# Public Methods

### registerTool

Parameters: tool object, toolId string, options hash

Description: Registers a tool in the tools array. This allows the system to fire events for each tool in the toolset. The options hash can carry event callbacks and grouping information to further help the framework understand the tool

### registerIntent

Parms: tool object, intent string

Parms: intent, callback function <----- perhaps this would be a better way to do this?

Description: Registers a tool for a specific intent. This will allow other tools to use the specified tool’s functionality without having to rewrite the tool or know the tool’s name.

### runIntent

Parms: intent string

Description: Gets the tool that is registered with a specific intent. Perhaps it might be best to use something more akin to ‘runIntent’ and let the system run the tool with some provided options. Then the tools themselves would not be directly callable from other tools.

### registerLayer

Params: layer object, update callback function, toolId string

Description: Adds a layer to the layers array. The tool must provide a way for the layer to be updated and a tool that owns/manages the layer.

### getLayers

Description: Gets all the layers that have been added to the system by tools

Params:

### getSite

Description: Gets a site by its ID (Should this be a copy? Force the tool to call updateSite?)

Params:ID of the site

### getSites

Description: Gets the whole array of sites on the map

Params:

### updateSite

Description: Updates a site by ID

Params: siteID

### invalidateScores

Description: Lets the system know that a tool has changed its scoreboard data and the rest of the system should see if it needs to reflect the changes

Params: The ID of the tool that updated it’s data

# Private Arrays

### tools[object]

An array that holds all the registered tools in the system. The tools will have to conform to a specification to have this make sense

### Layers [OpenLayers.Layer]

All the layers that are on the map. Perhaps we don’t need this as we have access to the map which has its own collection of layers

# Public Events

### onSiteChanged

Description: Will fire when a Site geometry or attribute is changed.

Params: The Site that has changed

### onToolAdded

Description: Raised after a tools has registered itself in the system. This may be used by other tools to monitor for co-tools.

Params: toolId string, intents[] strings

### onScoresInvalidated

Description: Will fire when something has changed on the scoreboard that may have caused it to be invalid. The event will provide the scoreboard and the intent object that invalidated the score. When a tool responds to this event, to avoid infinite looing, it must not also invalidate the scoreboard if it changes something.

Params: data object, event object

* The data object will have
  + Scoreboard: The system scoreboard that was invalidated
  + changedScoreIds[]: string names of the intents or toolIds of the tools that changed their score. This is may be used by other tools to determine if they too should recalculate their score or not. If the tool changes its score by using the updateScore method of the Scoreboard object then it will also be added to this array automatically.