Working with Biconditionals

We will here introduce two rules that pertain specifically to biconditionals. Any biconditional p <-> q is logically equivalent to (p -> q) & (q -> p). Since 'and' is commutative it is also logically equivalent to (q -> p) & (p -> q). These are the basis of our <->E and <->I rules. Here is the first, our elimination rule:

<->E

From	p <-> q	p <-> q
To	(p -> q) & (q -> p)	(q -> p) & (p -> q)

Here is a fairly simple derivation using this rule:

1. (A v B) <-> Q	Premise
2. ~A & Q	Premise
3. ~A	2 &E
4. Q	2 &E
5. ((A v B) -> Q) & (Q -> (A v B))	1 <->E
6. Q -> (A v B)	5 &E
7. A v B	4, 6 ->E
8. B	3, 7 DS

We have established that (A v B) <-> Q, ~A & Q ⊢ B. Here is another derivation using this rule:

1. (A <-> B) <-> Q	Premise
2. A & Q	Premise
3. ((A <-> B) -> Q) & (Q -> (A <-> B))	1 <->E
4. Q -> (A <-> B)	3 &E
5. A	2 &E
6. Q	2 &E
7. A <-> B	4, 6 ->E
8. (A -> B) & (B -> A)	7 <->E
9. A -> B	8 &E
10. B	5, 9 ->E

You should have no problem with this rule provided that you remember it only applies to a line that is a biconditional.

Our <->I rule is what you would expect:

<->I

From	(p -> q) & (q -> p)	(p -> q) & (q -> p)
To	p <-> q	q <-> p

Here is a derivation in which we utilize this rule:

1. P v Q	Premise
2. Q -> (R -> S)	Premise
3. P -> (R -> S)	Premise
4. Q -> (S -> R)	Premise
5. P -> (S -> R)	Premise
6. R -> S	1, 2, 3 vE
7. S -> R	1, 4, 5 vE
8. (S -> R) & (R -> S)	6, 7 &I
9. R <-> S	8 <->I

Here is another derivation:

1. P v ((A & B) -> C)	Premise
2. ~P & Q	Premise
3. Q -> (P v (C -> (A & B)))	Premise
4. ~P	2 &E
5. (A & B) -> C	1, 4 DS
6. Q	2 &E
7. P v (C -> (A & B))	3, 6 ->E
8. C -> (A & B)	4, 7 DS
9. ((A & B) -> C) & (C -> (A & B))	5, 8 &I
10. (A & B) <-> C	9 <->I

Here we have shown that P v ((A & B) -> C), ~P & Q, Q -> (P v (C -> (A & B))) |- (A & B) <-> C. As long as you pay careful attention to the form of the lines, you should have no problems in using these rules.