Using the raytracing programs:

Step 1: Creating an electron density profile
The raytracing program can use either a 1-D profile which varies with altitude or a 2-D profile which varies with altitude and latitude.

The 1-D profiles are represented by text files with two columns, the first is altitude in km and the second is electron density in cm^-3. These can be generated from IRI or by any other method. At the start of a 1-D profile, the number of altitudes is indicated (see example).

The 2-D profiles contain rows of altitude profiles of electron density. Each row is the electron density profile at one latitude. The first line in the 2-D profiles contains the following: the number of altitudes, starting altitude (km), altitude increment (km), number of latitudes, starting latitude, latitude increment (see example).

The idl program 'profilecreate.pro' can be used to convert a 1-d listing of electron densities into a 2-D profile and can add in a blob or trough of any size and amplitude. The blob or trough is added as a Gaussian distribution on the otherwise smooth background. Call the program and specify the blob amplitude (or trough by selecting a value less than 1.0), the blob width (standard deviation in latitude degrees), and blob height (standard deviation in km). At the moment, both the location of the blob and initial 1-D profile must be changed within the program. More than one blob or trough could be placed in a given profile by copy and pasting the appropriate portion of the program. Also, the blob is created by multiplying the existing electron density at a point by the appropriate amplitude, so blobs in the E-region will be different that the F-region. When creating E-region blobs, keep this in mind or modify the program to specify a set electron density for the blob (instead of an amplitude compared to the local background).

Note:
Until recently, IRI provided electron density profiles in cm^-3, but it now provides them in m^-3. The tracer program takes 1-D profiles in cm^-3, but 2-D profiles in m^-3 so keep this in mind. For this reason, it may be easier to simply always use 2-D profiles (just don't add any blobs or troughs if you wish to use a horizontally stratified profile).

Step 2: Setting up and starting a raytrace
A parameter file, like 'par_o.par' is used as the set-up for the tracer program. This file will specify all the details about a set of raytraces. In these parameter files, the following can be modified:
"Rays to trace" -- ('O' or 'E') specify which mode the program will trace (at UHF frequencies this will probably not matter as each mode will follow the same path)
"Trace mode" -- specify whether to have the tracer program iteratively converge ('C') the ray trace to a specific point or to just calculate one ray path ('N')
"Output mode" -- the coordinates for output (I've never changed this from Cartesian)
"Tracing parameters" -- various variable to specify the path length and step size of the raytrace (I've only ever changed maximal number of intermediate values (maximum points on a raytrace) and maximal length)
"number of traces" -- in convergence mode, the number of points to converge to (a new output file will be made for each point) (set to 1 if not converging)
"end coordinates" -- the end coordinates for convergence mode in longitude, latitude, and altitude (all that matters is the altitude if not converging)

"ionospheric parameters" -- as shown in example file, specify electron density profile to use (for 1-D start with '-1' then file name, for 2-D start with '-2' then file name)
"ellipse" and "disturbance" parameters -- ignore these, they don't affect the trace, but are needed to properly read the parameter file (if you wish, you could probably easily modify the tracer program so these can be removed, but I haven't bothered)
"Radar frequency" -- specify the frequency of the ray you wish to trace
"Start location of the rays" -- specify the starting latitude, longitude and altitude of the rays (need to start the rays at least 1000-m altitude, since the ground is simulated by a high electron density which can affect the initial step of rays with lower altitudes)
"number of elevations, azimuths, and rays" -- specify number of rays to trace for non-convergence mode (for convergence mode, only use one ray)
"number, starting elevations and azimuths" -- specify the starting elevation and azimuth of each ray to trace
e.g. "1 20.0 0.00
2 25.0 0.00
3 30.0 0.00"
will create an output file with three raytraces, one at elevation 20 degrees, one at 25 degrees and one at 30 degrees. If you are using convergence mode, specify the starting guess for elevation and azimuth to converge to the first point listed earlier.

Running the tracer program:
To run the tracer program, compile it with the command 'make tracer' and then simply run a batch file like 'runtracer.bat'. In the batch file, simply call the tracer program with a specific parameter file and a destination for the output to be created (e.g. './tracer /home/gillies/raytrace/2011/VHF/1GHz_blob1/par_o.par 1 /data/raytrace/2011/VHF/1GHz_blob1t2/'). Several parameter files can be listed in this batch file at once, although the output file names are something like 'o***.ray' for O-modes and 'e***.ray' for X-modes, so they may overwrite each other if more than one of a given mode is created in the same directory.

Step 3: Analysis of raytrace output
In idl, the program 'readrays.pro' can be used to read the ray files. Call this program and specify the directory and filename for the ray trace and a name of a structure to store the ray data in (e.g. 'oray'). The various ray parameters can then be printed or plotted by typing "print, (*oray[elevnum,azinum].rdat)(pt1:pt2).galt" (you need to specify the elevation and azimuth numbers of interest (elevnum and azinum) and you can specify the starting and ending points of the ray trace to print (pt1 and pt2)). The variables which can be printed or plotted are:
(*oray[0,0].rdat)(0:10).x -- (x-component of raytrace in Earth centered coordinates (km))
y -- (y-component of raytrace in Earth centered coordinates (km))
z -- (z-component of raytrace in Earth centered coordinates (km))
bx -- (x-component of magnetic field (nT))
by -- (y-component of magnetic field (nT))
bz -- (z-component of magnetic field (nT))
glon -- (geographic longitude (degrees))
glat -- (geographic latitude (degrees))
galt -- (geographic altitude (km))
aspect_angle -- (aspect angle (0 degrees is parallel to B-field, 180 degrees is anti-parallel to B-field))
edens -- (electron density (m^-3))
nref -- (refractive index^2) (**refractive index **squared**, not just refractive index**)

elevation_angle -- (elevation angle)
gcd -- (great circle distance (km))
slant_range -- (slant range (km))
bending_angle and rel_power -- (I'm honestly not completely sure what these two are exactly)

Now you can use these variables to calculate the various parameters of the rays at any point along the ray path (for example, you can integrate the refractive index and distance along the path to determine phase and Faraday rotation).

Already written analysis programs:
The programs 'Faradayrotcalc.pro' and 'trace_ana_nc_20110203.pro' can be used to analyse the ray traces.

'Faradayrotcalc.pro':
This program will take as input one ray file and the transmitter frequency and calculate the Faraday rotation that would occur along the ray path (assuming both modes followed the same ray path). At the moment it only works for ray traces which used the convergence mode, but that could easily be changed.

'trace_ana_nc_20110203.pro':
This program is quite extensive, but simple to use for raytraces which did not use the convergence mode. Simply call the subroutine 'analyze1' with the arguments of frequency, directory name, and electron density profile name. It will calculate and plot several parameters of the modelled signal at the specified altitude of the satellite. In the program, the directory names for the ray traces and electron density profiles will need to be modified (line numbers 12 and 16, I think). Unfortunately, this program is quite complex and not very well documented, but it should provide you with all the outputs you will need. It might be a good idea to write a simple program like 'Faradayrotcalc.pro' yourself to check the outputs of this program.