Meta.xml

The *meta.xml* file presents NRP with a set of metadata, such as the resource's name, the scripts to include, and which files to precache for sending to clients among other things. It is also the scope of "elements". It is written in XML, which is based on HTML and is the parent of XHTML.

Tags

XML is a textual data format which is widely used for the representation of data. NRP uses an XML-based language to describe the metadata for resources by using the tags below:

- **<info />** Information about this resource, possible parameters include (any arbitrary parameters can be used and read using getResourceInfo):
  - **author:** The author of this resource
  - **version:** The version of this resource
  - **name:** The name of this resource
  - **description:** A brief description of this resource
  - **type:** The type of this resource, that can be "gamemode", "script", "map" or "misc".
  - **gamemodes:** The gamemodes to be compatible with the resource. It must be the name of the gamemode resource, not the gamemode name. If you want it to be compatible with multiple gamemodes, it must be in a comma-separated list without spaces. (e.g. gamemodes="test1,test2").
- **<script />** Source code for this resource, possible parameters are:
  - **src:** The file name of the source code
  - **type:** The type of source code: "client", "server" or "shared".
    - A **shared** script will be ran for both client and server, but separately as usual (basically adds the script twice: once for server and once for client)
  - **cache:** When the script file type is "client", this setting controls whether the file is saved on the clients' hard drive. If you are concerned about your scripts security, make sure to read Script security guide. Default is "true". Using "false" will mean the file is not saved. *(Note: cache=false files are started at the client first, so lua file load order might differ when mixing cache settings)*
  - **validate:** If set to "false", compatibility checks are skipped.
- **<map />** The map for a gamemode, possible parameters are:
  - **src:** .map file name (can be path too eg. "maps/filename.map")
  - **dimension:** Dimension in which the map will be loaded (optional)
- **<file />** A client-side file. Generally these are images, .txd, .col, .dff or .xml files. They'll be downloaded by clients when the resources is started (or on join)
  - **src:** client-side file name (can be path too eg. "images/image.png")
  - **download:** Whether or not to be sent to the client automatically (optional). Default is "true". Using "false" will mean they are not sent on resource start but could later be used by downloadFile.
- **<include />** Include resources that this resource will use
  - **resource:** Resource name that you want to start with this resource
  - **minversion:** Minimum version that **resource** needs to be (optional)
  - **maxversion:** Maximum version that **resource** needs to be (optional)
- **<config />** Config file (.xml) can be accessed by resource, possible parameters are:
  - **src:** The file name of the config file
  - **type:** The type of the config file: "client" or "server"
- **<export />** This exports functions from this resource, so other resources can use them with call
  - **function:** The function name
  - **type** Whether function is exported server-side or client-side (valid values are: "client", "server" and "shared")
    - A **shared** export will make the function callable from both client and server scripts (basically adds the export twice: once for server and once for client)
  - **http:** Can the function be called via HTTP (true/false)
- **<html />**
  - **src:** The filename for the HTTP file (can be a path)
  - **default:** The html file is one that is shown by default when visiting /resourceName/ on the server. Only one html can be default, the rest are ignored. (true/false)
  - **raw:** The html file is not parsed by the Lua interpreter and is treated as binary data. Must be used for binary files (images mainly) (true/false)
- **<settings />** Most gamemodes use settings system to let server admins to configure it how they like. For instance you could set round time and then use get and set to get the value or change it, respectively.
  - **<setting />** Resource settings can be accessed by resource and Admin panel, possible parameters are:
    - **name:** The setting name used by the scripts to get or set the setting value
    - **value:** The setting default value used by the scripts
    - **friendlyname:** A friendly name to the setting (optional)
    - **accept:** The values the setting could accept (optional)
    - **examples:** An Example of a value (optional)
    - **desc:** A description of the setting (optional)
- **<min_nrp_version />** Minimum version requirements for this resource to run correctly. When authoring resources, the minimum version should usually be set to the current released version of NRP:SA (which at the

moment is "1.6.0"). See example for example.
    - **client:** The minimum client version
    - **server:** The minimum server version
- **<aclrequest />** A list of ACL rights this resource will need. Any user with admin permission can accept or reject a resource ACL request by using the command: /aclrequest [list/allow/deny] <resourceName> [<right>/all]
    - **<right />** an individual right
        - **name:** The right name.
        - **access:** Set to *true* to allow the resource to access this right. Set to *false* to deny the access to this right.

- **<sync_map_element_data />** Controls whether map element data such as "PosX" and "DoubleSided" are transferred to the client. This data is usually not required by most gamemodes or resources. (Map Editor and Interiors require this to be not set to false to work). When set in a gamemode meta.xml, the setting will apply to all maps loaded by that resource.
    - **false:** Disable transfer of map element data for all resources. This can reduce map download times considerably.
    - **true:** Enable transfer of map element data for all resources. (If **false** and **true** are set in different resources, true will have priority and all resources will transfer map element data)

- **<oop />** OOP - Please refer to OOP for documentation.
    - **false:** Disable OOP.
    - **true:** Enable OOP.

- **<download_priority_group />** If not set, the download priority group for a resource defaults to 0. If this is set higher than 0, then the resource will be downloaded and started on the client earlier than other resources. This option is useful for resources with critical client functionality that other things in your gamemode (or fair play) rely on. If set to less than 0 (a negative number, like -1), the resource will be downloaded and started on the client later than other resources. As this can be confusing, here is an example:
    - **Resource A: <download_priority_group>20</download_priority_group>** will start earlier than..
    - **Resource B: <download_priority_group>10</download_priority_group>**
      In this case, Resource A will start earlier than Resource B because its value (20) is higher than (10). In turn, Resource B will still start earlier than a resource with a negative value or a value below 10 (like 5).