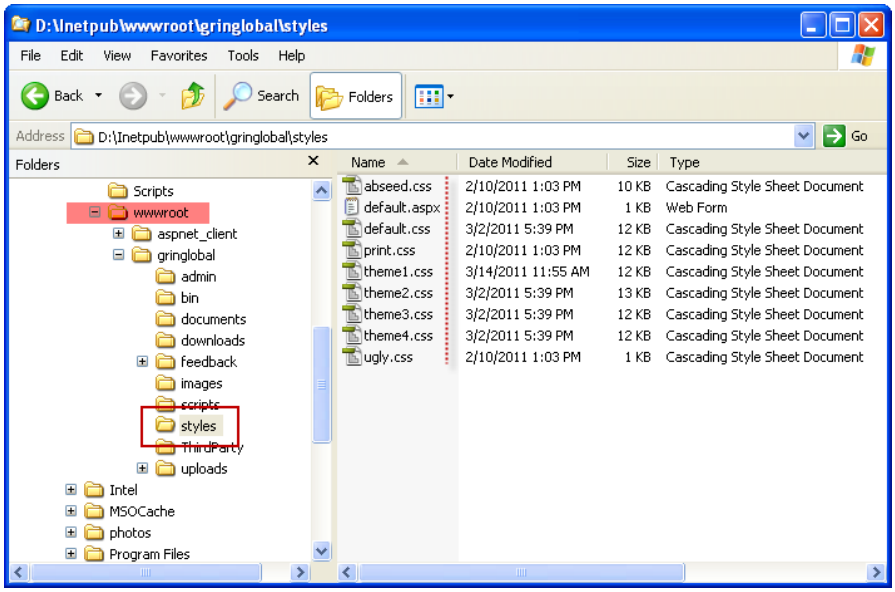
The second alternative is to edit the **web.config** file, and change one line so that the allusers group gets authorization to access **Tools**. (Every Curator Tool user is automatically in the allusers group when the user's ID is created in the Admin Tool.)

This avoids the extra step for the administrator of adding each user to the Web Query Users Group – all users will have access to Tools when the web.config file is edited as shown below in the red line:

```
<location path="Admin">
  <system.web>
    <authorization>
      <allow roles="allusers" />
      <deny users="*" />
    </authorization>
  </system.web>
</location>
```

Graphical Appearance

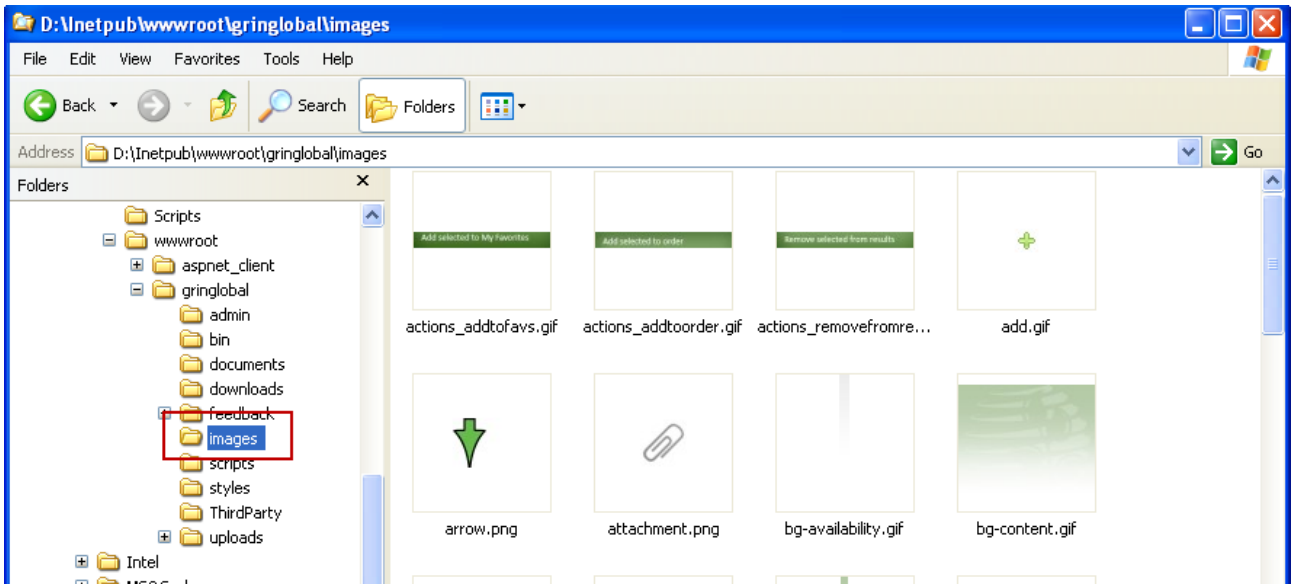
The Public Website uses CSS style sheets to control its appearance; the CSS files are stored in the **styles** folder under the **wwwroot** folder.



Shown above are several style sheets that were used in testing; by default **theme1** will be selected in the AT's Web Application Settings (discussed earlier). As with any CSS stylesheet, if desired, the individual styles can be modified.

Images

The style sheets use the graphic files stored in the **images** folder. If you substitute your own images, be sure that the image's dimensions and their filenames match the originals; otherwise your website may be distorted.



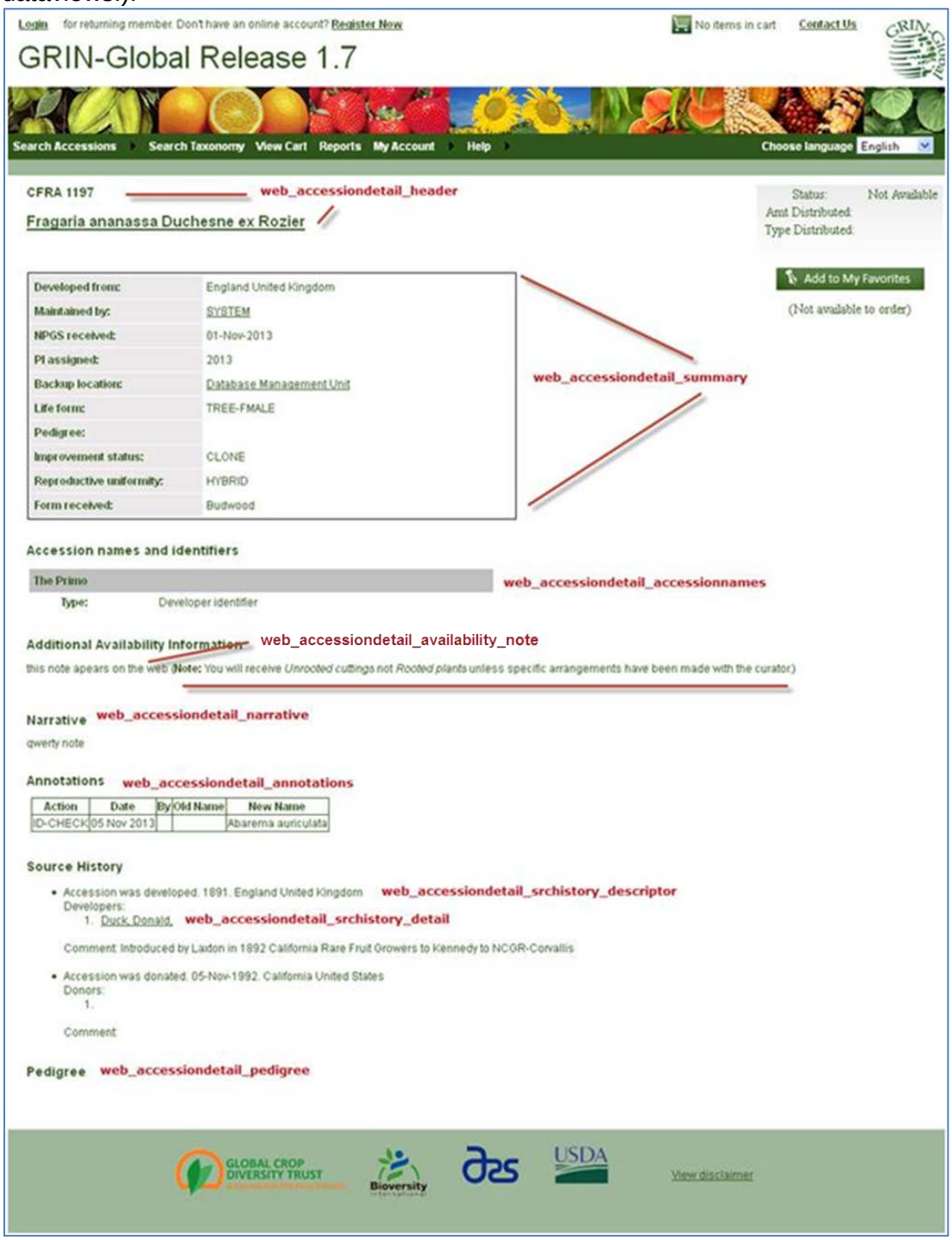
Titles, States, and Country Settings

On the **Registration** page, there are dropdowns that rely on data stored in GG tables. To review or edit, use the corresponding dataview::

- titles – **COOPERATOR_TITLE** values in the **get_code_values** dataview
- states and countries – **get_geography** dataview

Accession Details Page

The Accession Details page displays various accession-related fields. The following image illustrates the basic mapping of the page. Use the AT Dataview Editor to review the specific dataviews.):



Field / Text	Source / Note
Annotations	**accession_inventory_annotation** dataview. Fields:
Citations	From the related **Citation** for the accession. Fields: **Author(s) name**; **Citation Year**; **Citation Title**; **Literature Source**; **Reference**
Intellectual Property Rights	**accession_ipr** and **accession_ipr_citation** dataviews. Fields: **IPR Type**, **IPR Number**, **IPR Crop Name**, **Issued Date** and related **accession_ipr_citation** data
Narrative	a**ccession** dataview. **Note** field.
Other Information about accession	**accession_inventory_attach** dataview. Fields: **Title, Cooperator, and Created Date** field when the attachment record has a category “URL Link”
Pedigree	**accession_pedigree** dataview. Field: **Pedigree Description**
Source History	**accession_source** and the **accession_source_cooperator, and the source_descriptor_observation** dataviews
Vouchers for accession	**accession_voucher** dataview. Fields: **Collector Voucher Number**, **Voucher Cooperator**
Web Availability Note	Can come from the **Inventory Maintenance Policy** (Web Availability Note) or directly from the Inventory

Database Naming Conventions

The database names, which can be viewed in SQL Server Management Studio, follow some general guidelines. Table names and field names may be visible to the end user, but since foreign keys, constraints, index names, etc are not displayed in GRIN-Global, there may at times be some divergence in the naming conventions.

- Lowercase was used for all object names
- Split words use underscores where appropriate
- Fields with a data type of date typically end with "**_date**."
- Key fields (primary and foreign) – always end with
  - **_id** if they are an integer type
  - **_code** if they are a string type
- Primary key fields always start with tablename, such as inventory_id, code_group_code, accession_id, etc.
- Comment / notation fields are typically named "note" (not narrative or comment or something else)
- "System" tables begin with "**sec**_." These contain data not related to actual GRIN-Global data, but are needed for the system to function properly.
- Maximum length for any database object name is 30 characters (for compatibility with Oracle)
- Language-specific tables end with "**_lang**."
- All permission-related tables have "_perm" in their names. Similar pattern for dataviews (**_dataview**), tables (**_table**), users (**_user**), fields (**_field**), taxonomies (**taxonomy**_), etc.

Typically, the developers used the first letter of each word in the table name. For example:

- cooperator -> c
- accession -> a
- sec_lang -> sl
- inventory_action -> ia
- order_request_item -> ori
- app_user_item_list -> auil

Oracle and PostgreSQL have additional objects they use:

- Triggers start with "tg_"
- Sequences start with "sq_"

Also, there are some exceptions because there are sometime several tables beginning with the same letter, so sometimes a second letter was used. For example, since cooperator was already using "'c," "cg" was used for cooperator_group, "ci" for citation, and "cr" for crop.

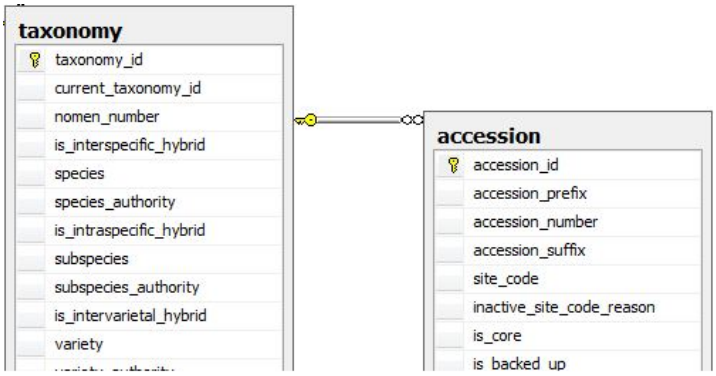
The name "fk_c_cur_c" means "foreign key from cooperator table to cooperator table via the current_cooperator_id field". "fk_c_modified" is the foreign key from cooperator to cooperator on the created_by field" The 2nd "c" was omitted because it was so common.

Some of database names do not follow that convention because they were created using a tool, such as SQL Server Management Studio. It is much better to use the Diagrams functionality in SQL Server than to look at the table tree when trying to view the schema.

In SQL Server Management Studio, click on **Database Diagrams.**

First time clicked it will prompt with “This database does not have one or more of the support objects required to use database diagramming. Do you wish to create them?” Answer “yes.” Then, right-click “Database Diagrams” again and choose “New Database Diagram.” A form displays that lets you select one or more tables to add to the diagram. Those tables are added and relationships are automatically displayed for related tables.

The Diagrams functionality auto-draws relationships for you in an intuitive manner. In the figure below, you can see there is a relationship from the accession table to the taxonomy table without knowing the name of the relationship:



Foreign Keys

Whenever possible, foreign key fields always start with the parent table name. Examples:

- inventory.accession_id
- accession.taxonomy_id

Exceptions are order_request.source_cooperator_id, order_request.requestor_cooperator_id, etc.

Audit fields must always be named exactly what they are named for default table mapping settings to work properly: created_by, created_date, modified_by, modified_date, owned_by, owned_date

If a ”normal” field name would have been a reserved SQL keyword, the tablename was prepended. For example, “number"' is a reserved keyword in Oracle, so it was instead named “accession_number.” Other fields in the same table may have their names prepended for consistency within that table. For example, “suffix” is not a reserved word, but since “prefix” and “number” fields are (in MySQL and Oracle, respectively) the developers prepended suffix with “accession_,” resulting with “accession_suffix” as the field name.

The developers avoided the use of abbreviations whenever possible. (it is easier to google a word than an industry-specific abbreviation.) One exception was “accession_ipr “ ("intellectual property rights"). Exceptions typically occur only when a good alternative short name could not be found.

Other Administrator Tasks

Besides understanding how to use the Admin Tool, which this document primarily covers, a GRIN-Global administrator needs to know other miscellaneous items. For example, the Curator Tool interface can be customized for certain features by adjusting the .txt files that are installed during the CT installation. Refer to the *User Files* section for details.

Background Information on Components, Design, and Code Samples

Source Code

Not that everyone needs to do this, but if you were to look at the source code, you would also find other code. The following components support, in one way or another, the installation or execution of the above applications:

- **GrinGlobal.Business**
  "Real" middle tier code. Web service is just a pass-through to this assembly.
- **GrinGlobal.Core**
  Common code used by basically every project.
- **GrinGlobal.DatabaseCopier**
  GUI for creating and loading database that supports all four database engines (SQL Server, MySQL, Oracle, PostgreSQL).
- **GrinGlobal.DatabaseInspector**
  Code used by DatabaseCopier project. This is where "real" code is for creating databases.
- **GrinGlobal.Interface**
  Declarative interface code for supporting third-party development of extensible portions of Grin-Global apps, such as Data Triggers for the middle tier.
- **GrinGlobal.UAC**
  Command line tool that causes a UAC prompt to appear in Vista and later for executing code in elevated context. Used by installers in Vista and later.
- **GrinGlobal.Web.Installer**
  Assembly for running install-time code for web site, as a web project does not allow calling into an Installer class.
- **GrinGlobal.*.Setup**
  Setup projects for packaging apps for installation on client servers and machines.
- **GrinGlobal.*.Prerequisites**
  Setup projects for packaging prerequisites of GRIN-Global apps (SQL Server, Powershell, .NET 3.5 SP1, Crystal Reports, etc.).

GRIN-Global Design Overview

The design of the GRIN-Global Application Suite software has the following primary goals:

- Reliability
- Flexibility
- Extensibility
- Granularity
- Customizability

To that end, the following approaches have been used:

- No assumptions of the client are made
    - Client controls how errors are reported from the middle tier -- an exception can be thrown or a return value can be checked
    - All communication is xml over http, meaning any client that can talk http can talk to GRIN-Global
    - Data scrubbing, auditing, and security checks are performed server-side to help prevent malicious tampering of data
- Custom business logic can be created without touching main code base
    - GrinGlobal.Interface.dll contains interfaces and classes that can be implemented or extended to tap into the GRIN-Global infrastructure
    - Custom filters can be added to perform additional tasks when data is read, updated, inserted, or deleted
- Dataviews can be changed or created with no code changes
    - Dataviews that ship with GRIN-Global can be tweaked.
    - Custom dataviews can be created via the DataView admin page. Basic SQL knowledge is required, but no coding is involved.
    - New tables can be added to the database schema itself, the tables mapped using the Admin Tool, then dataviews created to pull from the new tables -- all with no code changes.
- Security is highly granular and configurable
    - Security can be applied to a specific dataview, table, or globally
    - Security can be restricted to specific rows within a table (e.g. “updatable if the record is owned by John Q. Public”)
    - Security can be completely turned off to make administration easier
    - Administrative functions can be restricted to be accessed only from the local machine, or opened up to be accessible via a web service

- A database is a reliable data store -- nothing more, nothing less.
    - The database schema for GRIN-Global contains tables, indexes, and constraints -- no triggers, no views, no stored procedures, no user defined functions, etc
    - Foreign key constraints are required and enforced -- data reliability is more important than performance
    - Alternate key constraints are defined where appropriate
    - Transactions are used when appropriate

Tiers
The GRIN-Global Application Suite consists of three major tiers:

- Client Tier – Curator tool, Search tool, Admin tool, Updater, Web site
- Middle Tier – Web service
- Data Tier – SQL Server, MySQL, Oracle, PostgreSQL

Client Tier
All user interaction takes place in this tier. No code in the Client Tier ever talks directly to the Data Tier. It talks exclusively to Client Tier and Middle Tier code. This distinct separation of duties allows the overall design to be extremely flexible and change over time to fit each genebank's needs without affecting other genebanks.

Middle Tier
The “glue” for GRIN-Global. All business logic and data scrubbing occurs in this tier. It has a coarse API, allowing it to grow gracefully as new features are added over time.

Data Tier
This tier is essentially the database itself. There should be no reliance upon triggers or stored procedures to perform data validation or scrubbing. However, it should perform data integrity checks (foreign key constraints, unique constraints, etc.)

Reports
There are two kinds of reports that GRIN-Global handles:

- reports embedded in the Curator Tool that have been designed in Crystal Reports
- reports that run on the Public Website

Reports in the Curator Tool: General Steps in Producing Reports
The reports bundled with the Curator Tool were designed in Crystal Reports. The basic steps require that a Curator Tool dataview be designed first that obtains the data that will ultimately be reported in the report. Then a Crystal Reports report is designed. However, since the Crystal Reports tool doesn’t work directly from the GRIN-Global table and fields, the data must be dragged from the GG dataview into a spreadsheet from which the Crystal Reports report is designed.

For example, the **Order Packing** dataview was created and then the **Order Packing by Accession** report was designed using the fields displayed by the dataview.

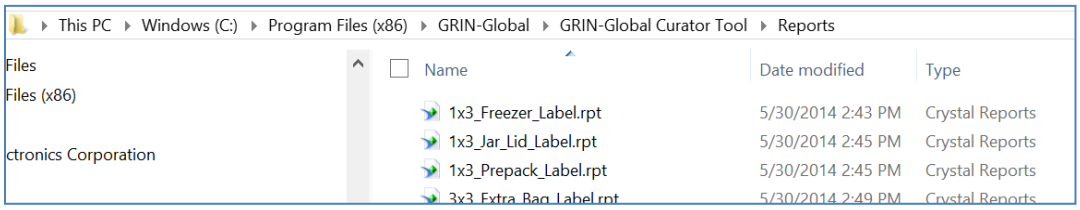
The main steps are:

- Open the Curator Tool; bring up sample data in the dataview on which the report is to be based
- Select all data by double clicking upper left corner
- CTRL drag data from CT dataview to an Excel spreadsheet (use CTRL to get schema field names instead of friendly names)
- In the spreadsheet, delete sheet2 and sheet3
- Save spreadsheet as type **Excel 97-2003 Workbook (*.xls)**
- Open Crystal Reports
- Select **Report Wizard**
- Click **Create New Connection**
- Click **Access/Excel (DAO)**
- Choose **Database Type: Excel 8.0**
- For Database Name: use ... to browse to saved xls file
- Finish
- Proceed with creating the report by selecting fields from sheet1 and designing the report (the Crystal Reports details of report design are omitted here)
- Save the report file and place in the Curator Tool **Reports** directory

Location of the Crystal Report (.rpt) Files

The .rpt files must be stored in a specific folder in order for the CT to use them. Typically the Curator Tool user will have these files loaded in this folder:

Operating System	Folder Containing the ReportsMapping.txt File
Windows 7.1; 8; 8.1	C:\Program Files (x86)\GRIN-Global\GRIN-Global Curator Tool\Reports



Public Website Reports
The Public Website has a menu option for Reports. Reports can be added as they are developed to meet an organization's needs.

GRIN-Global Technical Overview

GRIN-Global is a powerful, but easy-to-use, and very flexible, plant genetic resource (PGR) information management system. It is designed to meet the needs of organizations of various sizes. The system uses a.NET ("dot net") framework and Visual Studio development environment. This technology allows data to be stored locally, or on networks. Centralized data can be distributed to other off-site systems.

In the simplest scenario, a genebank may use GRIN-Global on one computer for a single curator. The entire system will run on that one computer; a network is not required in this scenario. The PC will house the database, a middle management layer, and the client interface. This configuration allows a small genebank to manage their genetic resources. Optionally, if networked later, the data can be exported to others for inclusion in their data systems.

In another scenario, the database and the middle management layer reside on a separate computer (server). Clients from multiple curators can access this centralized database from multiple locations, using the GRIN-Global Curator Tool. This installation could export data to other data gatherers such as the Global Biodiversity Information Facility (GBIF).

The connections between the database, the middle layer, and the client are through web services. (A web service is defined as "a software system designed to support interoperable machine-to-machine interaction over a network.")

GRIN-Global can also be used to access one database on the user's local PC, and at other times data stored on another networked computer. The remote, networked computer may be an

organization's database server or a colleague's computer to which the user has been granted security privileges for sharing data.

Design of the GRIN-Global System

GRIN-Global is built using a three-tier (Presentation, Business, and Data) architecture design.

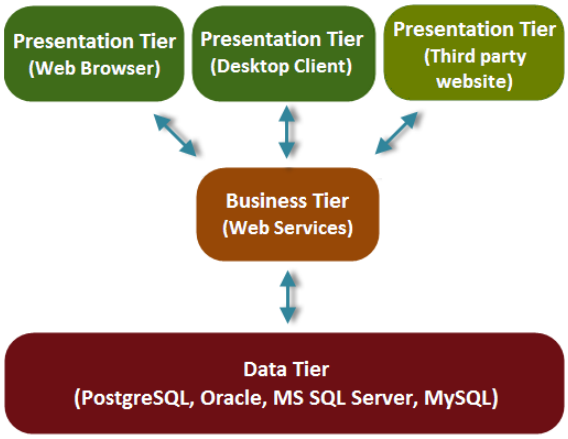
Presentation The first tier is the one with which users interact. This tier can assume different forms. For example, the GRIN-Global Curator Tool, is a .net ("dot net") application, designed to run on a PC. The Curator Tool connects to the web services applications, the middle tier, and provides the interface in which the user ultimately searches, reads, and updates the data in the database, in the third tier.

Instead of using the Curator Tool on your computer, as an alternative you can review the plant genetic resource data via the GRIN Public Website. The web pages there read/retrieve PGR data from the GRIN-Global database using the middle layer business tier's web services.

Another alternative for accessing GRIN-Global data is to use a third-party application, such as MaizeGDB or SoyBase.

Business The business tier is the middle tier comprised of business logic applications (programs). In some GRIN-Global installations this tier, as well as the data and presentation tiers, reside on the user's stand-alone PC. However, in larger installations, the business and data tiers will most likely be installed on a local area network or an intranet server. The business tier:

- contains a full complement of applications (software modules)
- is accessible to *any* computer connected to the internet via standard protocols
- provides data in XML format, using the SOAP protocol



Data This third tier is the database and transaction server where the PGR data is permanently stored. The illustration here shows four different database "products." Typically, an organization will use only one of the four database products, depending on their preferences.

Miscellaneous Facts

Text Box Search Fields

In text box searches, the Search Engine only searches certain database fields. (This is one of the reasons why searching via the QBE method is preferred for GRIN-Global searches.)

To add additional fields to the text search capabilities, add the fieldname into sys_search_autofields table. The Middle Tier search will automatically handle the additional field(s). However, adding additional fields will require more time to complete the search.

As an example, the SQL here can be used in the Management Studio to insert a row for searching the accession group name:

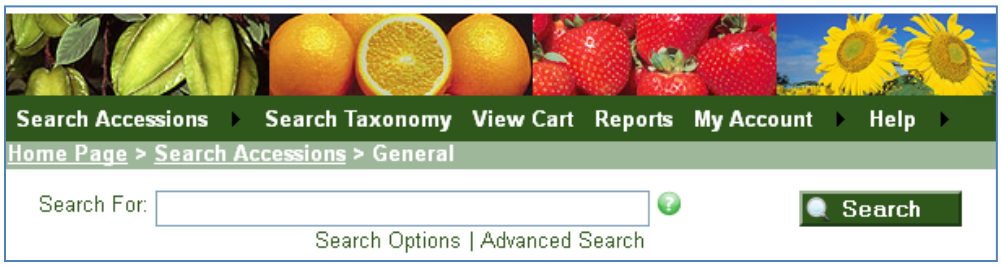
```
INSERT INTO sys_search_autofield
SELECT sys_table_field_id, GetDate(), '1', NULL, NULL, GetDate(), '1'
FROM sys_table_field stf INNER JOIN sys_table st ON stf.sys_table_id =
st.sys_table_id
WHERE st.table_name = 'accession_inv_group' AND stf.field_name = 'group_name'
```

Text Box Searches
The initial design included the following fields in the text box search:

Table Name	Field Name
accession	accession_number_part1, accession_number_part2, accession_number_part3, note
accession_ipr	ipr_number, ipr_crop_name, ipr_full_name, note
accession_inv_name	plant_name
accession_pedigree	description
cooperator	last_name, first_name
crop	name
geography	adm1, adm 2, adm3, adm4, country_code
inventory	inventory_number_part1, inventory_number_part2, inventory_number_part3,
taxonomy_common_name	name, simplified_name
taxonomy_family	family_name, alternate_name
taxonomy_genus	genus_name
taxonomy_species	nomen_number, species_name, name, alternate_name
code_value_lang	title

Curator Tool:

Public Website:



Appendix: SQL for Displaying Dataview Field Names

Note Any sample SQL included in this document has been tested under Microsoft SQL Server.

SQL Query for Displaying Tables and Table Fields

```
/* tables and table fields */

use gringlobal;
select
	st.table_name,
	stf.field_name,
	sl2.title as language_title,
	stl.title as table_title,
	stl.description as table_description,
	stfl.title as field_title,
	stfl.description as field_description
from
	sys_table st
	inner join sys_table_field stf
		on st.sys_table_id = stf.sys_table_id
	left join sys_table_field_lang stfl
		on stf.sys_table_field_id = stfl.sys_table_field_id
	left join sys_lang sl2
		on stfl.sys_lang_id = sl2.sys_lang_id
	left join sys_table_lang stl
		on st.sys_table_id = stl.sys_table_id
	left join sys_lang sl1
		on stl.sys_lang_id = sl1.sys_lang_id
order by
	st.table_name, stf.field_ordinal
```

SQL Query for Displaying Dataviews and Dataview Fields

```
/* dataviews and dataview fields */

use gringlobal;

select
	sd.dataview_name,
	sdf.field_name,
	sl2.title as language_title,
	sdl.title as dataview_title,
	sdl.description as dataview_description,
	sdfl.title as field_title,
	sdfl.description as field_description
from
	sys_dataview sd
	inner join sys_dataview_field sdf
		on sd.sys_dataview_id = sdf.sys_dataview_id
	left join sys_dataview_field_lang sdfl
		on sdf.sys_dataview_field_id = sdfl.sys_dataview_field_id
	left join sys_lang sl2
		on sdfl.sys_lang_id = sl2.sys_lang_id
	left join sys_dataview_lang sdl
		on sd.sys_dataview_id = sdl.sys_dataview_id
	left join sys_lang sl1
		on sdl.sys_lang_id = sl1.sys_lang_id
order by
	sd.dataview_name, sdf.sort_order
```

SQL Query for Downloading the Schema

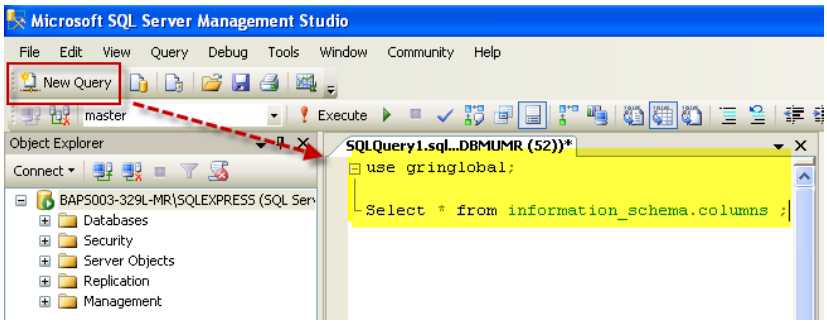
For those administrators who would prefer to download the GRIN-Global schema into a spreadsheet, the following directions explain how to do so using Microsoft's SQL Server Management Studio.

1. Start SQL Server Management Studio: **Start | All Programs | Microsoft SQL Server 2008 (2010) | SQL Server Management Studio**

2. Click **New Query**; input the following commands:

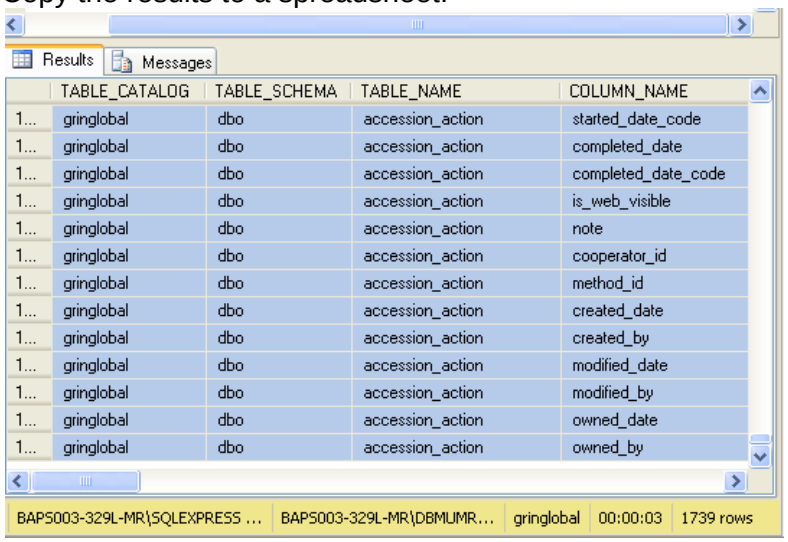
```
use gringlobal;
Select * from information_schema.columns ;
```

Appendix: SQL for Displaying Dataview Field Names



3. Click the **Execute** button.

4. Copy the results to a spreadsheet:



Results | Messages

	TABLE_CATALOG	TABLE_SCHEMA	TABLE_NAME	COLUMN_NAME
1...	gringlobal	dbo	accession_action	started_date_code
1...	gringlobal	dbo	accession_action	completed_date
1...	gringlobal	dbo	accession_action	completed_date_code
1...	gringlobal	dbo	accession_action	is_web_visible
1...	gringlobal	dbo	accession_action	note
1...	gringlobal	dbo	accession_action	cooperator_id
1...	gringlobal	dbo	accession_action	method_id
1...	gringlobal	dbo	accession_action	created_date
1...	gringlobal	dbo	accession_action	created_by
1...	gringlobal	dbo	accession_action	modified_date
1...	gringlobal	dbo	accession_action	modified_by
1...	gringlobal	dbo	accession_action	owned_date
1...	gringlobal	dbo	accession_action	owned_by

BAPS003-329L-MR\SQLEXPRESS ... | BAPS003-329L-MR\DBMUMR... | gringlobal | 00:00:03 | 1739 rows

Appendix: Changes in this Document

– *October 31, 2016*

- edited the security section drastically in order to clarify; also referenced new material added to a separate document

– *October 17, 2016*

- edited the reports section slightly; verified the .txt file locations

– *August 19, 2016*

- included new information on the Web Login regarding CT users having special PW privileges

– *June 2 & March 1, 2016*

- edited information on the .txt files

– *October 22, 2015*

- edited the Code Groups section – added SQL code details for displaying the Code Groups and Codes; also edited the text

– *May 4, 2015*

- edited Web Application parameters; including the adding of the added the SysUserPassword parameters

– April 9*, 2015*

- review and edit of the majority of the document
- replaced outdated screens

– *February 13, 2015*

- expanded background information on the folders installed by the CT (see GG User Files)

– *November 14, 2014*

- corrected minor typo on cover page

Appendix: Changes in this Document

– *October 6, 2014*
  - edited WebApplication WebSearchableIndexes parameter

– *July 10, 2014*
  - explained the three text files: AppSettings.txt, ReportsMapping.txt, and WebServiceURL.txt
  - described how to add icons and ”friendly names” for the CT List items

– *December 16, 2013*
  - added information about including reports in both the Curator Tool and the Public Website

– *November 12, 2013*
  - added more information about datatriggers

– *November 5, 2013*
  - minor edits added
  - added some [tbd] note – “to be developed” – due to time constraints, some items have been noted as needing some correction or new screen captures
  - added a section about the dataview “Shoe Dependencies” feature

– *April 16, 2013*
  - included information on AppSettings.txt

– *September 10, 2012*
  - included a note and a link to the separate guide that explains modifying and editing the interface for languages other than English

– *January 31, 2012*
  - included additional sections about extracting code values: *Extracting All Codes for a Specific Group Name* and *Extracting All of the Code Values*
  - removed an outdated section on searches

– *October 26, 2011*
  - included detailed instructions on obtaining a Google **GoogleMapsAPIKey**