Sloan Digital Sky Survey

From FITS to SQL
Loading and Publishing the SDSS Data

Ani Thakar and Alex Szalay (JHU), Jim Gray (Microsoft Research)

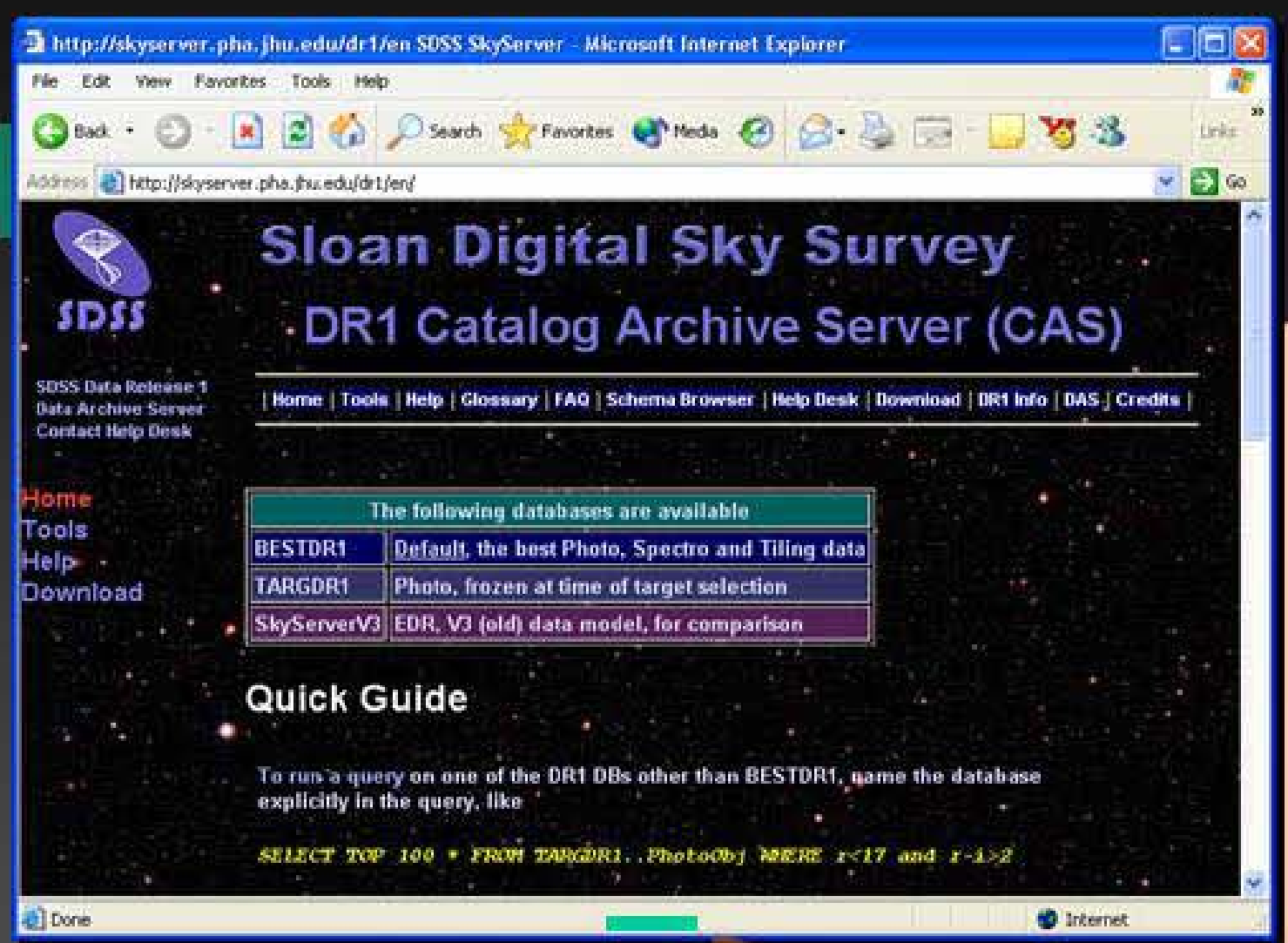
Abstract

The Sloan Digital Sky Survey Data Release 1 (DR1) contains nearly 1 TB of catalog data published online as the Catalog Archive Server(CAS) and accessible via the **SkyServer** web interface. The DR1 CAS is the end product of a data loading pipeline that transforms the FITS file data exported by the SDSS Operational Database or OpDB, converts it to CSV (comma separated values), and loads it into a MS Windows-based relational database management system (SQL Server DBMS).

Loading the data is potentially the most time-consuming and labor-intensive part of archive operations, and it is also the most critical: <u>it is realistically your one chance to get the data right</u>. We have attempted to automate it as much as possible, and to make it easy to diagnose data and loading errors. We describe this pipeline, focusing on the highly automated SQL data loader framework (sqlLoader) - a **distributed workflow system** of modules that check, load, validate and publish the data to the databases. The workflow is described by a **directed acyclic graph (DAG)** whose nodes are the processing modules, and it is designed for parallel loading on a cluster of load-servers. The pipeline first reads the data from Samba-mounted CSV files on the LINUX side and stuffs it into the SQL databases on the Windows side.

The **validation** step, in particular, represents a systematic and thorough scrubbing of the data before it is deemed worthy of publishing. The **finish** step merges the different data products (imaging, spectra, tiling) into a set of linked tables that can be efficiently searched with specialized indices and pre-computed joins.

We are in the process of making the sqlLoader generic and portable enough so that other archives may adapt it to load, validate and publish their data.



http://skyserver.pha.jhu.edu/dr1/en SDSS SkyServer - Microsoft Internet Explorer

Address: http://skyserver.pha.jhu.edu/dr1/en/

Sloan Digital Sky Survey

DR1 Catalog Archive Server (CAS)

| Home | Tools | Help | Glossary | FAQ | Schema Browser | Help Desk | Download | DR1 Info | DAS | Credits |

SDSS Data Release 1
Data Archive Server
Contact Help Desk

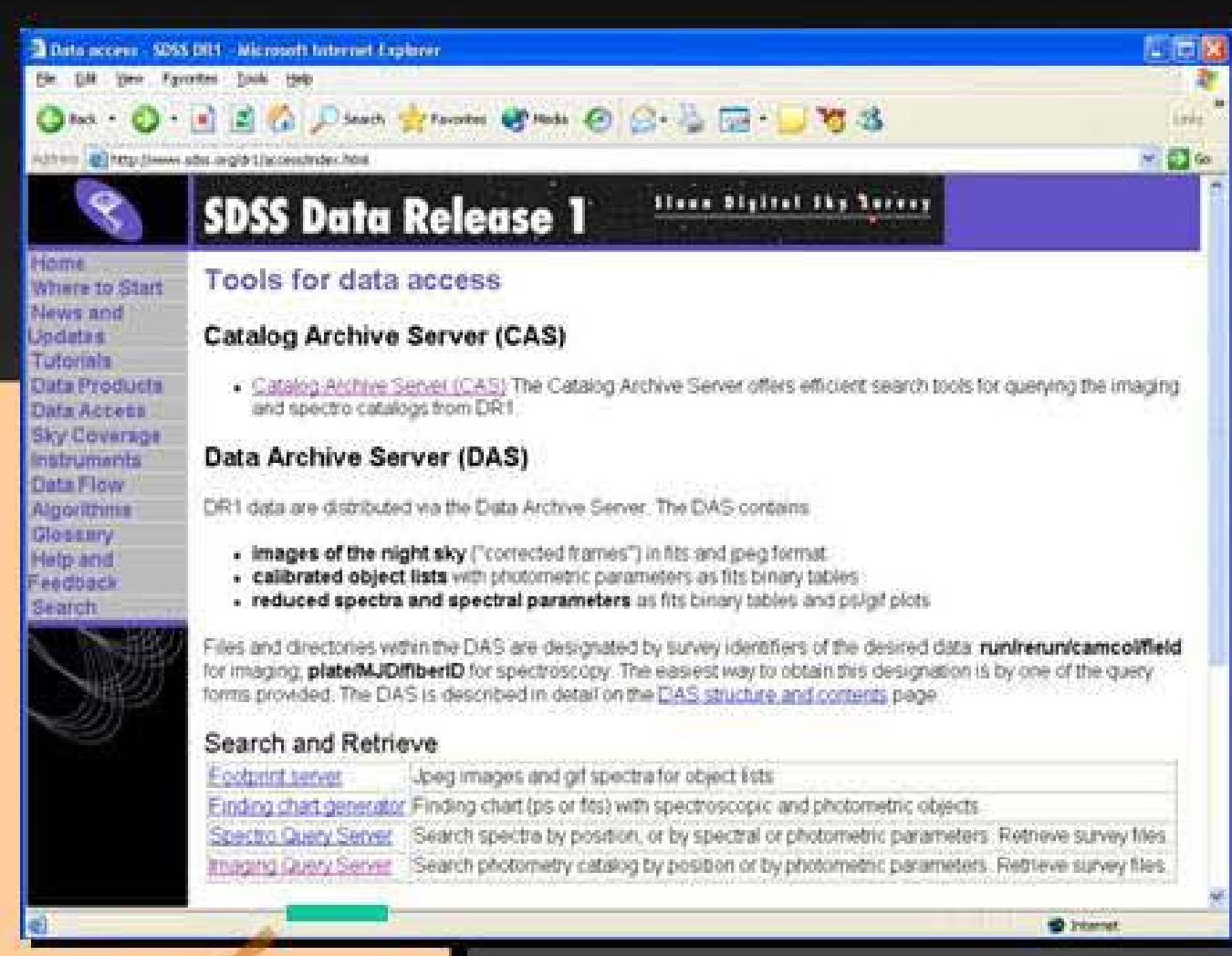
Home
Tools
Help
Download

The following databases are available	
BESTDR1	Default, the best Photo, Spectro and Tiling data
TARGDR1	Photo, frozen at time of target selection
SkyServerV3	EDR, V3 (old) data model, for comparison

Quick Guide

To run a query on one of the DR1 DBs other than BESTDR1, name the database explicitly in the query, like

```
SELECT TOP 100 * FROM TARGDR1..PhotoObj WHERE r<17 and r-i>2
```



Data access - SDSS DR1 - Microsoft Internet Explorer

Address: http://www.sdss.org/dr1/access/index.html

SDSS Data Release 1 — Sloan Digital Sky Survey

Home
Where to Start
News and Updates
Tutorials
Data Products
Data Access
Sky Coverage
Instruments
Data Flow
Algorithms
Glossary
Help and Feedback
Search

Tools for data access

Catalog Archive Server (CAS)

- Catalog Archive Server (CAS) The Catalog Archive Server offers efficient search tools for querying the imaging and spectro catalogs from DR1.

Data Archive Server (DAS)

DR1 data are distributed via the Data Archive Server. The DAS contains

- **images of the night sky** ("corrected frames") in fits and jpeg format
- **calibrated object lists** with photometric parameters as fits binary tables
- **reduced spectra and spectral parameters** as fits binary tables and ps/gif plots

Files and directories within the DAS are designated by survey identifiers of the desired data: **run/rerun/camcol/field** for imaging, **plate/MJD/fiberID** for spectroscopy. The easiest way to obtain this designation is by one of the query forms provided. The DAS is described in detail on the DAS structure and contents page.

Search and Retrieve

Footprint server	Jpeg images and gif spectra for object lists
Finding chart generator	Finding chart (ps or fits) with spectroscopic and photometric objects
Spectro Query Server	Search spectra by position, or by spectral or photometric parameters. Retrieve survey files
Imaging Query Server	Search photometry catalog by position or by photometric parameters. Retrieve survey files

SDSS Data Release 1
http://www.sdss.org/dr1/

Coverage

2100 sq.deg (imaging), 1400 sq.deg. (spectra)
20% of total survey area
5-6 times size of EDR (Early Data Release)

Catalog data
Catalog Archive Server (SkyServer)
http://skyserver.pha.jhu.edu/dr1/

Released June 2003, 1 Tbyte DBMS
85M photometric (image) objects
160k spectroscopic objects
2 versions of data: BEST and TARGET
Relational Data Model (tables)

Image data
Data Archive Server
http://www.sdss.org/dr1/access/

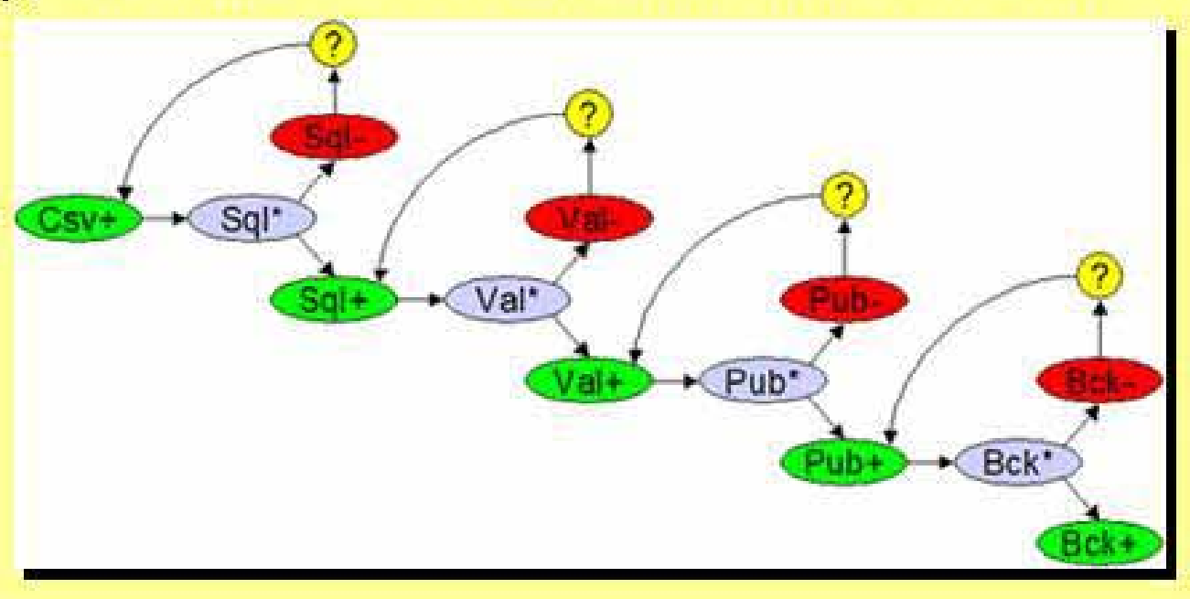
Released April 2003, 2-3 Tbyte flat files
Atlas images (cutouts) – PNG, FITS
Spectra - FITS, GIF
Corrected frames - FITS
Binned images - FITS
Mask images - FITS

The Loading Process

The basic processing entity is a *task*. A task is started when a data *chunk* is exported by the OpDB. Exported chunks are converted to CSV format, and are contained in a single directory. There are several different export types: TARGET, BEST, RUNS, PLATE and TILING. Each task comes with a id number that is unique within its category.

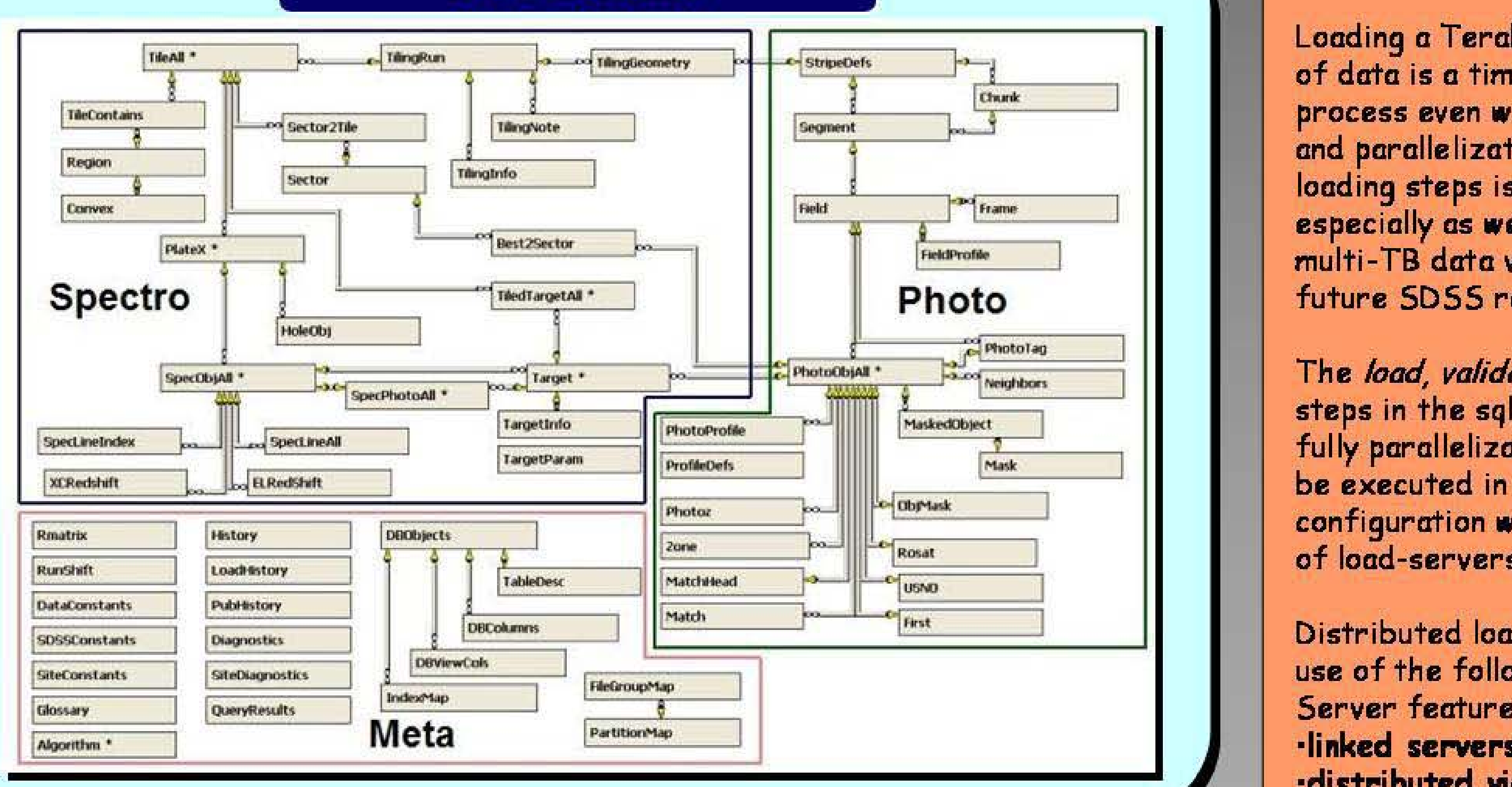
The loading process consists of *steps*. The first step is to load each chunk of data into a separate *task DB*, containing only a thin set of indices. Then we *validate* the data. This includes verifying that there are no primary key collisions and all foreign keys point to a valid record. We build several ancillary tables for spatial searches (HTM, Neighbors, etc.) After the validation step we *publish* the data: we perform a DB-to-DB copy, where the target is the final production database. After publishing it, we make a backup of the task DB.

At the very end, all the different datasets are merged together in the *finish* step, and indices are created for efficient data mining.



A state-machine representation of the loading process. Each step is a sequence of rather complex steps in itself. The yellow question marks represent a manual Undo step, which is performed as necessary.

DR1 Schema

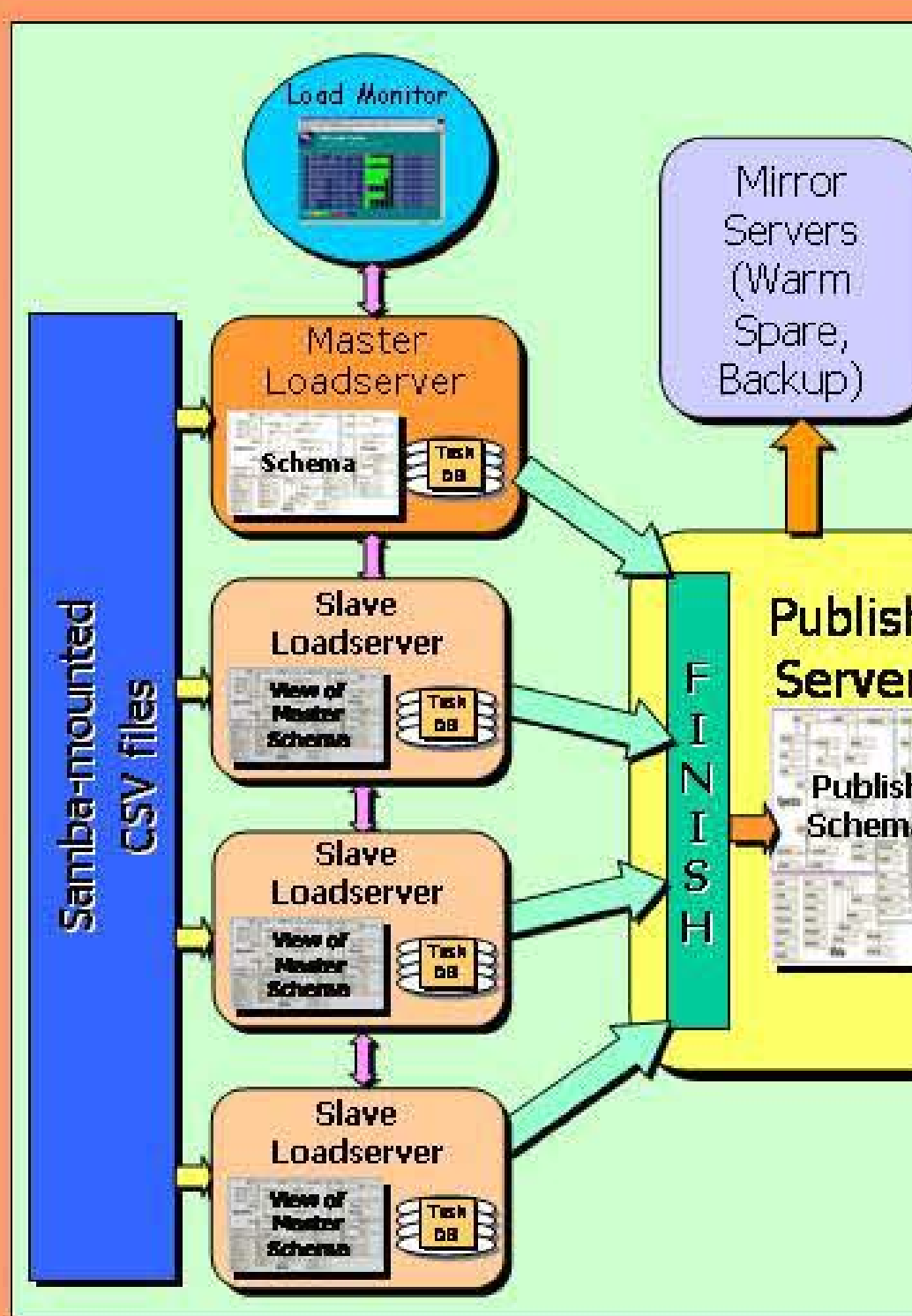


Distributed Loading

Loading a Terabyte or more of data is a time-consuming process even with fast disks, and parallelization of the loading steps is a big help, especially as we get into the multi-TB data volumes of future SDSS releases.

The *load*, *validate* and *publish* steps in the sqlLoader are fully parallelizable and can be executed in a distributed configuration with a cluster of load-servers.

Distributed loading makes use of the following SQL Server features:
·**linked servers** with
·**distributed views** and
·**distributed transactions**.

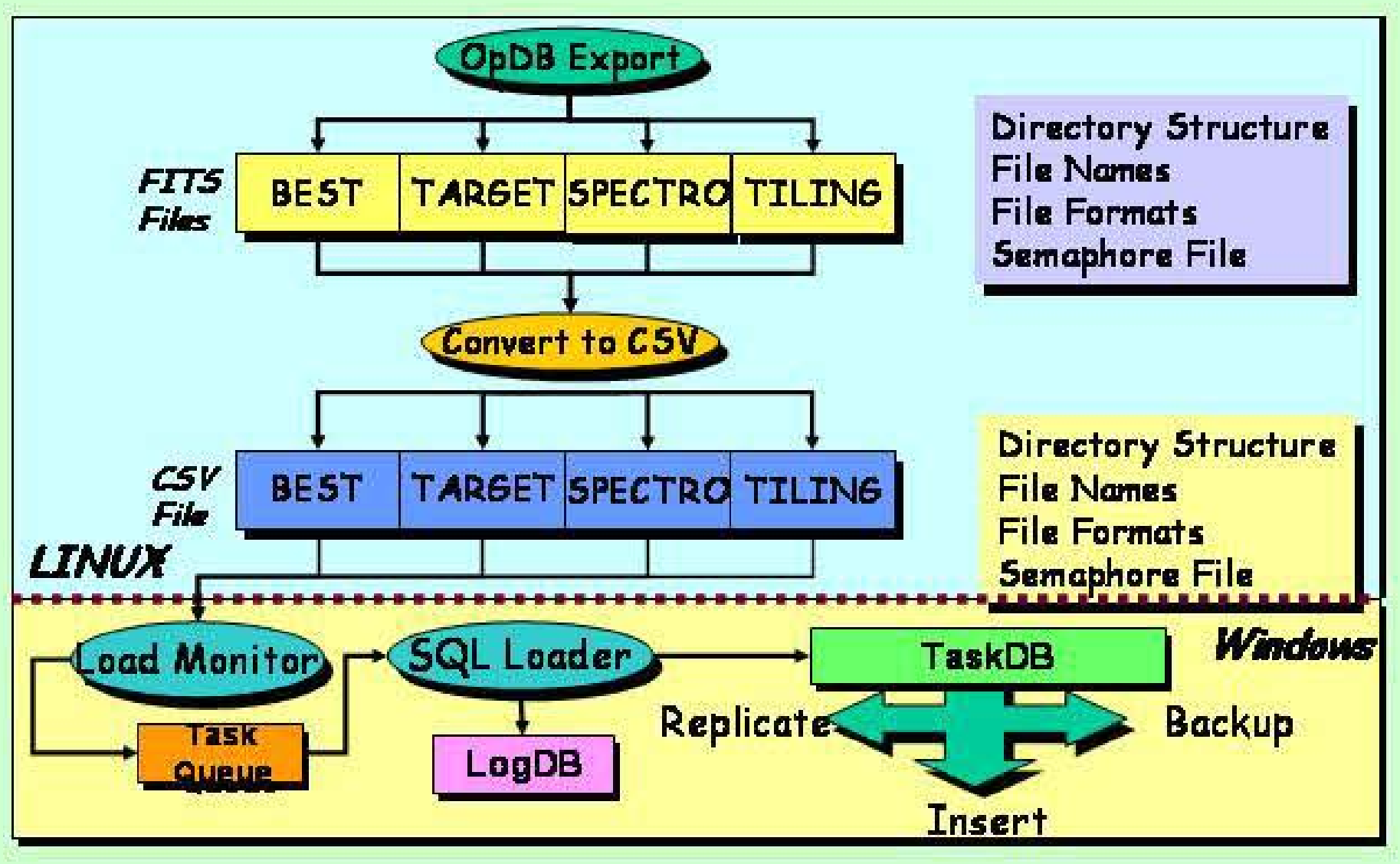


After loading, validating and publishing is done in parallel, the merging of the parallel data streams and the finish step are performed sequentially on the publish server.

Data Export Pipeline

The data is exported from the OpDB in the form of FITS files that are organized into blocks called *chunks*. Each chunk is the result of a block of data being <u>resolved</u> and exported by the OpDB. Four different datasets are exported: two for imaging data and one each for spectroscopic and tiling data. The imaging data is resolved at least twice – once when the spectroscopic targets are chosen, and once when the data is recalibrated with the latest, greatest calibration. These datasets are called TARGET and BEST respectively.

All the data must be converted to CSV format so it can be stuffed into the databases using **bulk insertion**. The CSV files also serve as a blood-brain barrier between the LINUX and Windows worlds. The CSV files are Samba-mounted and loaded into temporary DBs before being bulk-copied to their final destination as part of the *publish* step. All the tasks and steps are logged into a logDB that is queried by the Load Monitor to generate the various logs that it provides.



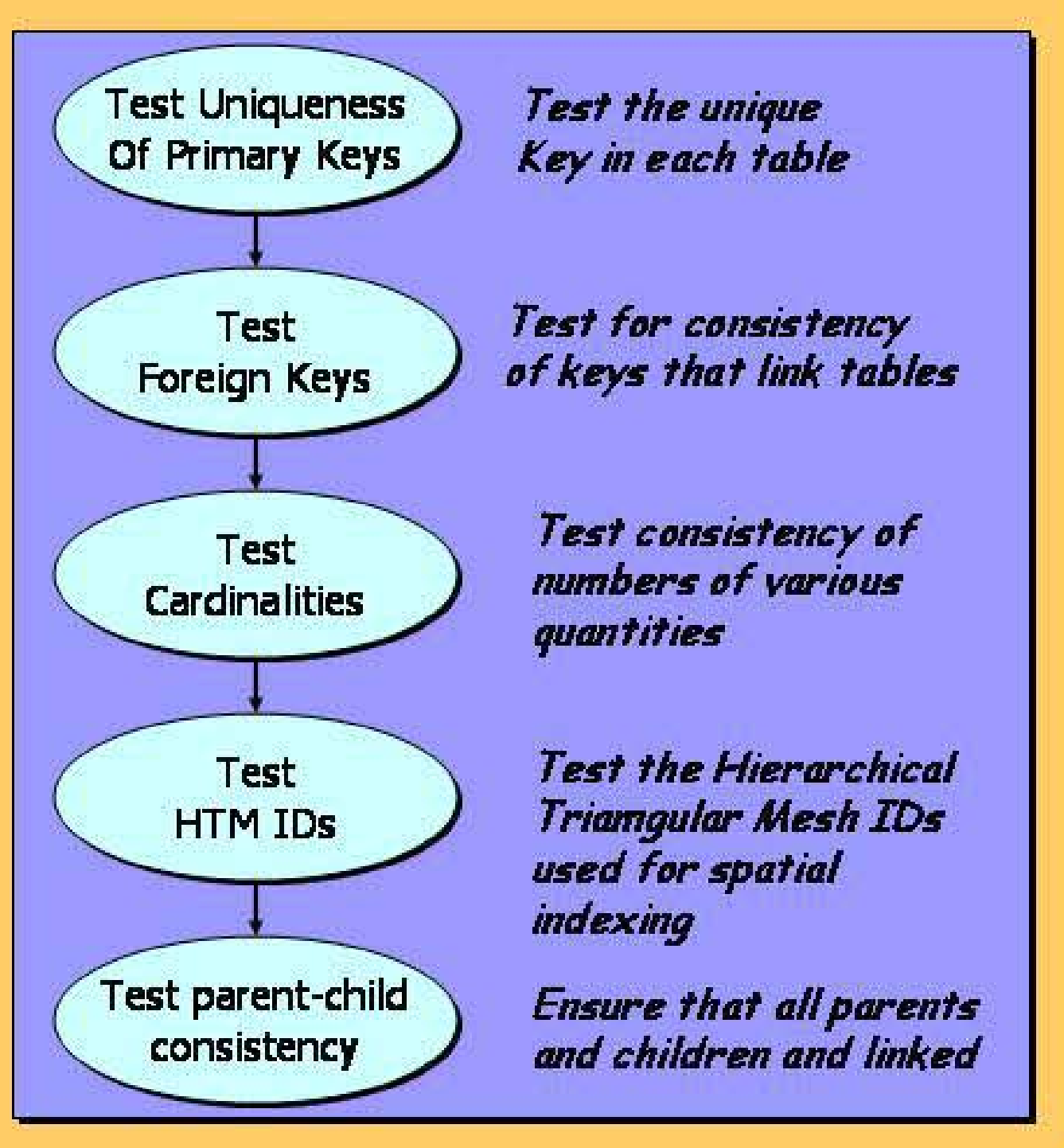
Data Validation

Validation is perhaps the most important step in the loading process. The speed, integrity and convenience that databases offer come at a price: data once published cannot be retracted or corrected easily. This is not only because the loading process itself is difficult and arduous, but also because the data must always be available once science has been done with it. <u>Hence it is crucial to get the data right the first time.</u>

The validate step in sqlLoader represents a systematic scrubbing and sanity-check of the data, from a scientific as well as data integrity point of view. The figure on the right shows the various operations that are performed on the data.

The primary and foreign key tests are run on all the tables. The photo (imaging) and spectro tables are tested for HTM IDs, which are 64-bit IDs that provide fast spatial indexing according to the Hierarchical Triangular Mesh indexing scheme. The image data also has parent-child relationships defined for deblended objects. The referential integrity of these is also checked as part of the validation. Finally, the consistency of counts of various quantities is checked.

This validation process has proven invaluable in finding numerous inconsistencies and errors in the data and catching them early, during the testing of DR1 rather than after the data is published.



Test Uniqueness Of Primary Keys — *Test the unique Key in each table*

Test Foreign Keys — *Test for consistency of keys that link tables*

Test Cardinalities — *Test consistency of numbers of various quantities*

Test HTM IDs — *Test the Hierarchical Triangular Mesh IDs used for spatial indexing*

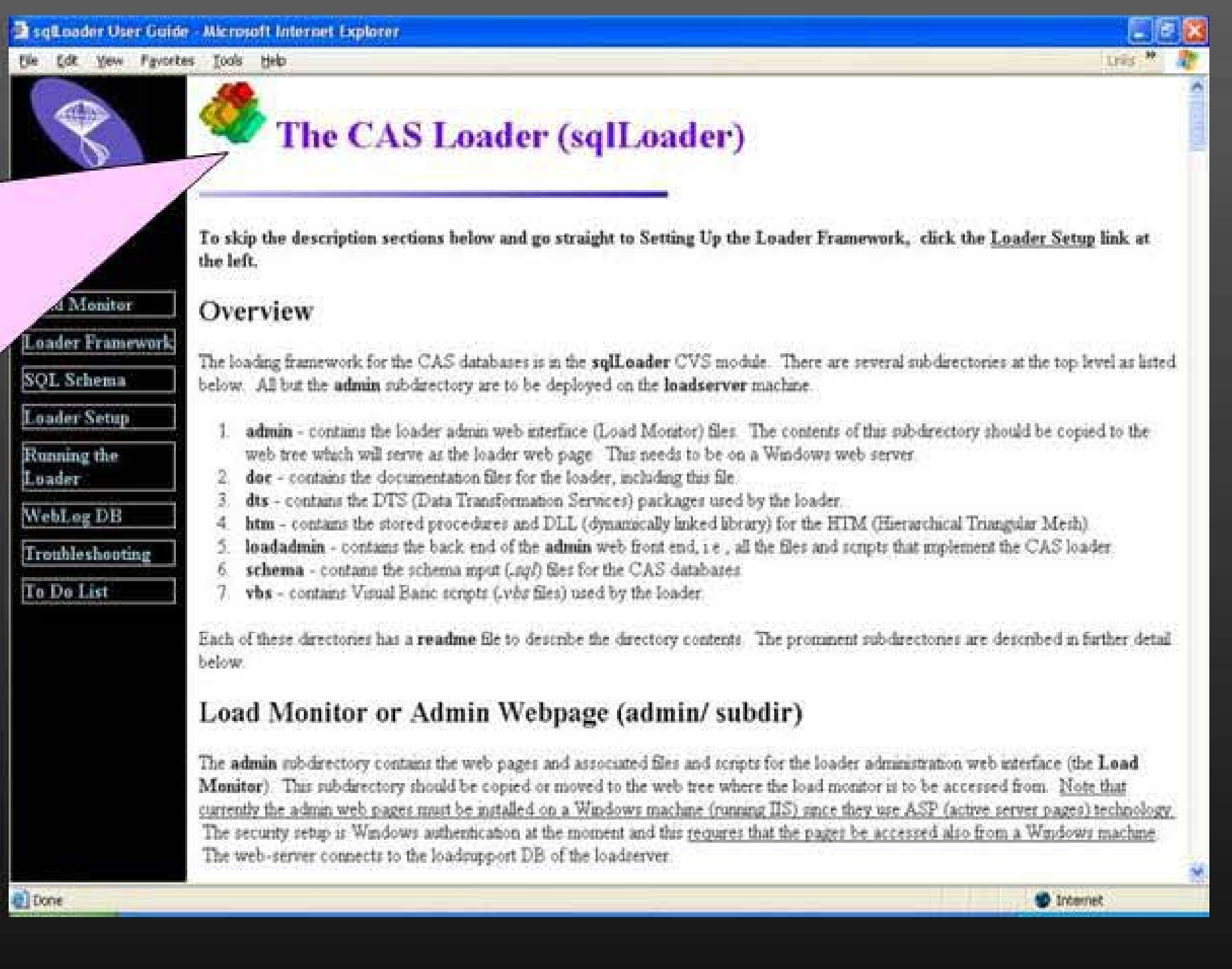
Test parent-child consistency — *Ensure that all parents and children and linked*

Help and Documentation

Online help is bundled with the sqlLoader product, and includes a user guide that describes how to set up and run the sqlLoader. The troubleshooting section lists solutions to typical problems.

There is also a document that details each step in the loader processing. When compared with the log for a loader task in the Load Monitor, this enables the user to troubleshoot a problem that causes the loader to get stuck.

The sqlLoader has enabled the entire loading for DR1 to be completed largely as a turnkey operation with very little human intervention.



sqlLoader User Guide - Microsoft Internet Explorer

Load Monitor
Loader Framework
SQL Schema
Loader Setup
Running the Loader
WebLog DB
Troubleshooting
To Do List

The CAS Loader (sqlLoader)

To skip the description sections below and go straight to Setting Up the Loader Framework, click the Loader Setup link at the left.

Overview

The loading framework for the CAS databases is in the **sqlLoader** CVS module. There are several subdirectories at the top level as listed below. All but the **admin** subdirectory are to be deployed on the **loadserver** machine.

1. **admin** - contains the loader admin web interface (Load Monitor) files. The contents of this subdirectory should be copied to the web tree which will serve as the loader web page. This needs to be on a Windows web server.
2. **doc** - contains the documentation files for the loader, including this file.
3. **dts** - contains the DTS (Data Transformation Services) packages used by the loader.
4. **htm** - contains the stored procedures and DLL (dynamically linked library) for the HTM (Hierarchical Triangular Mesh).
5. **loadadmin** - contains the back end of the **admin** web front end, i.e., all the files and scripts that implement the CAS loader.
6. **schema** - contains the schema input (*.sql*) files for the CAS databases.
7. **vbs** - contains Visual Basic scripts (*.vbs* files) used by the loader.

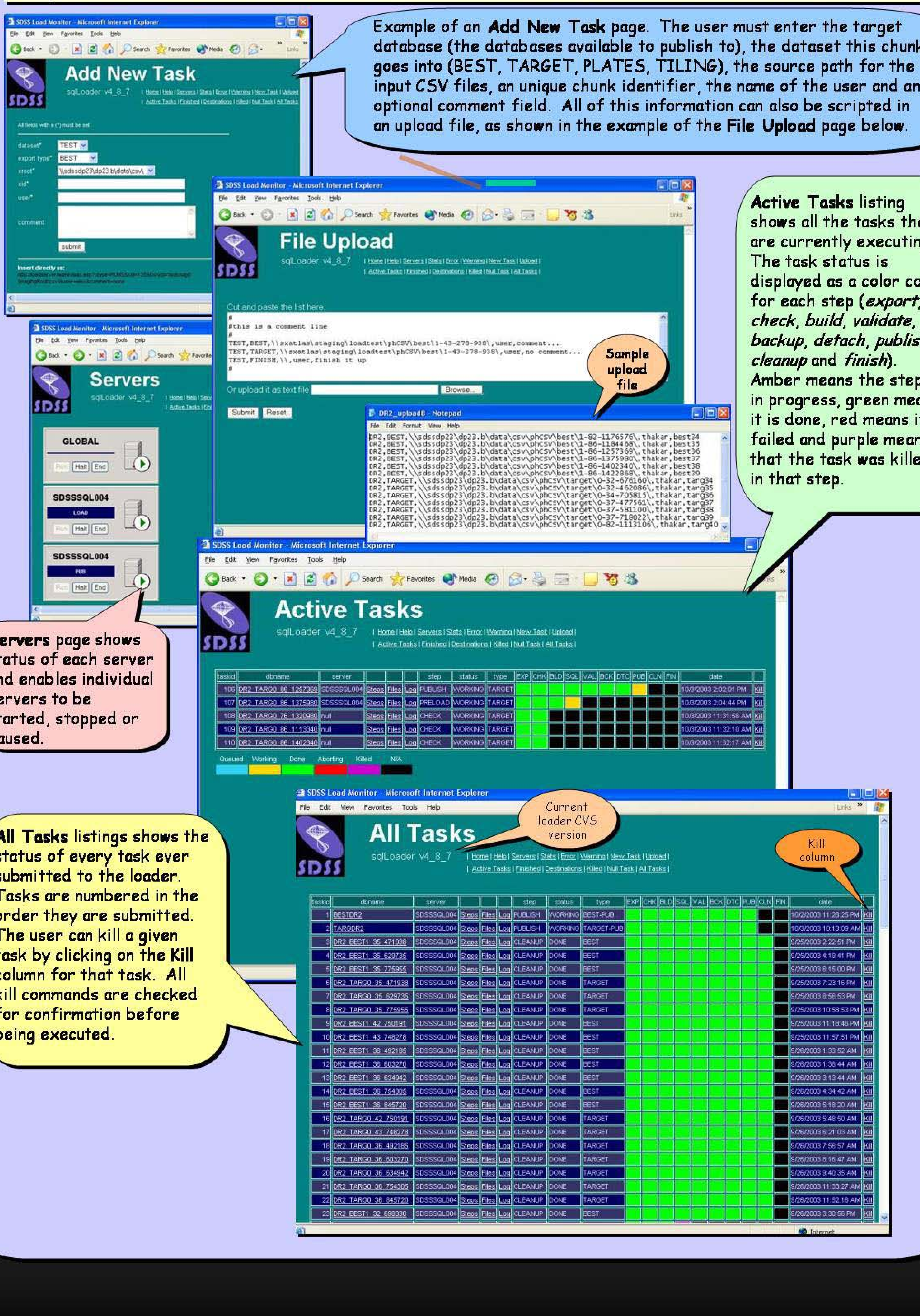
Each of these directories has a **readme** file to describe the directory contents. The prominent subdirectories are described in further detail below.

Load Monitor or Admin Webpage (admin/ subdir)

The **admin** subdirectory contains the web pages and associated files and scripts for the loader administration web interface (the **Load Monitor**). This subdirectory should be copied or moved to the web tree where the load monitor is to be accessed from. <u>Note that currently the admin web pages must be installed on a Windows machine (running IIS) since they use ASP (active server pages) technology.</u> The security setup is Windows authentication at the moment and this <u>requires that the pages be accessed also from a Windows machine.</u> The web-server connects to the loadsupport DB of the loadserver.

The Load Monitor

The Load Monitor is the admin web interface to the sqlLoader. It enables job submission, control and tracking via a user-friendly GUI. Loading jobs (tasks) can be submitted either a single chunk at a time or bulk-uploaded with the file upload feature. Tasks can be monitored at several levels, and information is available on the status of the individual files being loaded, detailed logs of each step in the task, and separate listings of errors and warnings encountered. Tasks are listed in a task table that shows the status of each step in the task at a glance in a color-coded chart. Sample Load Monitor screens are shown below to illustrate the features that are available.



Example of an **Add New Task** page. The user must enter the target database (the databases available to publish to), the dataset this chunk goes into (BEST, TARGET, PLATES, TILING), the source path for the input CSV files, an unique chunk identifier, the name of the user and an optional comment field. All of this information can also be scripted in an upload file, as shown in the example of the **File Upload** page below.

SDSS Load Monitor - Microsoft Internet Explorer

Add New Task
sqlLoader v4_8_7

All fields with a (*) must be set

dataset*: TEST
export type*: BEST
xroot*: \\sdssdp23\dp23.b\data\csv\
xid*:
user*:
comment:
submit

SDSS Load Monitor - Microsoft Internet Explorer

File Upload
sqlLoader v4_8_7

Cut and paste the list here:

```
#
#this is a comment line
#
TEST,BEST,\\sxatlas\staging\loadtest\phCSV\best\1-43-278-938\,user,comment...
TEST,TARGET,\\sxatlas\staging\loadtest\phCSV\best\1-43-278-938\,user,no comment...
TEST,FINISH,\\,user,finish it up
#
```

Or upload it as text file: Browse...
Submit Reset

Sample upload file

DR2_upload8 - Notepad

```
DR2,BEST,\\sdssdp23\dp23.b\data\csv\phCSV\best\1-82-1176576\,thakar,best34
DR2,BEST,\\sdssdp23\dp23.b\data\csv\phCSV\best\1-86-1184468\,thakar,best35
DR2,BEST,\\sdssdp23\dp23.b\data\csv\phCSV\best\1-86-1257369\,thakar,best36
DR2,BEST,\\sdssdp23\dp23.b\data\csv\phCSV\best\1-86-1375980\,thakar,best37
DR2,BEST,\\sdssdp23\dp23.b\data\csv\phCSV\best\1-86-1402340\,thakar,best38
DR2,BEST,\\sdssdp23\dp23.b\data\csv\phCSV\best\1-86-1422868\,thakar,best39
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-32-676160\,thakar,targ34
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-32-462086\,thakar,targ35
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-34-705815\,thakar,targ36
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-37-477561\,thakar,targ37
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-37-581100\,thakar,targ38
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-37-718022\,thakar,targ39
DR2,TARGET,\\sdssdp23\dp23.b\data\csv\phCSV\target\0-82-1113106\,thakar,targ40
```

SDSS Load Monitor - Microsoft Internet Explorer

Servers
sqlLoader v4_8_7

GLOBAL — Run Halt End
SDSSSQL004 — LOAD — Run Halt End
SDSSSQL004 — PUB — Run Halt End

Servers page shows status of each server and enables individual servers to be started, stopped or paused.

Active Tasks listing shows all the tasks that are currently executing. The task status is displayed as a color code for each step (*export*, *check*, *build*, *validate*, *backup*, *detach*, *publish*, *cleanup* and *finish*). Amber means the step is in progress, green means it is done, red means it failed and purple means that the task was killed in that step.

SDSS Load Monitor - Microsoft Internet Explorer

Active Tasks
sqlLoader v4_8_7
| Home | Help | Servers | Stats | Error | Warning | New Task | Upload |
| Active Tasks | Finished | Destinations | Killed | Null Task | All Tasks |

taskid	dbname	server				step	status	type	EXP	CHK	BLD	SQL	VAL	BCK	DTC	PUB	CLN	FIN	date	
106	DR2_TARG0_86_1257369	SDSSSQL004	Steps	Files	Log	PUBLISH	WORKING	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟨	⬛	⬛	10/3/2003 2:02:01 PM	Kill
107	DR2_TARG0_86_1375980	SDSSSQL004	Steps	Files	Log	PRELOAD	WORKING	TARGET	🟩	🟩	🟩	🟨	⬛	⬛	⬛	⬛	⬛	⬛	10/3/2003 2:04:44 PM	Kill
108	DR2_TARG0_76_1320960	null	Steps	Files	Log	CHECK	WORKING	TARGET	🟩	🟩	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	10/3/2003 11:31:58 AM	Kill
109	DR2_TARG0_86_1113340	null	Steps	Files	Log	CHECK	WORKING	TARGET	🟩	🟩	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	10/3/2003 11:32:10 AM	Kill
110	DR2_TARG0_86_1402340	null	Steps	Files	Log	CHECK	WORKING	TARGET	🟩	🟩	⬛	⬛	⬛	⬛	⬛	⬛	⬛	⬛	10/3/2003 11:32:17 AM	Kill

Queued | Working | Done | Aborting | Killed | N/A

SDSS Load Monitor - Microsoft Internet Explorer

All Tasks
sqlLoader v4_8_7
| Home | Help | Servers | Stats | Error | Warning | New Task | Upload |
| Active Tasks | Finished | Destinations | Killed | Null Task | All Tasks |

Current loader CVS version

Kill column

taskid	dbname	server				step	status	type	EXP	CHK	BLD	SQL	VAL	BCK	DTC	PUB	CLN	FIN	date	
1	BESTDR2	SDSSSQL004	Steps	Files	Log	PUBLISH	WORKING	BEST-PUB	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	⬛	10/2/2003 11:28:25 PM	Kill
2	TARGDR2	SDSSSQL004	Steps	Files	Log	PUBLISH	WORKING	TARGET-PUB	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	⬛	10/3/2003 10:13:09 AM	Kill
3	DR2_BEST1_35_471938	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 2:22:51 PM	Kill
4	DR2_BEST1_35_629735	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 4:18:41 PM	Kill
5	DR2_BEST1_35_775955	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 6:15:00 PM	Kill
6	DR2_TARG0_35_471938	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 7:23:16 PM	Kill
7	DR2_TARG0_35_629735	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 8:56:53 PM	Kill
8	DR2_TARG0_35_775955	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 10:58:53 PM	Kill
9	DR2_BEST1_42_750191	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 11:18:46 PM	Kill
10	DR2_BEST1_43_748278	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/25/2003 11:57:51 PM	Kill
11	DR2_BEST1_36_492185	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 1:33:52 AM	Kill
12	DR2_BEST1_36_603270	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 1:38:44 AM	Kill
13	DR2_BEST1_36_634942	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 3:13:44 AM	Kill
14	DR2_BEST1_36_754305	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 4:34:42 AM	Kill
15	DR2_BEST1_36_845720	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 5:18:20 AM	Kill
16	DR2_TARG0_42_750191	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 5:48:50 AM	Kill
17	DR2_TARG0_43_748278	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 6:21:03 AM	Kill
18	DR2_TARG0_36_492185	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 7:56:57 AM	Kill
19	DR2_TARG0_36_603270	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 8:16:47 AM	Kill
20	DR2_TARG0_36_634942	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 9:40:35 AM	Kill
21	DR2_TARG0_36_754305	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 11:33:27 AM	Kill
22	DR2_TARG0_36_845720	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	TARGET	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 11:52:16 AM	Kill
23	DR2_BEST1_32_698330	SDSSSQL004	Steps	Files	Log	CLEANUP	DONE	BEST	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	🟩	⬛	9/26/2003 3:30:56 PM	Kill

All Tasks listings shows the status of every task ever submitted to the loader. Tasks are numbered in the order they are submitted. The user can kill a given task by clicking on the **Kill** column for that task. All kill commands are checked for confirmation before being executed.