

Relationship Manager - All possible scenarios

Tuesday, 10 June 2003

Generated by **RmApiGen.py**

```
===== onetoone =====
X has singularapi - Y has no API
*****
```

---- X methods ----

```
void setY(y)    RM.R(this, y, "xtoy")
Y    getY()    RM.P(this, "xtoy")
void clearY()   RM.NR(this, getY(), "xtoy")
```

---- Y methods ----

None

```
===== onetoone =====
X has no API - Y has singularapi
*****
```

---- X methods ----

None

---- Y methods ----

```
X    getX()    RM.B(this, "xtoy")
void setX(x)    RM.R(x, this, "xtoy") [must first clearX(), though..see note 1A]
void clearX()   RM.NR( RM.B(this, "xtoy"), this, "xtoy")
```

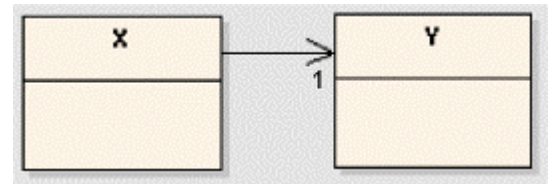
```
===== onetoone =====
X has singularapi - Y has singularapi
*****
```

---- X methods ----

```
void setY(y)    RM.R(this, y, "xtoy")
Y    getY()    RM.P(this, "xtoy")
void clearY()   RM.NR(this, getY(), "xtoy")
```

---- Y methods ----

```
X    getX()    RM.B(this, "xtoy")
void setX(x)    RM.R(x, this, "xtoy") [must first clearX(), though..see note 1A]
void clearX()   RM.NR( RM.B(this, "xtoy"), this, "xtoy")
```



```

===== onetomany =====
X has pluralapi - Y has no API
*****

```

```

---- X methods ----
void addY(y) RM.R(this, y, "xtoy")
list getAllY() RM.PS(this, "xtoy")
void removeY(y) RM.NR(this, y, "xtoy")

```

```

---- Y methods ----
None

```

```

===== onetomany =====
X has pluralapi - Y has singularapi
*****

```

```

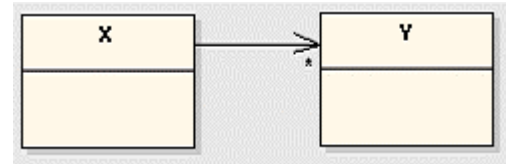
---- X methods ----
void addY(y) RM.R(this, y, "xtoy")
list getAllY() RM.PS(this, "xtoy")
void removeY(y) RM.NR(this, y, "xtoy")

```

```

---- Y methods ----
X  getX() RM.B(this, "xtoy")
void setX(x) RM.R(x, this, "xtoy") [must first clearX(), though..see note 1A]
void clearX() RM.NR( RM.B(this, "xtoy"), this, "xtoy")

```



```

===== manytoone =====
X has no API - Y has pluralapi
*****

```

```

---- X methods ----
None

```

```

---- Y methods ----
void addX(x) RM.R(x, this, "xtoy")
list getAllX() RM.BS(this, "xtoy")
void removeX(x) RM.NR(x, this, "xtoy")

```

```

===== manytoone =====
X has singularapi - Y has pluralapi
*****

```

```

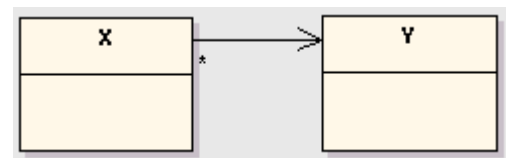
---- X methods ----
void setY(y) RM.R(this, y, "xtoy")
Y  getY() RM.P(this, "xtoy")
void clearY() RM.NR(this, getY(), "xtoy")

```

```

---- Y methods ----
void addX(x) RM.R(x, this, "xtoy")
list getAllX() RM.BS(this, "xtoy")
void removeX(x) RM.NR(x, this, "xtoy")

```



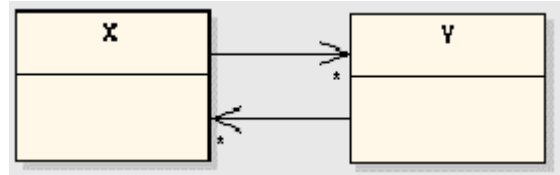
```
===== manytomany =====
X has ['pluralapi'] - Y has ['pluralapi']
*****
```

```
---- X methods ----
```

```
void addY(y) RM.R(this, y, "xtoy")
list getAllY() RM.PS(this, "xtoy")
void removeY(y) RM.NR(this, y, "xtoy")
```

```
---- Y methods ----
```

```
void addX(x) RM.R(this, x, "ytox")
list getAllX() RM.PS(this, "ytox")
void removeX(x) RM.NR(this, x, "ytox")
```



```
===== manytomany =====
X has ['pluralapi'] - Y has ['pluralapi', 'singularapi']
*****
```

```
---- X methods ----
```

```
void addY(y) RM.R(this, y, "xtoy")
list getAllY() RM.PS(this, "xtoy")
void removeY(y) RM.NR(this, y, "xtoy")
```

```
---- Y methods ----
```

```
void addX(x) RM.R(this, x, "ytox")
list getAllX() RM.PS(this, "ytox")
void removeX(x) RM.NR(this, x, "ytox")
```

```
X    getX() RM.B(this, "xtoy")
void setX(x) RM.R(x, this, "xtoy") [must first clearX(), though..see note 1A]
void clearX() RM.NR( RM.B(this, "xtoy"), this, "xtoy")
```

```
===== manytomany =====
X has ['pluralapi', 'singularapi'] - Y has ['pluralapi']
*****
```

```
---- X methods ----
```

```
void addY(y) RM.R(this, y, "xtoy")
list getAllY() RM.PS(this, "xtoy")
void removeY(y) RM.NR(this, y, "xtoy")
```

```
Y    getY() RM.B(this, "ytox")
void setY(y) RM.R(y, this, "ytox") [must first clearY(), though..see note 1A]
void clearY() RM.NR(getY(), this, "ytox")
```

```
---- Y methods ----
```

```
void addX(x) RM.R(this, x, "ytox")
list getAllX() RM.PS(this, "ytox")
void removeX(x) RM.NR(this, x, "ytox")
```

```

===== manytomany =====
X has ['pluralapi', 'singularapi'] - Y has ['pluralapi', 'singularapi']
*****

---- X methods ----
void addY(y)      RM.R(this, y, "xtoy")
list getAllY()   RM.PS(this, "xtoy")
void removeY(y)  RM.NR(this, y, "xtoy")

Y      getY()      RM.B(this, "ytox")
void setY(y)      RM.R(y, this, "ytox") [must first clearY(), though..see note 1A]
void clearY()     RM.NR(getY(), this, "ytox")

---- Y methods ----
void addX(x)      RM.R(this, x, "ytox")
list getAllX()   RM.PS(this, "ytox")
void removeX(x)  RM.NR(this, x, "ytox")

X      getX()      RM.B(this, "xtoy")
void setX(x)      RM.R(x, this, "xtoy") [must first clearX(), though..see note 1A]
void clearX()     RM.NR( RM.B(this, "xtoy"), this, "xtoy")

```